Efficient Deployment Algorithms for Mobile Sensor Networks

Salah Abdel-Mageid
Systems and Computers Department
Al-Azhar University
Cairo, Egypt
smageid@azhar.edu.eg

Rabie A. Ramadan
Computer Engineering Department
Cairo University
Cairo, Egypt
rabie@rabieramadan.org

***Abstract*— Sensor deployment problem is one of the important problems in Wireless Sensor Networks (WSN) since it represents the first phase that most of the network operations depends on. Sensor deployment strategies can be classified into two classes which are deterministic and autonomous (random) deployment. In the deterministic deployment, the deployment field is assumed accessible as well as the number of sensors is small to be manually deployed in specific locations. On the other hand, with large number of sensors and in inaccessible fields, the random deployment to the sensors turns out to be the solution. However, random deployment requires sensors to be automatically located (move) for coverage and connectivity purposes. In addition, after a period of time, the sensors topology might change due to some sensor hardware failure or deplaned energy. Therefore, redeployment and/or sensors relocation process is essential. Nevertheless, mobility consumed energy as well as sensor load balancing are essential factors to be considered during the initial deployment and relocation processes. This paper proposes two deployment algorithms to manage those situations. Those algorithms achieve sensor energy balancing and small amount of deployment energy consumption. A set of simulation experiments are conducted to compare between the proposed algorithm and the existing work in terms of coverage performance, average moving distance, and message complexity.**

***Keyw1ords*—mobile sensor networks, deployment, clustering, potential field, redundant sensors.**

I. INTRODUCTION

Recent advances in wireless sensors technologies have significantly broaden their applications. For example, wireless sensor networks (WSNs) are proposed to many civil and military applications including remote environment monitoring, smart homes, in-plant robotic control and guidance and military surveillance [12][5]. All those applications require deploying sensors to reach an adequate coverage level; therefore, the phenomena of interest can be efficiently sensed [6][14].

An autonomous (random) sensor deployment is required for un-reachable human environments, such as remote harsh fields or disaster areas in which sensor deployment cannot be performed deterministically or precisely. In such situations, the mobile sensors are necessary to move to improve the coverage performance and to additionally deal with node failure through the network lifetime.

Many proposals have been carried out to solve the problem of autonomous sensor deployment depending on the potential field theory such as [2], [8], [1] and [15]. The authors in [2], [8], and [1] proposed distributed algorithms in which mobile sensors can individually collect locations' information from their neighbors to make its movement decision based on the potential field computations. As a result, each sensor can move to certain location resulting in monitored field coverage enhancement. Unfortunately, this scenario spends number of rounds to reach near optimal coverage. Moreover, the movement cost in terms of energy consumption clearly increases. On the other hand, the work in [15] suggested a clustering technique to save the communication cost resulting in more computations at cluster heads. After running the potential field computations at cluster heads, each mobile sensor receives its final destination from the cluster head and the mobile node has to once move to it. However, the movement cost at the redeployment phase is slightly improved because this work uses traditional potential field computations. The same steps are also applied when the change in the network topology is discovered due to sensor failure. One of the disadvantages of such relocation of some of the mobile nodes is the reduction in coverage, in many cases, due to that the moved sensors might leave another spot without coverage.

As mentioned, the redeployment and relocation phases are essential phases to the point that some proposals only consider that phases. For instance, the authors in [9] assume a hybrid solution which a distributed solution was introduced for the redeployment phase [8] and clustered solution for the relocation phase. Clustered solution allows only the redundant sensors to move from one cluster to another in order to fill the coverage gaps due to sensor failure. In this case, the path from the redundant sensor position to the destination position is determined and all sensors belonging to that path move to make energy consumption balance. As a result, the higher number of sensors is used, the higher number of messages is required to find the path which increases the communication cost. It is obvious that the cluster head in the redeployment phase acts as any mobile sensor which in fact sacrifices a reasonable amount of its energy for the benefit of converge. Thus, a cluster head might be the weakest link in the redeployment process and its failure is highly expected.

Therefore, the node might not be able to reach its destination and if it is able to reach the destination, it might not be able to be part of the sensing and routing operations due to the lack of energy.

This paper introduces energy aware algorithms to avoid some of the drawbacks in [15] and [9]. The proposed algorithms adapt the concept of potential fields among the deployed nodes in a new form. The proposed solution uses the clustering technique in both of the redeployment phase and the relocation phase. As assumed, each cluster head can estimate the cluster area size from available information about the cluster boundary. At the redeployment phase, when the cluster head discovers the number of cluster's sensors is small to cover the cluster area size, some sensors are virtually added to reach the critical number to cover. Afterward, the potential field strategy is exploited to determine the final locations for cluster's sensors. We additionally introduce a sensor oscillation prevention technique to minimize the sensor's movement cost. Nevertheless, we restrict the movement of both of the cluster head and the redundant sensors to save their remaining energy to the relocation phase.

This paper is organized as follows: Section II introduces the first proposed algorithm which each cluster head individually runs to achieve high internal coverage or interacts with the cluster heads due to the virtual sensors presence. Another proposed algorithm is presented in Section III which the cluster heads follow interacting to solve the relocation problem due to missed sensor(s). The performance comparisons are conducted in Section IV to show the proposed model achieves high performance compared with the current solution. Section V concludes this paper.

II. Our Deployment Algorithms

In this section, we describe our deployment algorithms, the Cluster-Based Redeployment Algorithm (CBRA) and Energy Aware Relocation Algorithm (EARA).

A. *CBRA Algorithm*

As mentioned, CBRA adapts the concept of virtual force among the deployed sensor nodes. It also considers mobile and stationary nodes during the redeployment as well the relocation phase. The CBRA has some reasonable assumptions including: 1) the sensing area of each sensor inside a cluster is approximated by a circle with radius r_s indicating its sensing range, 2) after the initial random deployment, all sensors can determine their initial position by GPS services or based on a localization algorithm (i.e. [4], [3]) all sensor nodes are able to communicate with the cluster head to send their information, 4) the cluster head is responsible for executing the CBRA algorithm and managing the one-time movement of sensors to the desired locations, 5) the repulsive force among sensors belonging to that cluster is considered and the attractive force is neglected, 6) in order to minimize the network traffic and conserve energy, sensors only send a yes/no notification message to the cluster head when a target is detected ,7) the cluster head remains stationary to save its energy, 8) the cluster head is assumed to be able to estimate the cluster area size from the cluster boundary information; therefore, the minimum number of sensors sufficient to cover the cluster area can be determined, 9) when the number of sensors in a cluster is more than the maximum number of required sensors, the cluster head is capable of initially choosing some sensors as redundant and the redundant sensors in a cluster remain stationary until the relocation phase, and 10) when the number of sensors in a cluster is less than the critical number of sensors, the cluster head includes some virtual sensors to reach the critical number of sensors.

At the beginning of the redeployment phase, each cluster head determines its requirements according the number of sensors belonging to it. As assumed, each cluster head can estimate the cluster area size from available information about the cluster boundary. For example the full coverage of a square area is illustrated in Fig. 1. A sensor of sensing range r_s and a square area with $d \times d$ dimensions are given. The minimum number of sensors $\boldsymbol{n}$ covering the indicated area is determined as follows:

$$n = \left\lfloor \frac{2i^2 + 3i + 1}{2} \right\rfloor \quad (1)$$

where i is a positive integer and should satisfy the following relation:

$$i = \left\lceil \frac{d}{\sqrt{3}r_s} \right\rceil \quad (2)$$

For example, a square area (50 m × 50 m) and a sensor with 6 m of sensing range are given. Therefore, the value of i equals 5 and the minimum number of sensors required (n) is 33 sensors.

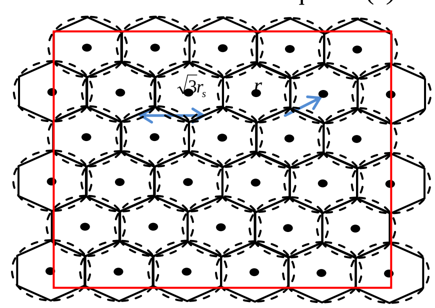

Figure 1: Full Coverage

The CBRA algorithm illustrated in Figure 2 is supported to run on the cluster head. A sensor (i) virtually behaves as a "source of force" for all other sensors. The new logical position is determined based on a virtual force in terms of a determined optimal distance $\sqrt{3}r_s$ and a current distance from a sensor to its neighbor. The optimal distance, as shown in Figure 1, is the distance between two neighbors in case of a full coverage state. The proposed algorithm is designed to push a sensor inside one cluster from a densely region to a sparsely region using the repulsive force. However, when a sensor discovers no communicated sensor closer, no logical movement is determined at the current round. The cluster boundary also exerts a repulsive force when the distance between a sensor and the cluster boundary is less than $\sqrt{3}r_s$. Therefore, the cluster boundary is represented by virtual nodes.

The resultant movement at sensor *i* can be analytically determined as follows. Define as the unit vector that connects S_i and S_j, the movement with respect to sensor S_j, when S_i and S_j are mobile, can be described as follows.

$$\vec{m}_{ij} = \vec{a}_{ij}.\left[\frac{\sqrt{3}r_s - D(s_i, s_j)}{2}\right] \quad (3)$$

where D (S_i, S_j) represents the current distance between S_j and S_j. When D (S_i, S_j) is less than or equal $\sqrt{3}r_s$, each sensor logically moves a half of such distance because each of them is mobile and nothing prevents them to move. On the other hand, when S_j is stationary, i.e. due to lower energy level or being cluster head, the movement can be determined from:

$$\vec{m}_{ij} = \vec{a}_{ij}.\left[\sqrt{3}r_s - D(s_i, s_j)\right] \quad (4)$$

As illustrated in Eq. (4), S_i moves the distance that makes S_i and S_j reach the equilibrium state when the optimal distance is greater than D (S_i, S_j). Finally, the movement $\vec{m}_{ib}$ with respect to the cluster boundary b can be determined from:

$$\vec{m}_{ib} = \vec{a}_{ib}.\left[\frac{\sqrt{3}r_s}{2} - D(s_i, b)\right] \quad (5)$$

As such, the resultant movement of the individual movements can be determined as follows.

$$\vec{M}_i = \sum_j \vec{m}_{ij} \quad (6)$$

CBRA Algorithm

```
Procedure CBRA(Cluster_Head  h) defined as:
   round := 0
    if Cluster.sensors_number  < critical_number then
       Add virtual sensors randomly to reach the critical number
    if Cluster.sensors_number  > maximum_number then
       Choose some sensors as redundant and set them stationary
   do
     round := round + 1
     Broadcast "hello" message at a random time slot to sensor (i)
     if sensor(i).isMobile then
          foreach communicated_sensor(j)
               if D(si, sj) < Dopt(si, sj) then
                         Add sensor(j) to sensor(i).NeighborList
          end for
         Determine the moving direction
         if  moving_step > max_moving_step then
              moving_step := max_moving_step       /* speed control */
        Load sensor(i). position (k-1);
        if oscillation_detected then
             Update moving_step
       Store the sensor(i). position(k)
   while round < specified_max_round
      Cluster_head (h) broadcasts the final distention to sensor(i)
      if Cluster has a virtual position then
          Cluster_head (h) broadcasts the clusters head with request
carrying the virtual node position (x, y)
      if Cluster receives a request (x, y) then
           Apply shortest cascading scheduling from the redundant to
(x, y).
end procedure
```

Figure 2: CBRA Algorithm

At running the CBRA algorithm, we can notice that the sensor's new logical location results in a significant overlap with these undiscovered sensors, the sensor could be pushed back to a position close to its initial position. Ignoring such scenario in this work could cause the sensor to keep oscillating between two or more locations in the field. Sensors oscillation would result in inadequate final destinations that increase the average moving distance. Moreover, the oscillation needs more rounds to reach an adequate coverage level and reduce the overall network performance due to elevated computation cost at the cluster head. In addition, mobile sensors are assumed having uncontrollable speed. This scenario causes that sensors can logically move long distance to reach an adequate coverage level. Consequently, a procedure for oscillation prevention with speed control is proposed to support the CBRA algorithm to achieve low computation cost the cluster head.

In this procedure, each sensor is assumed to store its location from the last round. Figure 3 illustrates the possible movement directions for a sensor movement avoiding oscillation. The figure illustrates the current location of the sensor C, the sensor location in the previous round A, and the possible target locations N.

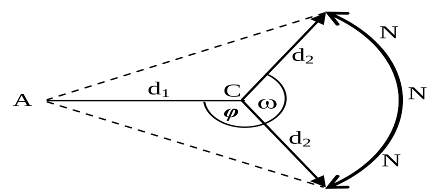

Figure 3: Non-oscillated movement.

The successive two moving steps of the sensor are d_1 and d_2, respectively. The oscillation is eliminated by limiting the angle φ between AC and CN to be more than 60^o and less than 180^o. Thus, to guarantee that the sensor is not oscillating, the sensor movement direction should be constrained by the angle ω. This angle ω varies from (φ-180^o) to (180^o-φ) for the general case. For example, for $\varphi = 90^o$, the distance between A and N equals $\sqrt{d_1^2 + d_2^2}$, and ω varies from -90^o to 90^o. Also, for $\varphi = 120^o$, the distance between A and N equals $\sqrt{d_1^2 + d_2^2 + d_1 d_2}$, and ω varies from -60^o to 60^o. On the other hand, if the new direction of $\vec{M}_i$ does not satisfy ω, a possible oscillation is detected. The logical movement distance for this sensor is gradually reduced in the successive rounds forcing the sensor to reach a stable state.

For this purpose, a counter c is introduced to track the number of oscillations for the sensor (i.e., number of times that the angle ω is violated). Thus, the moving distance can gradually be reduced using the following relation.

$$moving\_step = \frac{\left|\vec{M}_i\right|}{2^c} \quad (7)$$

Initially, when a sensor detects an oscillation at round (k), the oscillations counter increments by one. Thus, the moving step decreases to half of the current step. The oscillation counter remains incrementing as long as the oscillation is still detected. After few rounds, a sensor stops the movement. When the oscillation reason disappears, we reset the oscillation counter to zero. Figure 4 illustrates the pattern of the moving step in a detected oscillation scenario. As shown in figure, as the oscillation counter value increases, the moving step rapidly decreases bringing the sensor to its stable state.

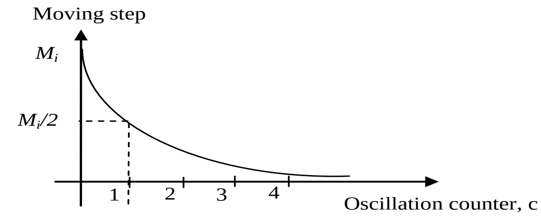

Figure 4: Estimated distance in case of oscillation is discovered.

Finally, when the CBRA algorithm reaches the specified maximum round, the results are examined to obtain the round that achieve the maximum coverage, and then the cluster head sends the final destination to each actual sensor. Simultaneously, when the cluster head has virtual positions, broadcast messages (requests) are sent to the other cluster heads carrying the virtual positions information. On the other hand, when the cluster having redundant sensors, the cluster head may receive a request then the shortest cascading scheduling procedure, discussed below in EARA algorithm, is invoked to fill the virtual positions.

III. EARA ALGORITHM

The energy aware relocation algorithm (EARA) runs at the cluster head as well when a topology change is discovered. The network topology changes when sensor(s) die inside one or more clusters. Two types of coverage are studied, the internal and the external coverage. The internal coverage is performed when a cluster head having redundant sensor(s) discovers sensor failure inside the cluster. On the other hand, the external coverage is performed when a cluster head without or consumed the redundant sensors, it is so called *consumer*, discovers sensor failure and there is other cluster(s) still having redundant sensors, it is so called *supplier*.

The contribution of this algorithm is how efficiently uses the redundant sensors to either internally or externally solve the sensor failure problem. The cluster chooses the redundant sensors which they are prevented to move at the first phase resulting in energy save to the second phase. As a result, the redundant sensors may be exploited to move as far as possible. At the external coverage case, the consumer cluster sends to the closest supplier to provide with the required redundant(s) and repeat the request to the following cluster head until reach sufficient number of sensors. When a supplier receives a request from a consumer carrying the target location, the nearby redundant is chosen and a reply is sent to the consumer carrying the redundant location.

After obtaining the location of the redundant sensor, the customer head determines how to move the redundant sensor to the target location. The simplest way is to move the redundant sensor directly to the target location. However, this scenario may occur at longer distance leading to high energy consumption for the redundant sensor and high relocation delay. The cascading movement is suggested which some intermediate sensors are used to reduce the relocation delay and balance the energy. Assume the movement path looks like a queue and the target location at the front. Therefore, a shift-right is required to move all one place from rear to front.

EARA Algorithm

```
Procedure EARA (Cluster_Head h) defined as:
   Determine its state, supplier or consumer

    if h.isConsumer then
       if action(sensor died) occurs then
           do
             Sends a message (target location) to the closest
Supplier(s)
             Wait for response from supplier (s)
             s := s + 1
           while message is received or s = max. number of clusters
           if message is received then
           Apply shortest cascading scheduling procedure

   if h.isSupplier then
       if action(sensor died) occurs then
           Determine the best redundant (r) to target location(t)
           Apply shortest cascading scheduling procedure internally
       if a message is received from consumer (c) then
           Determine the best redundant (r) to target location(t)
           Sends a message (redundant location) to the consumer (c)
       if h.redundantCount = 0 then
          Change cluster head state into consumer
end procedure
```

Figure 5: EARA Algorithm

We adapt the shortest cascading schedule in [9] in which “hello” messages are used among intermediate sensors carrying the energy information to select the path. That allows the last node in the queue (redundant sensor) to have the full energy while moving. We modify the shortest cascading schedule to exploit the full energy of the redundant sensors in which the distance between the current location and the target location for a redundant sensor is greater than intermediate distances under a constraint of the sensor’s speed. This modification leads to decrease the number of sensors belonging to the path between the source and the distention. Therefore, both of the communication cost and the relocation delay are reduced. Also, the energy of discarded sensors is saved to other operations; as a result, the network lifetime increases. EARA algorithm is illustrated in Fig. 5.

IV. PERFOMANCE COMPARISONS

A simulation tool is developed to examine the effectiveness of the proposed solution described above and compared its performance to VFA [15] and **S**ensor **R**elocation **A**pproach (SRA) [9]. The tool tracks each cluster environment during the different phases, and produces the corresponding coverage performance. This simulation tool is used to perform several experiments to understand the behavior of mobile sensor networks in both phases. In these experiments, the sensors deployment in a field of squared shape is considered. The field dimensions are 60 × 60 meters. Forty-eight sensors are randomly deployed in which this number is chosen to satisfy the Eq. (1) and (2). In fact, while the number of deployed sensors increases above 48, the coverage rather increases to reach the full coverage as shown in Fig. 4. However, the critical number of the deployed sensors is chosen to effectively measure the proposed algorithms performance. The deployed sensors with a sensing range of 6 meters are used to match with other current sensor prototypes, such as Smart Dust (UC Berkeley), CTOS dust, and Wins (Rockwell) [16]. From [17], the current communication range equals 20 meters. The speed of the mobile sensor is 2 m/s. In addition, four clusters are assumed. The simulation interface is shown in Fig. 6 which the grid circles represent the cluster heads, the dotted circles represent the redundant sensors and the blank circles represent the virtual positions.

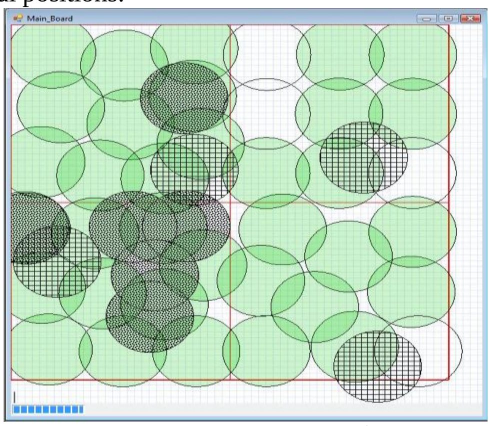

Figure 6: Simulation Tool Interface

The energy consumption per meter is assumed be equal to 27.96 J, similar to the one used in [7]. Each sensor has initial energy 2000 J. With the advent of the TI-MSP430 microcontroller used in the Telos mote [11], an active mode power dissipation of 693µW at 1MHz and 2.2V. A clock frequency of 100 KHz; therefore the instruction time is around 20 ns. The instruction cost relatively equals 1.4 pJ. Hence, the cost of transmitting one bit relatively equals the cost of executing 1000 instructions. Thus, the cost of transmitting one bit equals 1.4 nJ. Assuming the broadcast signal carries 1 kB; therefore, the cost of broadcast signal equals 11.4 mJ. We measure the performance of the proposed algorithms compared with the current work by three metrics: the coverage performance, the average moving distance and the communication (message) cost.

A. Coverage Performance

The CBRA, VFA [15] and SRA [9] run at the redeployment phase with set of different random deployments. We determine the average results of those experiments. As a result, the initial coverage approximately begins at 72%. As shown in Fig. 7, the CBRA algorithm reaches 93.39% because the coverage improvement is internally and externally performed. On the other hand, the VFA reaches 92.24% and the SRA, which the VOR strategy [8] is used, reaches coverage percentage of 92.03%. Consequently, the coverage performance of the proposed algorithm is visibly improved compared with VFA and SRA.

Fig. 8 shows the coverage performance at the second phase (relocation phase). We assume one sensor failure at each cluster, then four sensors are assumed died. The proposed algorithm for the second phase (EARA) begins at the coverage performance of 87.26% due to the dropped sensors. When EARA algorithm runs, the coverage reaches 93.21%.

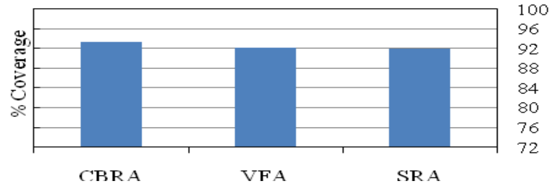

Figure 7: Coverage Performance (%)

This result is rather less than the result at the end of the redeployment phase because the redundant sensors leave small pieces of the monitored field uncovered. On the other hand, the VFA initially drops to 87.4 %, then increases to 91.28% and the SRA initially drops to 87.51 %, then increases to 91.87%.

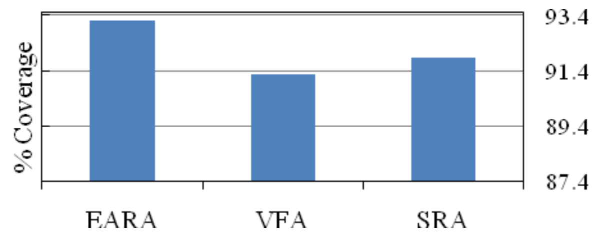

Figure 8: Coverage performance percentage

The coverage performance of SRA is relatively similar to the corresponding of EARA, but the coverage performance of VFA clearly decreases all sensors leave their locations resulting in different gaps to reach the final distention that internally improves the coverage.

B. Average Moving Distance

Fig. 9 shows the average moving for such solutions at both phases. At the redeployment phase, the average distance of the proposed algorithm (CBRA) is the average distance of sensors to achieve the internal coverage in addition to the average distance of sensors to achieve the external coverage and fill the virtual places. Those distances are 3.82 and 0.62 meters, respectively. As shown figure, the proposed algorithm (CBRA) achieves the smallest average distance (4.44 meter) because the oscillation prevention approach reduces the average moving distance as discussed above. Also, Fig. 9 shows the average moving distance at the second phase. The VFA achieves high average moving distance (2.051 meters) because the entire sensors move to improve the internal coverage.

On the other hand, the proposed algorithm (EARA) achieves average moving distance (0.81 meter) rather less than SRA (0.87 meter) because the redundant sensors takes a straight line to reach its target resulting in some reduction in the average moving distance.

C. Message Complexity

Fig. 10 shows the total energy consumption for both phases. At the redeployment phase, as shown figure, the VFA consumes minimum average communication cost (45.6 mJ) because the clustering technique reduces the message complexity. The proposed algorithm (CBRA) is similar to VFA in addition the energy consumed to fill the virtual positions. As a result, the average communication cost for CBRA is approximately (48.54 mJ). On the other hand, SRA consumes (228 mJ) for ten rounds because each sensor autonomously decides its new position and moves physically at each round.

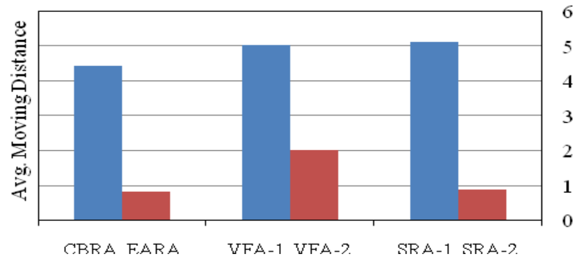

Figure 9: Average Moving Distance (meter)

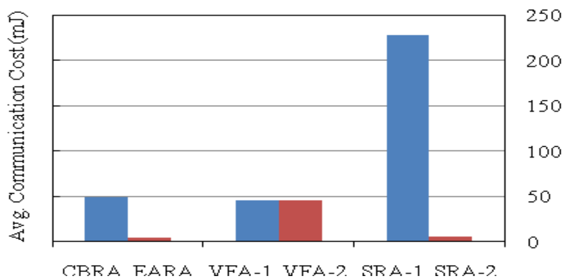

Figure 10: Average communication cost in terms of energy consumpsion (mJ)

At the relocation phase, the VFA consumes the same energy for the average communication cost similar to the first phase (45.6 mJ). On the other hand, some sensors communicate in EARA and SRA to solve the sensor failure problem. Consequently, the average communication cost for EARA and SRA is very small compared with VFA. As mentioned above, the average moving distance for EARA is rather less than SRA because the redundant movement in a straight line decreases the average moving distance. Fig. 10 shows the communication cost in terms of energy consumption at the relocation phase. As shown figure, the average communication cost for EARA and SRA is 4.75 and 5.7 mJ, respectively.

This work assumes the minimum number of sensors is deployed in the monitored field which is computed by Eq. (1) and (2) as shown above. In these experiments, the field dimensions are chosen 60× 60 meters. In fact, when we choose dimensions greater or less than 60 × 60 meters and choose the minimum number of sensors at the sensing range of 6 meters, we approximately obtain the same results illustrated in this work. To verify this fact, a set of experiments are conducted for a squared field with dimensions 120 × 120 meters, sensor's sensing range of 6 meters and sensor's communication range of 20 meters. The minimum number of deployed sensors is 162 according to Eq. (1) and (2). The target field is divided into 16 clusters. The results of those experiments are approximately similar to the results illustrated above.

Finally, the performance comparisons introduced in this section show that the proposed deployment model achieves a high network performance in terms of the coverage percentage, the average moving distance and the communication cost compared with VFA and SRA for the whole network lifetime.

V. Conclusion

A novel model based on clustering technique is introduced in this paper for the deployment of a mobile wireless sensor network. As part of this model, two algorithms are proposed to achieve an efficient deployment scheme that maximizes the overall field coverage at proper average moving distance for the whole network lifetime. Several simulation experiments are performed to examine the efficiency of the designed strategy. Based on the results of these experiments, the proposed solution enhances the sensors distribution in the field which improves the coverage performance.

References

[1]. A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks." Autonomous Robots, 13(2):113-126, 2002.

[2]. A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem." In Proc. of DARS, pp. 299-308, 2002.

[3]. A. Savvides, C. Han, and M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," Proc. ACM MobiCom, 2001.

[4]. D. Niculescu and B. Nath, "Ad Hoc Positioning Systems (APS) Using AoA," Proc. IEEE INFOCOM, 2003.

[5]. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, "Robomote: Enabling Mobility in Sensor Networks," Proc. IEEE/ACM Int'l Conf. Information Processing in Sensor Networks (IPSN-SPOTS '05), Apr. 2005.

[6]. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, pp. 102-114, Aug. 2002.

[7]. G. T. Sibley, M. H. Rahimi and G. S. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), 2002.

[8]. G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment." In Proc. of IEEE INFOCOM, vol. 4, pp. 2469-2479, 2004.

[9]. G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks." In Proc. of IEEE INFOCOM, pp. 2302-2312, 2005.

[10]. H. Liu, Z. Meng and S. Cui, "A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses" Int'l Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007.

[11]. J. Polastre, R. Szewczyk and D. Culler, "Telos: enabling ultra-low power wireless research," Proc. of the 4th int'l symposium on Information processing in sensor networks, 2005.

[12]. M. Srivastava, R. Muntz and M. Potkonjak, "Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problemsolving Environments," Mobicom, 2001.

[13]. N. Alsharabi, Li Ren Fa, Fan Zing, and M Ghurab, "Wireless Sensor Networks of Battlefields Hotspot: Challenges and Solutions," 6th Int'l Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, WiOPT, pp. 192-196, April 2008.

[14]. T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. Saluja, "Sensor Deployment Strategy for Target Detection," First ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[15]. Zou and K. Chakrabarty, "Sensor Deployment and Target Localizations Based on Virtual Forces," Proc. of IEEE Infocom, 2003.

[16]. "Wireless Sensing Networks," http://wins.rsc.rockwell.com, 2004.

[17]. "Berkeley Sensor and Actuator Center," http://www.bsac.eecs.berkeley.edu, 2004.