Structured and Real Time Heterogeneous Sensor Deployment in Preferential Areas

Rabie Ramadan
Al-Azhar University,
Cairo, Egypt
rabie@rabieramadan.org

Fatma Ziwar
German University in Cairo (GUC),
Cairo, Egypt
fatma.ziwar@student.guc.edu.eg

ABSTRACT
In this paper we investigate the problem of heterogeneous sensor deployment in preferential areas. The problem considers many of the sensors characteristics such as mobility, state-switching, reliability, and mobility cost; in addition, the problem takes into consideration that the monitored field areas may differ in their monitoring requirements from time to time. Different prediction methods namely Markov, double exponential smoothing, triple exponential smoothing, simple average, and weighted average are used to predicate the monitoring field preferential areas. Our approach in solving such problem starts by formulating the problem mathematically to show its complexity and to solve small-scale problems optimally. Then, we propose three different algorithms for large scale problems. The first algorithm deals with structured deployment where the monitored field is assumed accessible. The second and third algorithms deal with real time deployment where the sensed data is used for future planning and sensor relocation. The second algorithm is a centralized algorithm while the third algorithm is a distributed algorithm. An extensive set of experiments are conducted to show the performance of the proposed methods and algorithms.

Categories and Subject Descriptors
C2.1 [**Network Architecture and Design**]: Sensor network deployment- *heterogeneous sensors on preferential areas*

General Terms
Algorithms, Performance, Design, Reliability, Experimentation

Keywords
Sensor Networks, preferential Areas, Security, Monitoring, Sensor Deployment, Distributed Algorithms.

1. INTRODUCTION
The advances in sensing technology lead to the usage of sensor networks in many applications. For instance, sensors have been used to monitor animals in habitat areas and monitor patients' health. In addition, sensor networks have been used to monitor critical infrastructures such as gas, transportation, energy, and water pipelines as well as important buildings. Such applications require sensors to live for a long time which leads to a large body of research in efficient energy algorithms and protocols.

The monitored fields have been modeled differently due to the application requirements. Dividing the monitored fields into virtual grid of cells seems to be suitable for applications with mobile sensors [2] and critical infrastructure monitoring applications [10]. On the other hand, a free model might be suitable for applications such as health monitoring where patients can be located using different methods. In this paper, we divide the monitored field into a virtual grid of cells and assign a weight for each cell. This weight represents the importance of each cell or the monitoring requirement of this cell. However, this weight might differ from time to time. For instance, sensors could be used to monitor an airport; the runways monitoring importance certainly will increase when a flight is using it. Monitoring certain areas in a school might differ in their importance during the day due to the students' activities; while this importance will be the minimum during the night.

Currently, we have mobile sensors that can move from one place to another. For instance, sensors could be mounted on a robot or attached to animals or human bodies. This movement could help in the deployment of other sensors or could be utilized to cover blind spots in the monitored field [6]. However, the movement is always costly in terms of the sensors energy. Therefore, sensors movement has to be carefully decided. In addition, due to the sensors energy limitations many of the proposed algorithms and protocols utilize the switching capabilities of the sensors. For instance, Berfield at el. in [1] propose a distributed scheduling algorithm that assigns predefined slot for each sensor. The authors claim that their algorithm increases the sleeping time for each sensor; therefore it saves the sensors energy. In addition, sensors reliability is considered in many of the research proposals such as [8] and [4].

Sensor deployment represents the first phase in forming any sensor network. Deployment algorithms basically could be classified into two classes which are arbitrary and structured/offline. In the latter, the access to the monitored field is assumed granted and sensors could be placed in their exact positions. In such situation, sensors optimal placement could be examined such as in [5]. On the other hand, arbitrary deployment assumes limited access to the monitored field as well as large number of sensors to be deployed. Sensors in this case could be deployed using flying robots or unmanned aerial vehicles [3]. Most of the previous deployment algorithms consider only some of the sensors parameters (e.g. energy). The monitored field parameters (e.g. preferential areas and obstacles) are also ignored during the deployment process. In addition, current research proposals did not take the field activities into account after or even before the deployment. In other words, re-locating the sensors after the initial deployment according to the monitored field requirements did not have that much attention in the sensor networks community.

In this paper, we consider sensor deployment taking into account most of the monitored field as well as the sensors parameters. We believe that considering such parameters during the deployment process will increase the performance of the network as well as prolong its lifetime. Not only we consider the sensor deployment but also we study the re-placement of the sensors based on the monitored field activates. We propose to use some of the simple and efficient prediction algorithms to utilize sensors capabilities such as mobility, reliability, and switching. Moreover, the problem is formulated mathematically for small-scale deployment problems. Nevertheless, we propose centralized and distributed sensor deployment algorithms for dense networks. An exhaustive set of experiments are conducted to show the affect of utilizing such capabilities on the performance and lifetime of the sensor network.

The paper is organized as follows: in the next section, the problem is formally defined; in section 3 different solutions to the deployment problem is explained including the mathematical formulation, structured, real-time, and distributed algorithms. Section 4 depicts the experimental results; finally, the paper concludes in section 5.

2. PROBLEM DEFINITION, MODELING, AND FORMULATION

A deployment field *F(A)* with differential security requirements is to be monitored for a time horizon of length *T*. In addition, a set of heterogeneous sensors $|S|$ is given. These sensors differ in their capabilities such as lifespan/lifetime, allowed number of state-switching, allowed number of moves, movement cost, and reliability. The objectives are to achieve the maximum coverage of the monitored field, exploit sensors' capabilities, prolong sensor network lifetime, and increase the security of the monitored field. The latter is enhanced by using the highest reliable sensors on the most important areas in the field. The deployment problem is modeled as follows:

The monitored field is divided into a number of zones $|A|$. Each zone $i \in A$ is associated with a time-varying weight function w_i^t, where $t \in T$. This weight function defines the importance of the observations (surveillance requirement) in this zone over the horizon T. A sensor lifespan L_s is the number of time units that the sensor was used to monitor one or more zones of the monitored field. Each sensor lifespan is associated with a cost e_s. Therefore, sensors lifespan could be represented in terms of energy by E_{sij}^t. In addition to lifespan, sensors are assumed to have an allowed number of state-switching P_s in which a sensor $s \in S$ can change its state from "on" to "off" or vice versa based on the field requirements. For instance, a sensor $s \in S$ could be switched to "off" at time $t \in T$ to save its lifetime for a different zone with a higher security requirement w_i^t at different times. The security requirement w_i^t represent s the field preferential monitoring requirement in which it differs with time and could be estimated based on the monitored area history or collected by a centralized node in case of real time monitoring. Moreover, a sensing device could be stationary or mobile. If a stationary device is deployed on a zone $i \in A$, this device is assumed to remain in this zone for its entire lifespan. On the contrary, a mobile sensor can cover multiple zones over a time period *T*. All mobile devices are assumed to have no restrictions on the start or the end locations of their deployment, but they have restrictions, M_s, per sensor on the number of moves from zone to another. A sensor transfer between two zones is assumed to be associated with a cost. This cost is expressed in terms of the loss in the device lifespan E_{sij}^t. Nevertheless, each sensor $s \in S$ is characterized by a predefined reliability R_s^t that typically changes over time.

Considering the heterogeneity of sensing devices and the differentiated security requirements, an optimal deployment scheme is required. The scheme should exploit the sensors' capabilities as well as cover the most important areas of the monitored field.

3. SOLUTION APPROACH

Our solution approach to the large-scale deployment problems presented in this paper includes two phases which are the prediction and the sensor deployment algorithms. In the prediction phase, different prediction methods are implemented and tested on real field data taken from sensors already deployed at different places in our school for this purpose. We experimented with different prediction methods such as Simple Moving and Weighted Average, Markov Chain, Double Exponential Smoothing, and Triple Exponential Smoothing. There are some other prediction methods that might perform better than the selected ones. However, their computation complexities are not suitable for a limited sensor capability. For instance, Kalman filter [5] runs 135 slower than double exponential smoothing algorithm. Due to the limited space in this paper, the reader is referred to **Error! Reference source not found.** [5] for the details of the prediction methods. These methods are compared to each other in terms of the amount of computation as well as the accuracy.

Once the monitored field weights are available, we use one of our proposed algorithms to solve the deployment problem. In our previous work [9], the deployment problem is formulated in the form of an integer mathematical program and tested for small-scale problems. The formulation expresses the complexity of the problem and the exact solution that can be used only for small-scale problems. Therefore, we propose three different algorithms which are Offline Sensor Deployment Algorithm (OSDA), Real-time Sensor Deployment Algorithm (RSDA), and Distributed Real-time Sensor Deployment Algorithm (DRSDA). In OSDA the deployment, is based on the prediction data during the monitoring horizon. On the other hand, RSDA is combined of two steps; the first step considers the initial sensor deployment while the second step considers the real time sensor rearrangement based on the sensed data from the monitored field. In DRSDA, we are trying to minimize the number of messages to be sent from the sink node to the sensors every time a replacement is required. Nodes are clustered into a number of groups where each cluster head acts as a sink node. These cluster heads may coordinate for better monitoring as well.

3.1 Offline Sensor Deployment Algorithm (OSDA)

OSDA runs in two phases; in the first phase, we use our prediction algorithms to predict the weights w_i^t of each cell over the horizon *T*. In the second phase, we deploy the sensors accordingly. As depicted in Table 1 (part A), the algorithm works as follows:

1- For the entire horizon *T*, sensors are sorted based on their reliability and the cells are sorted based on their importance.

2- Assign the highest reliable sensor to the highest weight cell at any given time with taking into consideration the consumed energy for each sensor. The consumed energy is due to the switching as well the mobility.

If a sensor's energy is depleted, remove it from the sensors' list.

3.3 RSDA: Real-Time Sensor Deployment Algorithm

In this section, we elaborate on our real time deployment algorithm. Again, the deployment is done in two phases; in the first phase, the next time cells weighs are predicted while in the second phase, we deploy the sensors based on predicted values. In other words, based on the available history for each cell, the cells next time weights are predicted. Then, the sensors are deployed accordingly. Fortunately, we developed many of the prediction algorithms; all of them predict for one future time unit but the triple exponential smoothing algorithm predicts for three future time units. Once the sensors are deployed for the first time, the real cells weights are sensed and added to the cells weights history; then new weights are predicted. According to this prediction, some of the mobile sensors might be re-placed for better coverage.

Once more, the question in such case, is it better to use triple exponential smoothing prediction and plan the mobile sensors movement for three times a head or plan it only for one time a head? This question will be answered in the experimental results section. The algorithm can be formally described in Table 1 (part B).

Table 1: OSDA and RSDA algorithm

Input:

A the numbers of cells in the monitored field

w_i^t monitored field weight history for *k* time units where $t \in k$ and $i \in A$

E_s sensor's initial energy where $s \in S$

e_s sensor's monitoring energy per unit of time *t*, where $s \in S$

E_s^d sensor's moving energy per distance *d*

E_{on} sensor's switching *on* energy

E_{off} sensor's switching *off* energy

T the monitoring horizon

R_s^t sensor's reliability where $t \in T$ and $s \in S$

α_s sensor's energy threshold where $s \in S$. It means that s energy cannot be less than α_s

d_m sensor's consumed energy due to moving a meter.

PART A: OSDA ALGORITHM

Output :

Sensors deployment schemes

Steps:

1. Predict the cells weights w_i^t for the *T* times using one of the prediction algorithms based on the *k* times history, where $t > k$;
2. $\forall\ T \rightarrow$ sort w_i^t where $i \in A, t \in T\ and\ t > k$;
3. $\forall\ T \rightarrow$sort R_s^t where , $s \in S$ and $t \in T\ and\ t > k$
4. $\forall\ T$, assign s with the highest R_s^t to $i \in A$ with the highest w_i^t

4.1. Compute the current sensors energy C_s based on current deployment scheme as follows:

$$C_s = E_s - \sum_{t=k+1}^{T} X_s^t.e_s + Y_s^t.E_{on} + Z_s^t E_{off} + .E_s^d ,$$

Where X_s^t is a binary variable that is set to 1 if the sensor is used in monitoring process , Y is a binary variable that is set to 1 if the sensor is switched to on , Z_s^t is a binary variable that is set to 1 if the sensor is switched to off, d_s^t is the distance that the sensor moved at time *t*.

4.2. If ($C_s > \alpha_s$)

4.2.1. neglect the current assignment

4.2.2. Remove the sensor from the sensors list

4.2.3. Print the sensor deployment scheme

4.2.4. Continue

4.3. Else if ($C_s = \alpha_s$)

4.3.1. Remove the sensor from the sensors list

4.3.2. Print the sensor deployment scheme

4.3.3. Continue

PART B: RSDA ALGORITHM

Output :

Sensors deployment and replacement schemes

Steps:

1. Predict the cells weights w_i^t for the 1 time using one of the prediction algorithms based on the *k* times history, where $t > k$;
2. sort w_i^{tc} where $i \in A, t_c \in T\ and\ t_c > k$, where t_c is the current time unit ;
3. sort R_s^{tc} where , $s \in S$ and $tc \in T\ and\ tc > k$;
4. $\forall S$, assign s with the highest R_s^t to $i \in A$ with the highest w_i^t
5. Compute the current sensors energy C_s based on current deployment scheme as follows:

$$C_s = E_s - \sum_{t=k+1}^{tc} X_s^t.e_s + Y_s^t.E_{on} + Z_s^t E_{off} + d_s^t.E_m ,$$

Where X_s^t is a binary variable that is set to 1 if the sensor is used in monitoring process , Y is a binary variable that is set to 1 if the sensor is switched to on , Z_s^t is a binary variable that is set to 1 if the sensor is switched to off, d_s^t is the distance that the sensor moved at time *t*.

6. If ($C_s > \alpha_s$)
   a. neglect the current assignment
   b. Remove the sensor from the sensors list
   c. Print the sensor deployment scheme
   d. Continue
7. Else if ($C_s = \alpha_s$)
   a. Remove the sensor from the sensors list
   b. Print the sensor deployment scheme
   c. Get the current cells weights from the sensors
   d. Add the cells weights to the history
   e. If number of sensor in the sensors list > 0 go to step 2 else go to step 8
1. Stop

3.4 DRSDA: Distributed Real-Time Sensor Deployment Algorithm

In the previous section, we proposed a centralized real-time deployment algorithm where the overall prediction is done at the sink node and a message for sensors that need to change its state is sent every time unit. In addition, RSDA prioritized the importance of the monitored areas/cells over the sensors lifetime. In other words, the algorithm does not take into consideration the number of messages that need to be sent from the sink to the nodes in the network. It also assumed one hop network where every node is directly connected to the sensors. RSDA is expected to perform well (see the experimental result section) when a small number of sensors are deployed. However, in a dense network, changes in the sensors states as well as the topology might require forming an ad hoc network as well as sending a large number of messages to the deployed sensors.

Therefore, in this section, we introduce a distributed real-time sensor deployment algorithm named DRSDA. The algorithm considers a dense sensor network that is deployed in a large area. Therefore, the area is partitioned into subareas and a node is selected to act as a sink node for each partition. At the same time, it acts as a cluster head for a number of sensors that are initially deployed on its partition. In other words, the overall area is initially partitioned into *p* subareas. Then, a cluster head is selected for the deployed sensors in each partition. A cluster head is changed periodically to balance the nodes energy. After selecting the cluster head for each partition, RSDA is applied within each partition. We adapted the clustering algorithm proposed in [11] to fit our problem modeling. Such clustering is expected to minimize the number of exchanged messages among the sensors and the sink node.

4. SIMULATION RESULTS

In this section, we limit ourselves to some of the experiments that have been conducted, based on our C# simulation, due to the strict limitations on the paper number of pages. A set of experiments is used to compare the performance of the proposed algorithms. All sensors parameters are based on Mica2 characteristics such as initial energy, monitoring energy, switching energy, and reliability.

4.1 OSDA and Optimal Solution

In this section, we report our finding in comparing the optimal solution to OSDA algorithm. In general and as explained in [9], the optimal solution was limited to almost 20 cells, 10 sensors, and 12 unit of times. We used the same experiments configuration presented in [9] and compared its results to OSDA algorithm. It turned to be, on average; OSDA gives 80% of the optimal solution and consumes almost 20% of the optimal solution required computation.

4.2 Prediction Algorithms Comparison

To select the best prediction algorithm, we experimented with the five prediction algorithms proposed in this paper for the deployment problem. In order to compare between the predictions algorithms, we selected the Mean Square Error (MSE) as an evaluation value. As shown in Figure 1, six cells are selected to be monitored in this experiment where different history lengths are used to measure the performance of the algorithms.

As can be seen, the first three algorithms (Markov, Moving average, and Weighted average) perform almost the same. However, it seems that the best two prediction algorithms are double and triple exponential where the MSE is the least. We are taking the triple exponential smoothing into consideration while the double slightly over performs it due to its importance in generating the next three predicted values instead of only one predicted value. In addition, in terms of processing our experiments show that Markov chain requires the most processing capabilities and moving average requires the least.

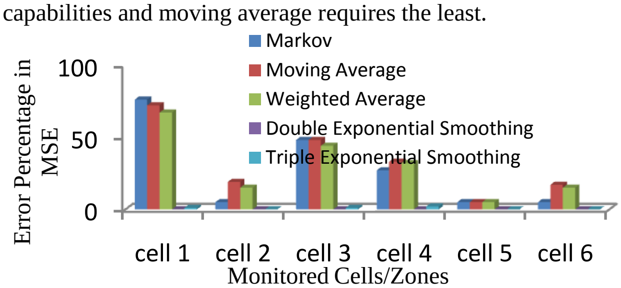

Figure 1: Prediction algorithms comparison

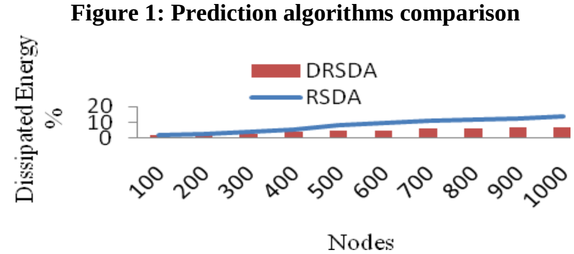

Figure 2: Comparison between RSDA and DRSDA in terms of the dissipated energy

4.3 Effect of the Number of Nodes on RSDA and DRSDA Dissipated Energy

In this set of experiments, we compare between the RSDA and DRSDA in terms of the dissipated energy. As shown in Figure 2, the number of nodes is increasing gradually from 100 to 1000 nodes. The average dissipated energy per node is measured in each case due to the real time deployment process. In all experiments, using DRSDA, the monitored area is divided into subareas and a node is selected to serve as a cluster head for each subarea. In addition, this results id due to different number of subareas. The average values where the initial energy is generated randomly using a normal distribution function.

As can be seen in Figure 2, DRSDA reduces the average dissipated energy per node by almost 45%. For instance, using 1000 nodes, the dissipated energy percentage of RSDA is 12% of the overall sensors' energy while DRSDA consumes only 7% of the overall sensors' energy.

4.4 Effect of the Number of Nodes on RSDA and DRSDA Coverage Performance

In this section, a set of experiments is conducted to study the effect of the number of nodes on RSDA and DRSDA coverage performance. Again, the number of nodes is increased from 100 to 1000 nodes. In DRSDA, different number of partitions is used and the coverage performance is averaged over all the conducted experiments. The coverage performance is computed by multiplying the covered cell weight by the sensors reliability at the time of monitoring. The coverage performance is computed over half monitoring day which is 12 hours.

Figure 3: Comparsion between RSDA and DRSDA in terms of the coverage performance

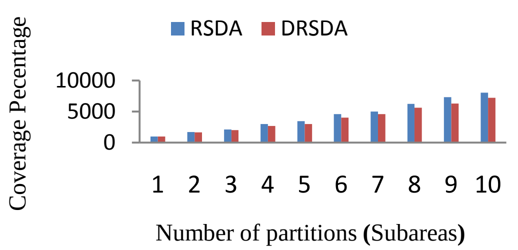

Figure 4: Comparison between RSDA and DRSDA with various numbers of partitions

As shown in Figure 3, DRSDA performance is slightly less than RSDA coverage. However, it is a tradeoff between saving the sensors energy and coverage as given in Figure 2. At the same time, it seems that the coverage difference may increase with the number of the number of subareas used with DRSDA algorithm as shown in Figure 4. The results show that the increasing in the number of subareas gives less coverage performance due to the limited number of movements allowed to mobile sensors; mobile sensors movement are restricted within the their subareas. Figure 4 results are based on different network topologies for 1000 heterogeneous nodes.

5. CONCLUSION

In this paper, we studied five prediction methods for predicting the importance of the monitored area. These prediction methods include Markov, simple average, weighted average, double exponential smoothing, and triple exponential smoothing algorithms. Our experiments show that the double and triple exponential smoothing algorithms are the least error values. In addition, we proposed three different deployment algorithms; the first algorithm considered the offline deployment where the cells importance is predicted for the monitored time and the sensors deployed accordingly. In the second algorithm, the cells importance is predicted per unit of time using double exponential smoothing or by three time units using triple exponential smoothing algorithm. Then the sink node sends a message to the nodes that need to change its location or state. Moreover, we proposed another distributed algorithm to save some of the sink messages that need to be sent every time unit to the sensors. This algorithm is based on dividing the monitored area into subareas and a node is selected to act as the sink for this area. By comparing the centralized and the distributed approaches, we found that there is a tradeoff between saving the sensors energy using the distributed algorithm and enhancing the coverage performance using the centralized approach.

6. REFERENCES

[1]. Berfield A. and Mosse D. ," efficient Scheduling for Sensor Networks," IWASN, 2006

[2]. Chih-Kuang L. , Vladimire Z., and Prashant K., "Grid-Based Access Scheduling for Mobile data Intensive Sensor Networks," the Ninth International Conference on Mobile Data Management , pages 197-204, 2008.

[3]. Corke P., Harbar S., Peterson R. Rus D, Saripalli S., and Sukhatme D., "Deployment and connectivity repair of sensor network using an unmanned aerial vehicle," In proceedings of IEEE International on Robotics and Automation, Vol. 4, pages 3602-3608, 2004.

[4]. Elouedi Z., Melloudi K., and Smets P., " Assessing sensor reliability for multisensor data fusion within the transferable belief model," IEEE transactions on Systems and Cybernetics, Vol 34, Issue 1, pages 782-787, 2004.

[5]. LaViola, J. Double Exponential Smoothing: An Alternative to Kalman Filter-Based Predictive Tracking. In Proceedings of the Immersive Projection Technology and Virtual Environments 2003 (May 2003), ACM Press, pp. 199-206.

[6]. Liu B., Brass P., Dousse O., Nain P., and Towsely D., "Mobility Improves Coverage of Sensor Networks,"Proceedings of MOBIHOC, 2005.

[7]. Mica2DataSheetat http://www.xbow.com/products/product_pdf_files/wireless_pdf/MICA2_Datasheet.pdf

[8]. Park S. and Sivakumar R. , " Sink-to-sensors reliability in sensor networks,", SIGMOBILR Mobile Computing and Communications review , Vol 7, Issue 3, pages 27-28, 2003.

[9]. Ramadan, R., Abdelghany, K., El-Rewini, H. Optimal and approximate approaches for deployment of heterogeneous sensing devices. In EURASIP JWCN Journal, special issue in Mobile Multi-hop Ad hoc Networks, 2007.

[10]. Roman R., Alcarza C., and Lopez J. "The role of wireless sensor networks in the area of critical information infrastructure protection," Information Security Technical Report Vol. 12 , Issue 1, pages 1363-4127, 2007.

[11]. Seema S. and Coyle E., "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," INFOCOM, IEEE computer and Communication societies, vol. 3, pp 1713-1723, 2003.

[12]. Toumpis S. and Tassiulas L., "Optimal deployment of large wireless sensor networks," IEEE transaction on information theory, Vol. 52, Issue 7, pages 2935-2953, 2006.