Sensor Deployment for Composite Event Detection in Mobile WSNs

Yinying Yang and Mihaela Cardei*

Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
yyang4@cse.fau.edu, mihaela@cse.fau.edu
http://www.springer.com/lncs

Abstract. In wireless sensor networks, sensors can be equipped with one or more sensing components. An atomic event can be detected using one sensing component. A composite event is the combination of several atomic events. The monitored area is divided into grids. To achieve high reliability, the composite event must be k-watched in each grid. Otherwise, a detection breach occurs. In this paper, we study the movement-assisted sensor deployment for minimum-breach composite event detection. Given the initial deployment, our goal is to relocate sensors to minimize the breach for all regions. We also impose a limit on the maximum distance that a sensor can move. A centralized approach and a localized algorithm are proposed. Simulation results are presented to validate the performance of our algorithms.

Keywords: wireless sensor networks, composite event detection, sensor deployment, reliability, sensor mobility.

1 Introduction

Sensors are used to monitor and control the physical environment [1]. In mobile sensor networks, sensors can move via springs [2], vehicles [4], and robots [3]. A wireless sensor network (WSN) can detect single (or *atomic*) events or *composite* events [6].

Let us consider a single sensing component first, for example, the temperature. If the sensed temperature value rises above a predefined threshold, we say that an *atomic* event occurred. A *composite* event is the combination of several *atomic* events. For example, the composite event fire may be defined as the combination of the temperature and light. The *composite* event fire occurs only when both the temperature and the light rise above some predefined thresholds.

A large number of sensors can be distributed in mass by scattering them from airplanes, or rockets [1]. In that case, the initial deployment is hard to control. However, a good deployment is necessary to improve coverage or reliability.

* This work was supported in part by the NSF grants CCF 0545488 and CNS 0521410.

Y. Li et al. (Eds.): WASA 2008, LNCS 5258, pp. 249–260, 2008.

In this paper, the square monitored area is divided into grids. To achieve a reliable surveillance, a *composite event* has to be k-watched in each grid (region). Otherwise, a detection *breach* occurs. We propose the movement-assisted sensor deployment for minimum-breach composite event detection (MDCED) problem. Given the initial deployment of a sensor network, our goal is to relocate sensors after the initial deployment such that to minimize the maximum breach among all regions. Sensors are energy constrained devices. To avoid consuming too much energy in moving, we limit the maximum distance a sensor can move.

In our paper, we study reliable composite event detection such that the composite event is k-watched in each region. We propose a centralized integer programming approach and a localized algorithm as solutions for the MDCED problem. We analyze their performance through simulations.

2 Related Works

Recent research works [5][6] focus on sensor collaboration to detect composite events. In [5], a tree is built using a scheme similar to the publish-subscribe communication model. In [6], Vu *et al.* propose an algorithm to construct a set of detection sets satisfying the k-watching requirement in a greedy manner. Different detection sets work alternatively to achieve energy efficiency.

[2][7][10] focus on improving the initial deployment using sensors' mobile ability. In [2], a mechanism based on the minimum cost maximum flow algorithm is proposed to achieve the maximum coverage. In [7], Wang *et al.* present VEC, VOR, and Minimax to maximize the coverage. In VEC, sensors that are too close to each other will be pushed away. In VOR and Minimax, sensors are pulled to the sparser area. Yang *et al.* present a localized method to achieve load balancing in [10]. Regions with underload and overload find a matching between them.

Unlike [5] and [6], which focus on static sensors, this paper focuses on relocating mobile sensors. [2][7][10] study the single event detection, however, the composite event detection in this paper is more complex.

3 Network Model and Problem Definition

The main notations used in the paper are introduced in Table 1. Sensors are randomly deployed in a square monitored area, divided into $r \times r$ grids (see Fig. 1). We assume that there is at least one sensor in each region. Otherwise, a mechanism similar to [8] can be applied, where regions with more than one sensor donate their sensors to regions with no sensors. To ensure sensor coverage (a sensor can monitor the whole region), we choose r such that $r \leq \frac{\sqrt{2}}{2} R_s$, where R_s is the sensing range of the sensor. Each region has a *representative* which takes care of communication with representatives of the neighbor regions and with the sink. It can be chosen according to the sensor's residual energy or contribution. To ensure the communication between adjacent representatives (left, right, top, and bottom), we require that $r \leq \frac{R_c}{\sqrt{5}}$, where R_c is the sensor communication range.

Table 1. Notations

M	The number of atomic events which form the composite event
x_j	The sensing component which detects atomic event j
$n_i^{x_j}$	The total number of sensing components x_j in region i
k	Fault tolerance level
d_{max}	The maximum moving distance of a sensor
$d(i,j)$	The Manhattan distance between region i and region j
R	The total number of regions in the monitored area
P	The number of sensing combinations using one or more sensing components
X_{st}^l	The number of sensors equipped with sensing combination l which move from region s to region t
w_s^l	The number of sensors equipped with the sensing combination l in region s
σ	The number of extra rounds the algorithm could run after the k^{th} round, $\sigma \geq 0$
φ_i	The contribution of sensor i

A sensor can move to its neighbor regions (left, right, top, and bottom). We consider that sensors have limited mobility capabilities due to the energy constraints. d_{max} denotes the maximum Manhattan distance a sensor can move, computed as $\Delta x + \Delta y$. In Fig. 1, the Manhattan distance between regions 2 and 3 is $d(2,3) = \Delta x + \Delta y = 1 + 1 = 2$.

Sensors can have single or multiple sensing components. Taking MTS400 multi sensor board of Crossbow Technology, Inc. [9] as an example, it can sense temperature, humidity, barometric pressure, and ambient light. When we consider a single sensing component, for example, the temperature, if it rises above some predefined threshold, an *atomic event* is detected. A *composite event* is a combination of several *atomic events*. For example, consider a fire-detection application. A *composite event* fire might be defined as a combination of the *atomic events* temperature $> th_1$, light $> th_2$, and smoke $> th_3$, where "th" denotes a threshold for the corresponding attribute. That is $fire = (temperature > th_1) \wedge (light > th_2) \wedge (smoke > th_3)$. It is more accurate to report the fire when all these atomic events occur, instead of the case when only one attribute is above the threshold.

Let us consider that M atomic events $x_1, x_2, ..., x_M$ form the composite event. Sensors are equipped with single or multiple sensing components. Fig. 1 shows an example with $M = 3$. For example, x_1, x_2, and x_3 are temperature, light, and smoke respectively. For a sensor which has only the temperature and light sensing components, we use the set $\{x_1, x_2\}$ to denote its sensing ability.

Sensor nodes may be equipped with different numbers and types of sensing components due to the following reasons [5]: they might be manufactured with different sensing capabilities, a sensor node might be unable to use some of its sensing components due to the lack of memory for storing data, or some sensor components might fail over time. A sensor can be equipped with at most one sensing component of each type. All of a sensor's sensing components turn on or off simultaneously. To achieve a reliable surveillance, an event is required to be

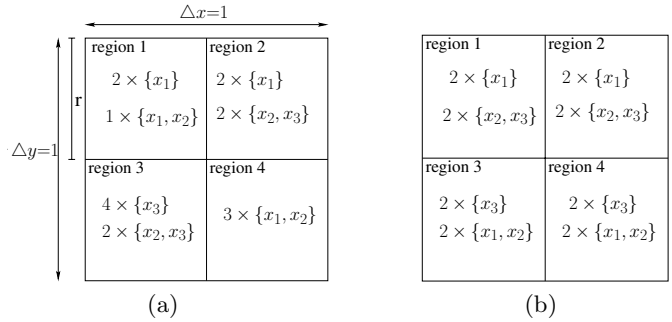

Fig. 1. An example. (a)Initial network deployment. (b)Network deployment after sensor relocation.

observed by more than one sensor. We define the k-watched atomic/composite event [6]:

Definition 1 (k-watched atomic event). *An atomic event is k-watched by a set of sensors if at any time this event occurs at any point within the interested area, at least k sensors in the network can detect this occurrence.*

Definition 2 (k-watched composite event). *A composite event is k-watched by a set of sensors if every atomic event part of the composite event is k-watched by the set of sensors.*

In this paper, the objective is that the composite event is k-watched by the sensors of each region. Due to a random initial deployment of the WSN, some regions will not provide the k-watching property. We define the *breach* of a region as the maximum gap in achieving the k-watching of an atomic event.

$n_i^{x_j}$ denotes the total number of sensing components x_j in region i. The *breach* of the region i is defined as:

$$breach_i = \max(0, k - n_i^{x_1}, k - n_i^{x_2}, ..., k - n_i^{x_M}) \quad (1)$$

To achieve a good quality in monitoring the composite event, the objective is to minimize the maximum breach for all the regions in the monitored area. In the initial deployment, some regions might have a higher breach, while others might have additional sensors. We propose to use sensor movement to relocate sensors such that to minimize the maximum breach in the network.

Definition 3 (Movement-assisted sensor Deployment for Composite Event Detection in wireless sensor networks - **MDCED** problem). *Given a WSN with N sensors randomly deployed in a square area which is partitioned into $r \times r$ grids, and given the maximum sensor moving distance d_{max}, design a sensor moving strategy such that to minimize the maximum breach for all the regions in the network, that is $MAX_{i=1..R} breach_i = MINIMUM$.*

Fig. 1 shows an example with $k = 2$, $M = 3$, and $d_{max} = 1$. This means that each of the three atomic events has to be 2-watched in each region and

the maximum moving distance of a sensor is 1 (each sensor can only move to a neighboring region).

In this example, the monitored area is divided into four regions and there are four sensing combinations $\{x_1\}$, $\{x_1, x_2\}$, $\{x_2, x_3\}$, $\{x_3\}$. Fig. 1a shows the initial sensor deployment. Region 1 has two sensors $\{x_1\}$ and one sensor $\{x_1, x_2\}$, region 2 has two sensors $\{x_1\}$ and two sensors $\{x_2, x_3\}$, region 3 has four sensors $\{x_3\}$ and two sensors $\{x_2, x_3\}$, and region 4 has three sensors $\{x_1, x_2\}$. The maximum breach in the initial deployment is 2, occurring in the regions 1 (no sensing component x_3), 3 (no sensing component x_1), and 4 (no sensing component x_3).

Our goal is to relocate sensors such that to minimize the maximum breach for all regions. A possible deployment after sensor repositioning is shown in Fig. 1b. The moving strategy is: two sensors $\{x_2, x_3\}$ move from region 3 to region 1, one sensor $\{x_1, x_2\}$ moves from region 1 to region 3, two sensors $\{x_3\}$ move from region 3 to region 4, and one sensor $\{x_1, x_2\}$ moves from region 4 to region 3. The total number of sensor movements is 6 and every region has breach 0 (or no breach) after sensors relocation.

4 Solutions for the MDCED Problem

4.1 Centralized Integer-Programming Approach

We use P to denote the number of sensing combinations deployed, containing one or more sensing components which can watch atomic events of interest. In the worst case, $P = \binom{M}{0} + \binom{M}{1} + ... + \binom{M}{M} = 2^M$. However, in practice, a fewer number of combinations might be deployed. In our example in Fig. 1, there are $P = 4$ sensing combinations: $\{x_1\}$, $\{x_1, x_3\}$, $\{x_2, x_3\}$, and $\{x_3\}$ instead of $2^3 = 8$.

The objective is to compute the number of sensors with sensing combination l ($1 \leq l \leq P$) which move from a region s to a region t, $X_{st}^l \in \{0, 1, ..., w_s^l\}$, when $d(s,t) \leq d_{max}$. If the Manhattan distance between the regions $d(s,t) > d_{max}$, then $X_{st}^l = 0$. Note that X_{ss}^l represents the number of sensors with sensing combination l which do not leave region s. We denote w_s^l the original number of sensors equipped with sensing combination l in region s. The optimal solution is modeled using the following IP formulation:

$$\text{Minimize} \quad \underset{1 \leq i \leq R}{MAX}\ breach_i$$

$$\text{subject to} \sum_{t=1}^{R} X_{st}^l = w_s^l \text{ for } 1 \leq s \leq R, 1 \leq l \leq P$$

$$X_{st}^l = 0 \text{ for } d(s,t) > d_{max}, 1 \leq s, t \leq R, \text{ and } 1 \leq l \leq P$$

$$\text{where} \quad X_{st}^l \in \{0, 1, ..., w_s^l\} \text{ for all } 1 \leq s, t \leq R, \text{ and } 1 \leq l \leq P$$

Remarks:

- The objective function asks to minimize the maximum breach for all regions i, $1 \leq i \leq R$.
- The first constraint requires that the number of sensors with sensing combination l that leave or stay in region s equals the total number of sensors with sensing combination l originally in region s, w_s^l.
- The second constraint ensures that no sensor moves from a region s to a region t if the Manhattan distance between them is greater than the maximum distance d_{max}. This constraint guarantees that the total movement distance of a sensor does not exceed d_{max}.

The value $breach_i$ in the IP objective function is computed using formula 2. It compares the number of each sensing component the region has with k to get the breach for each sensing component and then finds the maximum one as the breach of the region.

$$breach_i = \max(0, k - \sum_{x_1 \in l, l=1}^{P} \sum_{j=1}^{R} X_{ji}^l, \ldots, k - \sum_{x_M \in l, l=1}^{P} \sum_{j=1}^{R} X_{ji}^l) \qquad (2)$$

The updated values of the number of sensing components x_u in region i are computed as $n_i^{x_u} = \sum_{x_u \in l, l=1}^{P} \sum_{j=1}^{R} X_{ji}^l$.

X_{st}^l are the only variables in the IP formulation, thus the total number of variables is R^2P. We use CPLEX solver to implement the IP approach.

After getting the optimal maximum breach, we can use other integer programming formulations to optimize the number of movements and the moving distance. The objective function for optimizing the number of movements is $\sum_{l=1}^{P} \sum_{i=1}^{R} \sum_{\substack{j=1 \\ i \neq j}}^{R} X_{ij}^l$. For optimizing the moving distance, the objective is $\sum_{l=1}^{P} \sum_{i=1}^{R} \sum_{\substack{j=1 \\ i \neq j}}^{R} (d(i,j) \times X_{ij}^l)$. They have the same two constraints with the previous IP formulation and there is the third constraint, which is $breach_i \leq optBreach$ for $1 \leq i \leq R$, where $optBreach$ is the optimal maximum breach.

The IP algorithm is executed as follows. The representative of each region communicates with all the sensors in the region and transmits region location and w_i^l values to the sink. The sink uses an IP-solver to compute the sensors' movement plan, e.g. X_{st}^l values and then inform region representatives. The representatives coordinate sensors movement inside their regions.

4.2 Localized Algorithm

The localized solution is organized in rounds and incrementally reduces the maximum breach in the network. The algorithm runs $k + \sigma$ rounds, where $\sigma > 0$ is a tunable parameter. The regions that initiate the mechanism to reduce their breach in a round are called *initiator regions*, or simply *initiators*.

After the initial deployment, the maximum breach in the network is up to k. In the first round, the initiators are the regions with breach k. At the end of this round, depending on the network characteristics, there might be initiators

Algorithm 1. Localized Method - Initiator Region

```
1: Wait time T
2: Broadcast Request message using TTL to limit the number of hops
3: if Reply messages received then
4:     Decide which sensors to take
5:     Broadcast ACK message to candidate regions
6: end if
7: if receive sensors from candidate regions then
8:     Update the breach
9: end if
```

that could not decrease their breach. In the second round, the initiators are the regions with breach k or $k-1$. More generally, in the $h^{th}(h \leq k)$ round, the initiators are the regions with breaches greater than or equal to $k+1-h$. In the rounds $k+1$ ... $k+\sigma$, the initiators are all the regions with breaches greater than 0. A higher priority is given in reducing higher breaches.

In general, there are two ways to reduce the breach of an initiator. When a region has transferable sensors, then it can directly assign sensors to the initiator. Otherwise, if no sensors can be moved but there are transferable sensing components, the region first tries to get transferable sensors by exchange and then assigns transferable sensors to initiators. The first method has higher priority since fewer sensor movements are involved.

In a round, each initiator has zero or more *candidate regions*. A candidate region must have the breach less than the initiator's breach and must satisfy condition 1 and one of the conditions 2 or 3:

1. Being located at a distance less than or equal to d_{max} from the initiator, since sensors' maximum moving distance is upper-bounded by d_{max}.
2. *First class candidate region*: have transferable sensors equipped with one or more of the initiator's *key sensing components* (sensing components with maximum breach in the initiator). A sensor is *transferable* if moving that sensor from the candidate region will keep the breach of the candidate region less than the initiator's breach. Our goal is to minimize the maximum breach in every round, and therefore sensor movements must not generate a new higher breach.
3. *Second class candidate region*: have one or more transferable key sensing components. The candidate may get transferable sensors through exchanging sensors with other regions. In this case, the breach of the candidate will not drop as result of the exchange.

In each round, the algorithm consists of a negotiation process between initiators and their candidates. The process is started by the initiators and after the negotiation process, sensors are moved. The main steps of a round in the algorithm are presented in Algorithms 1, 2, and 3.

In each round, initiators use controlled flooding to send *Request* messages, in an attempt to reduce their breaches. An initiator waits a time T before sending the request in order to reduce collisions, avoiding adjacent neighbors sending

Algorithm 2. Localized Method - Candidate Region

```
1: if first class candidate region then
2:   if Request message received then
3:     Decide the allotment
4:     Send back Reply message to the initiator
5:     if TTL > 1 then
6:       TTL = TTL − 1
7:       Forward the request message
8:     end if
9:   end if
10:  if ACK message received then
11:    Move required sensors to the initiator
12:  end if
13: end if
14: if second class candidate region then
15:  if Request message received then
16:    if TTL > 1 then
17:      Attach exchange information to the request message
18:      TTL = TTL − 1
19:      Forward the request message
20:    end if
21:  end if
22:  if Reply message received then
23:    Record the exchange region
24:    Send Reply message to the initiator
25:  end if
26:  if ACK message received then
27:    Exchange sensors and send the corresponding transferable sensors to the initiator
28:  end if
29: end if
```

requests at the same time. The value of T is computed based on the initiator's breach and a small random number. In general, the higher the breach is, the smaller the value of T is.

The request message generated by initiator i has the format: *Request* ($RID = i, n_i^{x_1}, n_i^{x_2}, ..., n_i^{x_M}, breach, TTL = d_{max}$). RID is the initiator's identifier. $\{n_i^{x_1}, n_i^{x_2}, ..., n_i^{x_M}\}$ are the numbers of each sensing component the initiator has and they also indicate which types of sensing components are needed. *breach* is the breach of the initiator and TTL (Time-To-Live) is used to control the number of hops the message is forwarded. Every intermediate region that receives the message for the first time decreases the TTL by 1 and forwards the message only if $TTL \geq 1$.

When a second class candidate receives the request, it asks for exchanging sensors. Consider the case when a request asks for the key sensing component $\{x_1\}$ and the candidate has a sensor $\{x_1, x_2\}$ which can not be directly assigned to the initiator since sensing component x_2 is still needed by the candidate. If the candidate region can exchange the sensor $\{x_1, x_2\}$ for one sensor

Algorithm 3. Localized Method - Non-candidate Region

```
1: if Request message received then
2:    Check the exchange information
3:    if can make the exchange then
4:       Send a Reply message to the candidate who asks for the exchange
5:       Remove exchange information from the request message
6:    end if
7:    if TTL > 1 then
8:       TTL = TTL − 1
9:       forward the request message
10:   end if
11: end if
```

$\{x_1\}$ and another sensor $\{x_2\}$, then it can give the initiator the transferable sensor $\{x_1\}$. The candidate region asks for an exchange by attaching the fields: $\{RID, keyComp, otherCompList, EXCHAGE\}$ to the end of the request message, where RID is the identifier of the exchange requestor, $keyComp$ is the key sensing component, and $otherCompList$ is the list of the other sensing components in the sensing combination and $EXCHAGE$ shows the type of the movement. The candidate region decreases TTL in the request by 1 and forwards the message if $TTL \geq 1$.

When a first class candidate receives a request message, it follows the following steps to decide whether it will give one or more sensors to the initiators.

1. A region may receive more than one request messages. The candidate region i computes a priority for each request it receives and sorts them in decreasing order, assigning first sensors to the initiators with higher priorities. For a request from the initiator region j, the priority is the initiator's breach value, taken from the request message. The Manhattan distance between these two regions, which can be computed from the TTL value in the request message, is used to be the second criterion to break the tie. A shorter distance has higher priority.
2. The contribution $\varphi_i = \frac{hf_i}{sc_i}$ of each transferable sensor i is computed, where hf_i is the number of helpful sensing components in the transferable sensor i and sc_i is the total number of sensing components sensor i has. Note that the transferable sensor must contain at least one key sensing component of the initiator. Otherwise, it can not reduce the breach of the initiator. For example, if the initiator's request message has breaches for the sensing components x_1 and x_2, then the contribution of a sensor $\{x_1, x_3\}$ is $\frac{1}{2}$, since only $\{x_1\}$ is helpful for this request and the sensor has 2 sensing components in total. The contribution of a sensor $\{x_1\}$ is 1.
3. A candidate region addresses the requests in the sorted order. It computes the contribution of each transferable sensor, sorts them in decreasing order and assigns the first $\lfloor \delta \cdot N_{transfer} \rfloor$ sensors to the request. δ is an input parameter and $N_{transfer}$ is the number of transferable sensors.
4. After having made the decision, the candidate sends back *Reply* messages to the initiators to whom sensors have been assigned. A reply message has

the form *Reply* (RID_1, RID_2, *sensorList*, *breach'*, *dist*, $GIVE$), where RID_1 is the candidate region identifier and RID_2 is the initiator's identifier. *sensorList* is the list of sensors which can be assigned to the initiator. The breach of the candidate may change due to the assignment. *breach'* is the new breach and *dist* is the distance between the candidate region and the initiator region. $GIVE$ means the sensors come from a first class candidate. The reply is a unicast message sent along the reverse path established when the request message was sent.

5. If the TTL in the received request message is greater than 1, it decrements the TTL by 1 and forwards it. If the received request message contains an exchange request whose requested sensing components have been assigned by the candidate region, then the exchange information is removed from the request message.

When an intermediate region which is not a candidate (Non-candidate Region) receives a request message, it checks the attached exchange information if there is any. If it can make the exchange, it sends back a *Reply* message to the exchange requestor. The message has the form *Reply* (RID_1, RID_2, *keyComp*, *otherCompList*, $EXCHAGE$), where RID_1 is the identifier of the region sensing the reply message, RID_2 is the identifier of the exchange requestor, *keyComp* and *otherCompList* are the sensing components that can be exchanged. Then the request message is updated (TTL is decremented by 1 and the exchange information is removed) and forwarded if $TTL \geq 1$.

When the exchange requestor receives the reply message, it updates the reply message with its own transferable sensor information and forwards it to the corresponding initiator.

An initiator may receive multiple reply messages from several candidate regions. It follows the following steps to decide which sensors to take from the first class candidates and then, if the breach is still greater than 0, the initiator considers taking sensors from second class candidates.

1. It computes the contribution $\varphi_i = \frac{hf_i}{sc_i}$ for each sensor i in the *sensorList* of the message.
2. The initiator considers sensors from *sensorList* of the candidates in increasing order of the *breach'* values first and then in the decreasing order of the sensor's contribution to break the tie.
3. The initiator considers taking sensors as long as its breach can be decreased and is greater than 0.
4. The initiator sends one ACK message indicating the movement plan (number and types of sensors it will take from each candidate). The ACK message is sent using localized flooding with TTL equal to the distance between the initiator and the farthest candidate with sensors in the movement plan. The sensor reservation made by other candidates will expire if they are not included in an ACK message.

When first class regions receive ACK messages, sensors are actually moved to the initiator. When second class regions receive ACK messages, they first exchange sensors and then move the corresponding transferable sensors to the initiator.

5 Simulations

Simulation environment: Metrics in the simulations include the breach, which is computed according to formula 1 in the localized algorithm and formula 2 in IP approach, the total moving distance and the total number of movements, which are metrics to measure the energy consumption in moving.

We conduct the simulations on a custom discrete event simulator, which generates a random initial sensor deployment. All the tests are repeated 50 times and the results are averaged. In the simulations, there are 6×6 grids in the monitored area and $M = 3$, $k = 3$.

Simulation results: In Fig. 2, $d_{max} = 3$, and $\sigma = 2$, which means that the localized approach runs 5 rounds. Fig. 2a measures the maximum breach among all regions after applying integer programming and localized approaches. It shows that the localized algorithm gets close results with the IP approach, and they improve the initial deployment in terms of the maximum breach. In Fig. 2b and Fig. 2c, the curves for the integer programming are both optimal results using integer programming formulations discussed in section 4.1. When there are 150 sensors in the network, the number of movements and moving distance of the localized method is the highest. It is because when there are only 100 sensors, the density is too low to decrease the breach, and in this case, there are

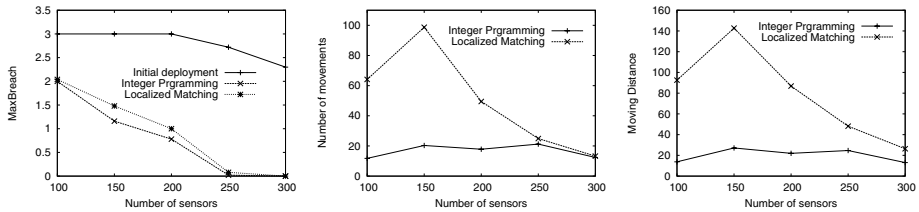

Fig. 2. Comparisons between two algorithms (a)Maximum breaches. (b)The total number of movements. (c)The total moving distance.

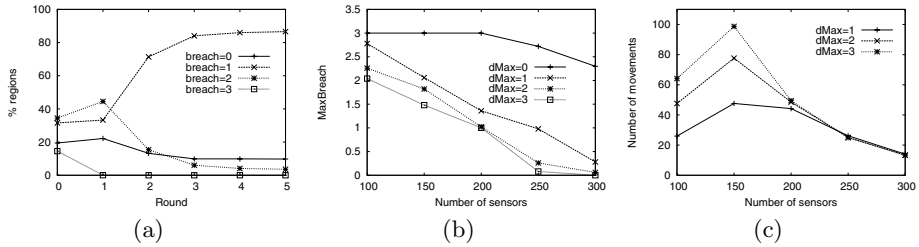

Fig. 3. Localized approach (a)Maximum breaches in every round. (b)Maximum breaches using different d_{Max}. (c)The total number of movements using different d_{Max}.

fewer movements and shorter moving distance. When the density is high enough, sensors do not have to move much to achieve the lower breach.

In Fig. 3, we study the localized method. Fig. 3a shows the percentage of regions with each breach value for each round. Round 0 shows the initial deployment. With running of the algorithm, the number of regions with breach 3 decreases. No region has breach 3 starting with the second round. In the end, most regions have breach 1 and there are a small number of regions having breach 2. In Fig. 3b and Fig. 3c, d_{max} varies from 1 to 3. In Fig. 3b, the curve, $d_{max} = 0$, shows the maximum breaches in the initial deployment. When $d_{max} = 3$, it has the lowest maximum breach. When $d_{max} = 1$, it achieves the highest maximum breaches, however, it still improves the initial deployment. When the density is low, $d_{max} = 3$ can achieve lower breach and consequently, the number of movements is larger. When the density is high, e.g., 250 and 300, $d_{max} = 3$ moves less compared with other two cases.

6 Conclusions and Future Work

In this paper, we focus on the Movement-assisted sensor Deployment for Composite Event Detection in wireless sensor networks (MDCED) problem. The integer programming approach and localized method are proposed. Simulation results show that the localized method gets close maximum breach results with the integer programming approach. In our future work, we will continue to study additional methods for the MDCED problem.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer Networks 38(4), 393–422 (2002)
2. Chellappan, S., Bai, X., Ma, B., Xuan, D., Xu, C.: Mobility limited flip-based sensor networks deployment. IEEE Transactions of Parallel and Distributed Systems 18(2), 199–211 (2007)
3. Dantu, K., Rahimi, M.H., Shah, H., Babel, S., Dhariwal, A., Sukhatme, G.S.: Robomote: enabling mobility in sensor networks. In: IPSN 2005, pp. 404–409 (2005)
4. Lee, U., Magistretti, E.O., Zhou, B.O., Gerla, M., Bellavista, P., Corradi, A.: Efficient data harvesting in mobile sensor platforms. In: PerCom Workshops, pp. 352–356 (2006)
5. Kumar, A.V.U.P., Reddy, A.M.V., Janakiram, D.: Distributed collaboration for event detection in wireless sensor networks. In: MPAC 2005 (2005)
6. Vu, C.T., Beyah, R.A., Li, Y.: Composite event detection in wireless sensor networks. In: 26th IEEE IPCCC (2007)
7. Wang, G., Cao, G., LaPorta, T.F.: Movement-assisted sensor deployment. IEEE Transactions on Mobile Computing 5(6), 640–652 (2006)
8. Wu, J., Yang, S.: SMART: a scan-based movement-assisted sensor deployment method in wireless sensor networks. IEEE INFOCOM, 2313–2324 (2005)
9. `http://www.xbow.com/Home/HomePage.aspx`
10. Yang, S., Wu, J., Dai, F.: Localized movement-assisted sensor deployment in wireless sensor networks. In (WiMa) (in conjunction with IEEE MASS 2006) (2006)