Two Centralized Energy-Efficient Deployment Algorithms for Mobile Nodes in a Mixed Wireless Sensor Network

Zi-Qi Hao[1] Zhen-Jiang Zhang[1*]

[1] Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University

Beijing, 100044, China

{09211079, zhangzhenjiang}@bjtu.edu.cn

Abstract. The coverage problem is a fundamental issue in wireless sensor network (WSN). As the sensor nodes of WSN has limited energy, energy saving becomes important in increasing the lifecycle of the network. Constructing a sensor network with a mix of mobile and static sensors can achieve a balance between sensor coverage and sensor cost. In this paper, we design two centralized deployment algorithms of sensors in terms of the coverage problem to achieve the goal of energy-efficient coverage, and thus to enhance the lifecycle of mixed WSN: The Hungarian method-based algorithm (HBA) and the threshold-based algorithm (TBA). The former algorithm can minimize the overall moving distance of mobile sensors. And the latter algorithm is an improvement of the former, which can do better in load-balance. We compared the two algorithms by simulation and discussed which algorithm is better in prolonging the lifecycle of network.

Keywords: WSN, energy-efficient algorithm, coverage

1 Introduction

The emerging technology of wireless sensor network (WSN), which consist of a large number of sensing nodes each capable of sensing, processing and transmitting environmental information, is expected to provide a broad range of applications, such as battlefield surveillance, environmental monitoring, smart spaces and so on[1].

In most early research, sensor nodes are assumed to be static so that a large number of redundant nodes are deployed to achieve a desired level of coverage. This may introduce high cost and still cannot guarantee coverage. With the emergence of mobile sensors, which are able to move from densely covered areas to sparsely covered areas to achieve balanced coverage, the research of movement strategy of mobile nodes becomes a hot topic.

Various algorithms and protocols [2-4] have been proposed to guide the mobile sensors moving from the original position to the proper position. All of the moving strategies mentioned above are designed for the mobile networks, where all the sensing nodes are capable of mobility. But in most cases, it’s unnecessary to equip every sensor with a motion when the coverage requirement is not very strict. We only need a few mobile nodes in order to save the network cost. We call a networks consist of both mobile nodes and static nodes a *mixed network.*

Energy consumption is of critical importance in WSN because the sensors are typically battery powered and once deployed, they are usually unattended. The energy consumption of mobile sensors includes two parts, mechanical movement and communication. In reality, the moving distance consumes more energy than communication, which is crucial in prolonging the lifecycle of network. Wang et al. [5] designed two distributed bidding protocols for the placement of mobile sensors in a mixed network: a basic protocol and a proxy-based bidding protocol. The two protocols have achieved the same high coverage which is quite ideal, and the proxy-based bidding protocol has some progress in reducing the moving distance of mobile sensors by virtual movements and coverage hole exchange.

However, previous research failed to find an optimal solution to make the overall moving distance minimum as the limitation of the distributed algorithm, for a node can only get the information surrounding itself as the communication radius is limited. Further studies to find an algorithm more Energy-Efficient are still necessary.

In contrast with the distributed algorithm, the centralized algorithm can achieve global information of the whole network by a sink node, which is superior to the distributed algorithm when calculating a global optimal solution. In this paper, we proposed using a mix WSN which consists of both mobile and static sensors, we proposed two centralized energy-efficient algorithms to dispatch mobile nodes to the coverage holes, thus to enhance the lifecycle of the mixed WSN, namely are the Hungarian method-based algorithm (HBA) and the threshold-based algorithm (TBA). HBA can achieve an optimal match between the mobile sensors and the coverage holes, which can minimize the overall distance of mobile sensors. Take the load-balance into account, we proposed TBA, which can make the longest moving distance of mobile sensors minimum. We also compared the two algorithms by simulation. The main contributions of the paper are summarized as follows:

1) In terms of the energy-efficient problem in the coverage problem, we designed a centralized algorithm

HBA to dispatch mobile sensors to the optimal destinations.

2) Another algorithm TBA is proposed considering the load-balance.

3) We evaluated the performance of two proposed energy-efficient algorithms by simulation and gave the conclusion which algorithm is better in different situations.

The rest of the paper is organized as follows: we present our related work in Section 2. Section 3 is some preliminaries before we introduce our centralized energy-efficient algorithms. Section 4 gives the centralized energy-efficient algorithm overview and detail steps of the Hungarian method-based algorithm (HBA) and the threshold-based algorithm (TBA). In Section 5, we evaluate the performance of the two algorithms by simulation. We conclude the paper in Section 6.

2 Related Work

In this section, we review some related work on our research in the field of coverage improvement and efficient energy conservation.

Y.-C. Wang et al. [2], [6] propose the coverage pattern based movement, the target locations for mobile nodes are computed based on a predefined coverage pattern that can meet both area coverage and network connectivity requirements. The authors in [3] propose a virtual force algorithm (VFA) as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors, the overall force specifies the direction and distance that a mobile node should move to. In the grid quorum based node movement strategies [4], [6] The sensor field is partitioned into many small grid cells, and the number of nodes in each cell are considered as the load of the cell, which can be viewed as a load balancing problem in traditional parallel processing systems. All of the moving strategies mentioned above are designed for the mobile networks.

The authors in [7] discuss three distributed self deployment algorithms: VEC, VOR, and Minimax to calculate the target locations of the mobile nodes. In VEC, sensors move away from a dense area; in VOR, sensors migrate towards holes; in Minimax, sensors also move towards holes, but more conservatively with the consideration of not generating new holes. The same authors in [5] propose two bidding protocols for the mix WSN. Their algorithms consider a random initial deployment, where static nodes detect the coverage holes by a Voronoi diagram and bid for the mobile nodes to move to the farthest Voronoi vertex. However, the distributed algorithms failed to achieve the information of the whole network, and sometimes it needs several rounds of calculation to decide the final target locations, so energy wastage of sensors still existed. In this paper, we design two centralized algorithms for the placement of mobile sensors in a mixed network to better reduce the energy waste and prolong the life cycle of the network.

Many previous work focused on the energy-efficient problem of WSN had created energy consumption models to analyze the performance of their proposed protocols or algorithms. Guiling Wang in [5] classified the energy consumption into two parts, namely the mechanical movement and communication, using message complexity and normalized moving distance to measure the energy consumption in communication and movement respectively. C-P Chen in [8] suggested an energy consumption model considering the path loss of radio transmission, which is expressed by the dissipated energy for the transmit circuitry or receives circuitry per bit and the package size. In reality, mechanical movement of sensor nodes is the dominant factor in energy consumption [5], and this paper aims at deployment algorithms of mobile sensors, so we only consider the moving cost of sensors, energy consumption for the other operations such as message transmission, processing, and storage are not taken into account.

3 Theoretical Framework

3.1 Problem Statement and assumptions

We deploy a certain number $N(s_i)$ of static nodes and $N(m_j)$ of mobile nodes randomly in a fixed rectangular sensing area. What we should do is to allocate $N(m_j)$ of mobile nodes to the proper locations, which can prolong the lifecycle of the whole network and make sure the coverage rate is up to standard at the same time. We assume an isotropic sensing model in which the sensing area of each sensor node is represented by a circle with the same radius. Anywhere inside the sensing area can be sensed without difference. All sensor nodes know their locations, we assume to use the techniques in [9] to solve this problem. We also assume that each static sensor needs only one mobile sensor to heal the hole. We will use a Voronoi diagram to detect the coverage holes. Aiming at the energy-efficient problem and the load-balance problem, our objective is to solve the following problems:

1. How to dispatch the $N(m_j)$ mobile sensors to the $N(s_i)$ coverage holes that can minimize the overall moving distance of mobile sensors and make sure the coverage rate is qualified at the same time.
2. If we solved problem 1, what about the circumstance when one of the sensors is penalized (the sensor moved far longer distance than others).
3. If we can find strategies to solve the former two problems, which one is better in prolonging the lifecycle of the network.

3.2 Voronoi Diagrams

In this paper, we use a *Voronoi diagram* to detect coverage holes. Voronoi diagrams are well known structures in computational geometry. In 2-dimension, the Voronoi diagram of a set of discrete points divides the plane into a set of convex polygons according to the nearest neighbor rule: all points inside a polygon are closest to only one point. The divided convex polygons are called *Voronoi cells*. Fig. 1 is a Voronoi diagram formed by eight sensors in the plane. We assume using the algorithms introduced in [10] to form a Voronoi diagram.

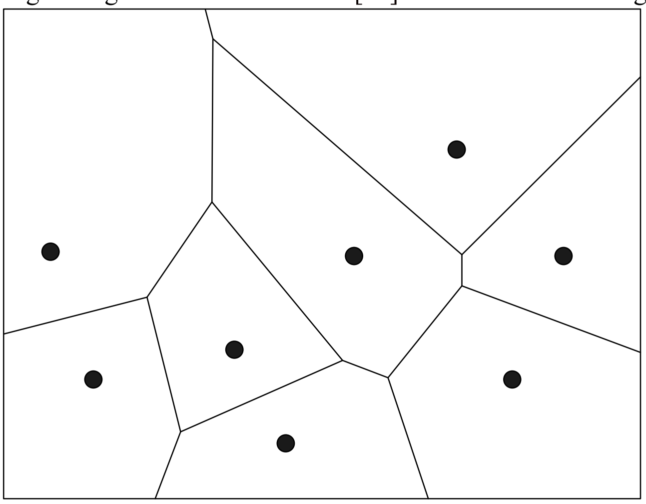


Fig. 1. Voronoi diagram formed by 8 sensors

4 Centralized Energy-efficient Algorithm Overview

In this section, we give an overview of our centralized energy-efficient algorithm. We use a *sink node* to get the global information of the network.

In the initialization phase, we form a Voronoi diagram with all the static nodes, and every static node inside its Voronoi polygon calculates the hole size and the target location of the coming mobile node, then send the target locations and hole sizes to the sink node, and the mobile sensors send their locations to the sink node as well. After initialization phase, we will determine value coefficient matrix whose elements presents the distances of $N(m_j)$ mobile nodes and $N(s_i)$ coverage holes respectively. In the core part of two algorithms, we will match the $N(m_j)$ mobile nodes to the $N(s_i)$ coverage holes by the method of Hungarian and threshold respectively. Finally, we dispatch the mobile nodes according to the calculations results. The following is exact method of determining target location and hole size calculation.

4.1 Determine target location

We form a Voronoi diagram with all the stationary nodes. According to the property of a Voronoi diagram, all points are closest to only one node inside a Voronoi cell. Therefore, if some areas of a voronoi cell are not covered by its generating node, these areas will not be covered by any other sensor and thus a coverage hole is generated. As illustrates in Fig. 2, V_1 and V_4 are not covered by the coverage area of S_1, and thus some coverage holes exist around sensor S_1.

In this paper, we assume that every static node needs only one mobile node to heal its hole. If there exists a coverage hole, the static sensor chooses the farthest Voronoi vertex as the target location of the coming mobile sensor. Take Fig. 2 for example, to the static node S_1, V_1 is the farthest Voronoi vertex, so we choose V_1 as the destination of the coming mobile node. However, there's circumstance when the two or three neighbor static sensors share the same farthest Voronoi vertex, we propose to use the greedy method to determine all the target locations.

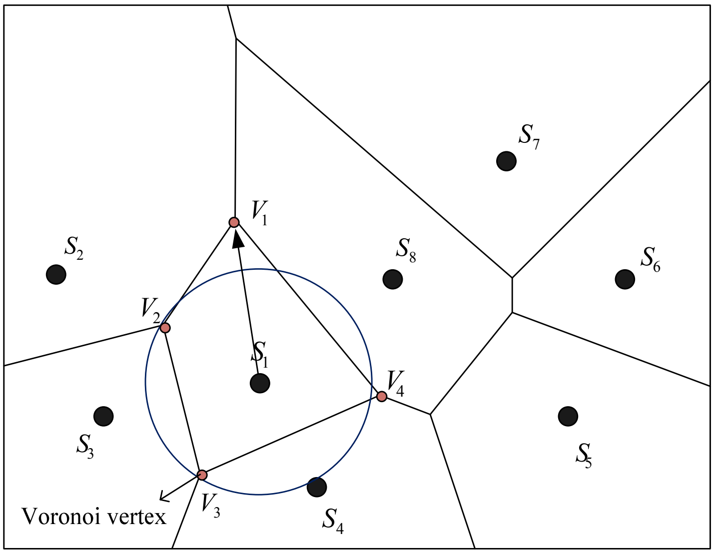


Fig. 2. Hole detection by a Voronoi diagram

4.2 Hole size calculation

We choose a rough calculation of the hole size: $\pi(d-R_s)^2. d \geq R_s$. Where d is the distance between a static sensor and its farthest Voronoi vertex. R_s represents the unified sensing range of a senor. As shown in Fig. 3, O is the farthest Voronoi vertex of sensor S_1, the shadow area in the smaller circle around O is estimated as the size of the hole generated by S_1.

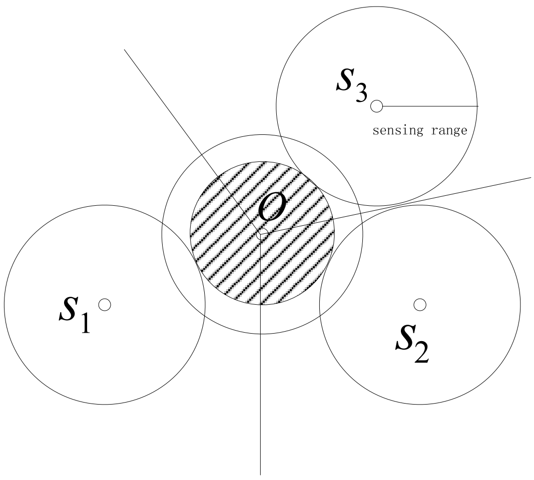


Fig. 3. Hole size estimation

5 The Hungarian Method-based Algorithm (HBA)

The Hungarian method-based algorithm (HBA) solved the following problem: How can we dispatch the $N(m_j)$ mobile sensors to the $N(s_i)$ coverage holes to ensure the overall distance is the shortest under the premise of the coverage rate is qualified? We will discuss this problem in the following two conditions: First, situation when the number of mobile sensors $N(m_j)$ equals the number of static nodes $N(s_i)$; Second, when $N(m_j)$ is less than $N(s_i)$.

At first, we consider the simple case when $N(m_j)$ equals $N(s_i)$. The mathematical model of this matching problem can be attributed to the assignment problem. The assignment problem can be described as follows: there are n tasks to be done by n persons, each person can only do one task and each task can be done by only one person, but the efficiency and capacity of each person is different, how to assign the n people to do the n tasks to gain the highest efficiency (or the shortest time) is an assignment problem. An effective method to solve the assignment problem is the Hungary method in the operational research.

We take a simple example to illustrate the Hungarian method. Assume the initial *value coefficient matrix* C_{ij} (the elements of which represent the value coefficient, such as cost, time or efficiency. In this paper, they represent the distances of each mobile sensors to move when it match to different coverage holes.) is as follows, the dimension of the matrix n=5. The operation process is as follows:

$$\begin{bmatrix} 10 & 5 & 9 & 18 & 11 \\ 13 & 19 & 6 & 12 & 14 \\ 3 & 2 & 4 & 4 & 5 \\ 18 & 9 & 12 & 17 & 15 \\ 11 & 6 & 14 & 19 & 10 \end{bmatrix}$$

1. Each element of the current row minus the smallest element.

$$\begin{bmatrix} 5 & 0 & 4 & 13 & 6 \\ 7 & 13 & 0 & 6 & 8 \\ 1 & 0 & 2 & 2 & 3 \\ 9 & 0 & 3 & 8 & 6 \\ 5 & 0 & 8 & 13 & 4 \end{bmatrix}$$

2. Each element of the current column minus the smallest element.

$$\begin{bmatrix} 4 & 0 & 4 & 11 & 3 \\ 6 & 13 & 0 & 4 & 5 \\ 0 & 0 & 2 & 0 & 0 \\ 8 & 0 & 3 & 6 & 3 \\ 4 & 0 & 8 & 11 & 1 \end{bmatrix}$$

3. Cover all the zero elements with a minimum number of m lines.

$$\begin{bmatrix} 4 & 0 & 4 & 11 & 3 \\ 6 & 13 & 0 & 4 & 5 \\ 0 & 0 & 2 & 0 & 0 \\ 8 & 0 & 3 & 6 & 3 \\ 4 & 0 & 8 & 11 & 1 \end{bmatrix}$$

4. In step 3, m=3<n=5,we will find out the minimum element of the elements which are not covered by the m lines(the element is 1), and subtracts it from all elements uncovered, and add it to all the elements crossed by both the horizontal lines and the vertical lines, then we get the new simplified value coefficient matrix as follows:

$$\begin{bmatrix} 3 & 0 & 4 & 10 & 2 \\ 5 & 13 & 0 & 3 & 4 \\ 0 & 1 & 3 & 0 & 0 \\ 7 & 0 & 3 & 5 & 2 \\ 3 & 0 & 8 & 10 & 0 \end{bmatrix}$$

5. Turn to step 3 and step 4, until m=n=5. the final matrix is as follows:

$$\begin{bmatrix} 0 & 0 & 4 & 7 & 2 \\ 2 & 13 & 0 & 0 & 4 \\ 0 & 4 & 6 & 0 & 3 \\ 4 & 0 & 3 & 2 & 2 \\ 0 & 0 & 8 & 7 & 0 \end{bmatrix}$$

6. Find a feasible solution. Now we should find 5 zeros in different rows and different columns, the specific steps are as follows:
   (1) Find out the rows (or the columns) that have only one zero, mark the zero with a "√".
   (2) Mark the zeros in the column (or row) where there's a "0"already marked with "√" with "×".
   (3) Repeat step (1) and step (2) to the end. If there're more than one zero in all rows, then choose the row which has the least zeros and mark any one of zeros in the row with a "√".
   According to the steps, the match of the simplified value coefficient matrix is as follows:

$$\begin{bmatrix} 0\surd & 0\times & 4 & 7 & 2 \\ 2 & 13 & 0\surd & 0\times & 4 \\ 0\times & 4 & 6 & 0\surd & 3 \\ 4 & 0\surd & 3 & 2 & 2 \\ 0\times & 0\times & 8 & 7 & 0\surd \end{bmatrix}$$

Hence, the optimal solution is :

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The variable $x_{ij} = 1$ indicates we should distributing mobile sensor i to heal the hole j to get the optimal solution. So the minimum overall distance is $Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} = 10 + 6 + 4 + 9 + 10 = 39$.

The steps of the Hungarian method are clarity that can be easily programmed. In the pseudo-code of HBA, we write a function *Hungarian_assign (matrix)* to calculate the optimal match of the sensors and the coverage holes when $N(m_j)$ equals $N(s_i)$.

In the situation when $N(m_j)$ is less than $N(s_i)$, we can add $|N(s_i) - N(m_j)|$ virtual nodes or virtual destinations to make the value coefficient matrix a square matrix. And then match the square matrix with the Hungarian method aforementioned. However, we can't guarantee the coverage rate in this circumstance. As the situation illustrated in Fig. 4: M_1, M_2, M_3 are three mobile sensors and a, b, c, d, e are five destinations, the size of the circles express the size of the coverage holes.

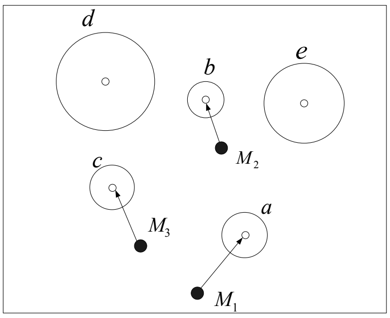


Fig. 4. The situation when $N(m_j)$ is less than $N(s_i)$

Algorithm 1 The Hungarian method-based algorithm

Notations:

s_i, m_j : a static and a mobile node, respectively

R_s : uniform sensing radius of all the nodes

$V(s_i)$: Voronoi polygon of node s_i

V_i : The farthest Voronoi vertex of $V(s_i)$

$(x_i, y_i), (x_j, y_j), (x_{V_i}, y_{V_i})$: coordinates of node s_i, m_j and V_i , respectively

d_i: the distance between s_i and V_i

$N(s_i)$, $N(m_j)$: total number of static nodes and mobile nodes deployed randomly, respectively

matrix: the value coefficient matrix whose elements presents the distances of $N(m_j)$ mobile nodes and $N(s_i)$ coverage holes respectively

Initialization phase:

1. Deploy all the static and mobile nodes randomly in the target area and number them. Choose a random static sensor S_i as the sink node.

   At each static node s_i :

   send (x_i, y_i) to S_i
2. Construct a Voronoi diagram with all the static nodes.
3. Coverage hole discovery and estimation phase.

   At each static node s_i :

for i = 0 to $N(s_i)$ **do**

Scan each vertex of $V(s_i)$ and determine the farthest Voronoi vertex V_i as the target location, send (x_{V_i}, y_{V_i}) to S_i

$d_i \leftarrow \sqrt{(x_{V_i} - x_i)^2 + (y_{V_i} - y_i)^2}$

$hole\_size_i \leftarrow \pi(d_i - R_s)^2$, send $hole\_size_i$ to S_i

end for

Determine value coefficient matrix:

If $N(m_j)$ equals $N(s_i)$

for i = 1 to $N(m_j)$

for j = 1 to $N(m_j)$

$d_{ij} \leftarrow \sqrt{(x_{V_i} - x_j)^2 + (y_{V_i} - y_j)^2}$

end for

end for

end if

If $N(m_j)$ is less than $N(s_i)$

The sink node compare the hole sizes of the $N(s_i)$ static sensors, and choose the number of $N(m_j)$ larger coverage holes to be the target holes, renumber them from 1 to $N(m_j)$

for i = 1 to $N(m_j)$

for j = 1 to $N(m_j)$

$dis_{ij} \leftarrow \sqrt{(x_{V_i} - x_j)^2 + (y_{V_i} - y_j)^2}$

end for

end for

end if

matrix $\leftarrow \left[dis_{ij}\right]_{N(\mathrm{m}_j)\times N(\mathrm{m}_j)}$

Core part of the algorithm: using Hungarian method to calculate the optimal match：

Hungarian_assign (matrix);

Final motion phase:

Dispatch the mobile nodes to heal the coverage holes according to the calculation results.

Fig. 5. Detailed description of HBA

If we use the Hungarian method directly, the solution may be as follows: sensor M_1 is matched with hole a, sensor M_2 with hole b, and sensor M_3 with hole c. But we can't guarantee the coverage rate for there're larger holes d and e in farther places. So we should do some improvement of our algorithm to enhance the coverage rate. Our solution is as follows: before matching the $N(m_j)$ mobile nodes to the $N(s_i)$ coverage holes by the Hungarian method, the sink node choose $N(m_j)$ number of larger coverage holes as the destination of the mobile nodes. Fig. 5 is the pseudocode of HBA.

6 The Threshold-based Algorithm (TBA)

Using the HBA, we can get the optimal solution ensures the overall moving distance of mobile sensors is shortest, but it still can't guarantee load balance. For instance, if the value coefficient matrix is as follows:

$$(c_{ij}) = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 4 & 1 \\ 3 & 3 & 2 \end{pmatrix}$$

We can get the minimum overall moving distance is 1+1+3=5 according to the HBA. However, in a mixed WSN, the sensor which moved 3 units consumed more energy than the two ones moved 1 unit, then we call the sensor moved 3 units is penalized. This problem motivated us to propose *the threshold-based algorithm* (TBA) to better allocate mobile sensors to coverage holes such that the overall moving distance is as short as possible and no sensor is penalized. An ideal solution of this instance is 2+2+2=6.

The core idea of the TBA is to find an allocation scheme, which can minimize the energy consumption of the mobile sensor that consumes the most, namely make sure that the longest distance that a sensor moves is the shortest. This problem can be attributed to the assignment of bottle neck. An effective method to solve this problem is the *small ring element algorithm* [11]. The author in [11] has fully proved the correctness of the small ring element algorithm.

The steps of the small ring element algorithm are as follows:

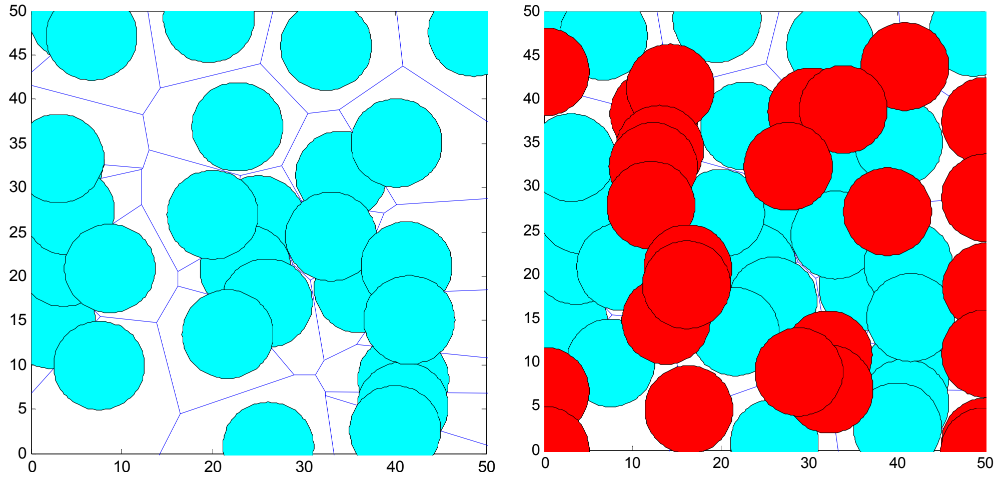
1. Find out the minimal element in each row of the value coefficient matrix $(C_{ij})_{n\times n}$ and mark a circle with it, then the same method to the columns.
2. Try to find out a match with the same method in step 6 of the example given in HBA (we regard the elements marked with circles zero elements).
   If we can find a feasible solution by this way, the solution is the optimal allocation scheme. Else, turn to step 3.
3. Find out the minimal element in the elements not be marked with circles, and then mark it with a circle. Turn to step 2.

The pseudocode of TBA is roughly same with the HBA except for the Core part of the algorithm is using small ring element algorithm to calculate the optimal match. We also write a function *Small_ring_assign (matrix)* to gain the optimal match which minimizes the longest moving distance of the sensors.

7 Performance evaluations

7.1 Coverage rate

Our centralized energy-efficient algorithms are implemented in Matlab. We consider a simulation sensing area of 50m×50m. Initially, 28 static sensors are randomly placed in the field, and the unified sensing range of both static sensors and mobile sensors is 5m. According to our centralized algorithm, the most number of mobile sensors we need is 27 in this instance. As shown in Fig. 6, the circles with a green shadow represent the sensing coverage of the static sensors, and the red shadows are that of the mobile sensors. Fig. 6. a shows the initial coverage rate of all the static sensors, which is 45 percent, and the coverage rate reaches 87 percent after the mobile sensors moved to the proper destinations calculated by our algorithm.



(a) (b)

Fig. 6. Coverage rate improvement of the centralized algorithms. (a) Initial coverage of the static sensors. (b) Final coverage of all sensors.

7.2 Energy consumption

Now we will compare the performance of energy-efficiency of the HBA and TBA. We assume that the number of the mobile sensors $N(m_j)$ is equal to the number of the static sensors $N(s_i)$.

First, we use an example to observe the performance of the HBA and TBA qualitatively. As illustrated in Fig. 7, we generate 18 mobile sensors and 18 destinations randomly in the sensing area. (a), (b) is the optimal match calculated by HBA and TBA, respectively. As shown in Fig. 7. a, the longest distance of mobile sensor calculated by HBA is between sensor 1 and destination a, which is much longer than some other sensors' moving distance, so the sensor 1 is penalized. In Fig. 7. b, we can observe that by using the TBA, the longest distance has been shortened by 29 percent, and it seems that the load-balance of TBA is superior to HBA. Fig. 8 is the condition when $N(m_j)$ is 23, we came to the same conclusion with Fig.7.

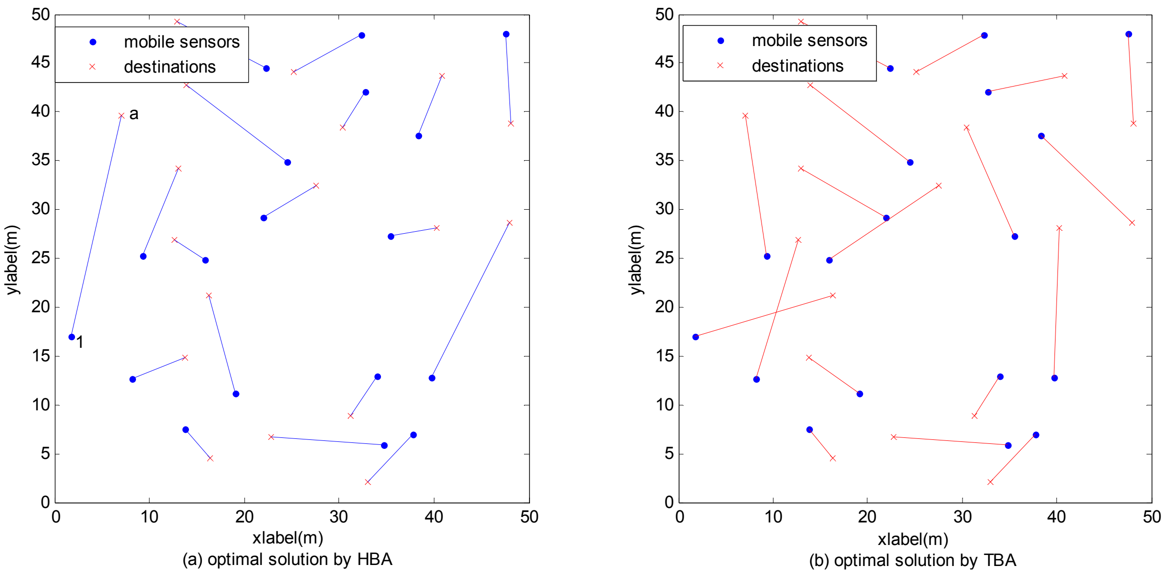


Fig. 7. Trace of the mobile sensors when $N(m_j)$ is 18

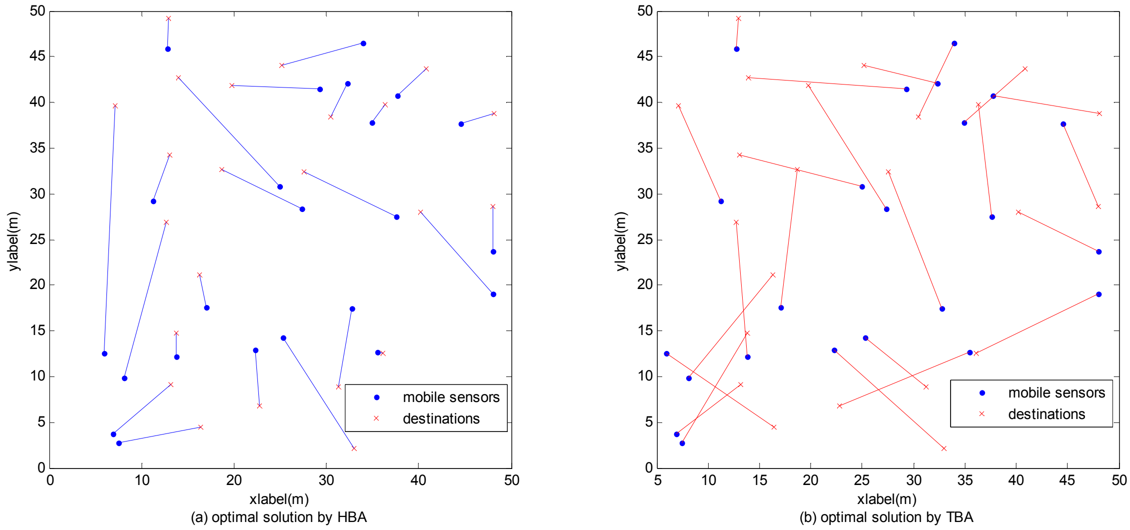


Fig. 8. Trace of the mobile sensors when $N(m_j)$ is 23

Then, we will give detail quantitative analysis of the two algorithms. We run both HBA and TBA while $N(m_j)$ ranges from 5 to 35. We have two objectives in conducting the simulation: first, comparing the overall moving distance of mobile sensors of the two algorithms while the number of the mobile sensors ranges; Second, comparing the variance of each node in mobile distance. In each situation we generate 100 sets of value coefficient matrixes randomly to represent the distance between each mobile sensor and destination, then run the HBA and the TBA respectively, calculate the average value of the overall distance and the variance of the distance average of the same. Fig. 9 is the comparison of the two algorithms.

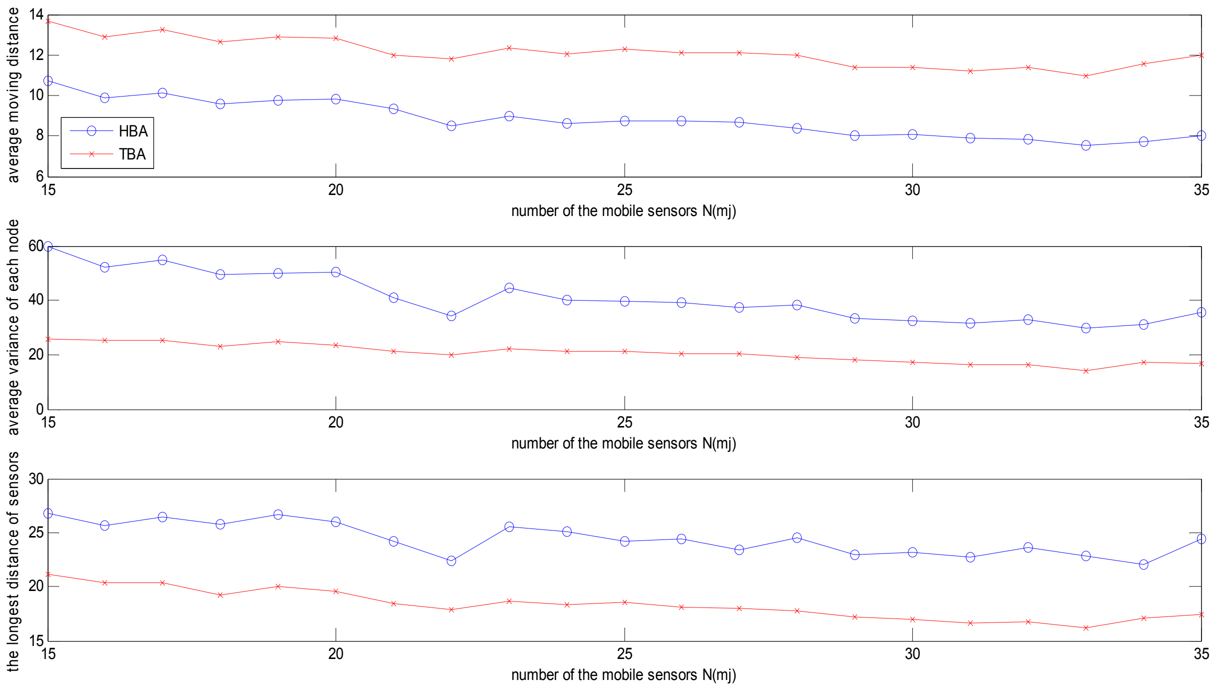
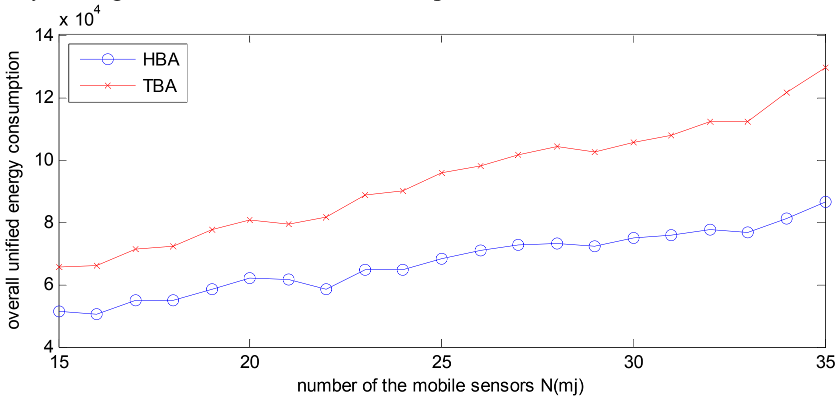


Fig. 9. Quantitative comparison of HBA and TBA when $|N(m_j)|$ ranges from 5 to 35

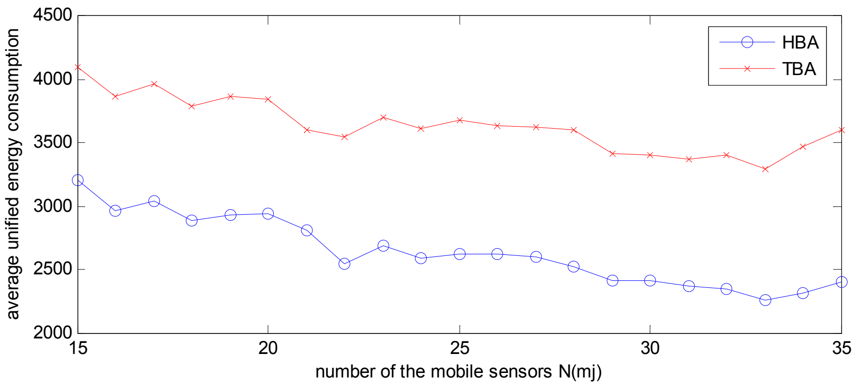
Fig. 9. a is the average moving distance of a mobile sensor calculated by HBA and TBA respectively, we can conclude that the distance calculated by HBA is shorter than that of TBA, this conclusion is obvious as HBA is the algorithm which can minimize the overall moving distance of mobile sensors. We also observed that when the number of the mobile sensors ranges, the difference of average moving distance changed little. When the number of mobile sensors is 15, the distance calculated by TBA is 2.95m longer than HBA, and 3.98m longer

when the number is 35. In Fig. 9. b, we compare the variance of each node in moving distance of HBA with that of TBA, we can see that the variance calculated by TBA is much smaller than that of HBA. As variance is a measurement of the load balance of network, so we can conclude that the load balance of the network calculated by TBA is superior to that of HBA. Fig. 9. c illustrates the longest distance of a single sensor calculated by HBA compared to that of TBA. When the number of mobile sensors is 15, the longest distance by TBA is 5.65m shorter than that of HBA, when mobile sensor number is 35, the gap of distance between HBA and TBA becomes 7m. As the longest distance of a single sensor is crucial to the lifecycle of the network, so we came to a conclusion that the TBA is much better than HBA in prolonging the lifecycle of the network.

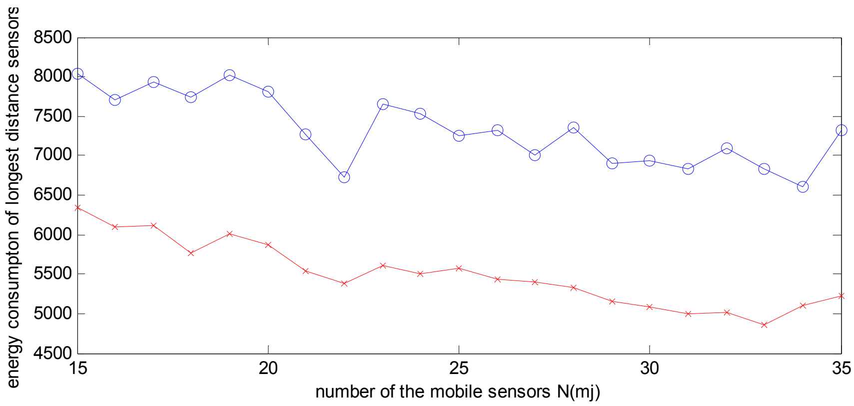
To establish the relationship between moving distance and energy consumption, we fellow the Robomote specification, approximately, to move a sensor one meter consumes amount of energy as transmitting 300 messages [5]. Fig. 10 shows the unified energy consumption of HBA and TBA respectively in the overall energy consumption, the average energy consumption and the largest energy consumption of a sensor, from which we can conclude the HBA is more energy-efficient when considering the whole network cost; As the lifecycle is closely related to the largest energy consumption node, when talking about the comprehensive performance and prolonging the lifecycle, Fig. 10. c illustrated TBA is a superior one.



(a)



(b)



(c)

Fig. 10. Unified energy consumption

8 Conclusions

In this paper, we proposed using a mix of mobile and static sensors to construct sensor networks. In terms of the energy-efficient problem in the coverage problem, we designed two centralized algorithms (HBA and TBA) to dispatch mobile sensors to the optimal destinations. HBA can minimize the overall moving distance of mobile sensors while the TBA is an improvement of the former, which can minimize the longest moving distance of the mobile sensor. We compared the two algorithms by simulation to verify the conclusion that TBA can do better in load balance and achieve great improvement in prolonging the lifecycle of a mixed WSN, which is a better algorithm in terms of comprehensive performance.

In the future, we will work on algorithms which can enhance the coverage rate to a relative high degree. We may use other sensing models instead of the sensing disk model such as the *hybrid sensing model* to describe the degradation of sensor sensing capability as the distance increases accurately. And we may combine the centralized algorithm with the distributed algorithm to balance the coverage rate of the network and energy consumption.

9 Acknowledgements

This research is supported by Beijing Science and Technology Program under Grant Z121100007612003.

References:

[1]. Akyildiz, I.F., et al., "A survey on sensor networks." *Communications Magazine, IEEE,* pp. 102- 114, 2002.

[2]. You-Chiun, W. and T. Yu-Chee, "Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multilevel Coverage." *Parallel and Distributed Systems, IEEE Transactions on*, pp. 1280-1294, 2008.

[3]. Zou, Y. and C. Krishnendu. "Sensor deployment and target localization based on virtual forces." in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*. IEEE Societies. 2003.

[4]. Chellappan, S., et al., "Mobility Limited Flip-Based Sensor Networks Deployment. Parallel and Distributed Systems", *IEEE Transactions on*, pp. 199-211, 2007.

[5]. Wang, G., et al., "Bidding Protocols for Deploying Mobile Sensors." *Mobile Computing, IEEE Transactions on*, pp. 563-576, 2007.

[6]. Shuhui, Y., W. Jie and D. Fei. "Localized Movement-Assisted Sensor Deployment in Wireless Sensor Networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*. 2006.

[7]. Guiling, W., C. Guohong and T. La Porta. "Movement-assisted sensor deployment." in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies.* 2004.

[8]. Chia-Pang, C., et al. "A novel coverage-preserving algorithm with energy efficiency." in *Sensors, 2009 IEEE. 2009. Christchurch.*

[9]. Niculescu, D. and B. Nath. "Ad hoc positioning system (APS) using AOA." in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. 2003.

[10]. Franz Aurenhammer. "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure." in *ACM Computing Surveys*, pp. 347-405, 1991.

[11]. Zhan Mingqin, "Small Ring Element Algorithm for Assignment Problem of Bottle Neck." *Journal of WuHan Institute of Technology*, pp. 61-64, 1994.