Boundary Monitoring Algorithms for Wireless Sensor Networks of Grouping Capabilities

Cheng-Ta Lee and Frank Yeong-Sung Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
e-mail: {d90001,yslin}@im.ntu.edu.tw

***Abstract*—In this paper, we develop three heuristic algorithms to construct efficient boundary monitoring for wireless sensor networks of grouping capabilities. We try to find the maximum *K* groups of sensors for boundary monitoring of sensor field. The mechanism can prolong the system lifetime. This problem is formulated as 0/1 integer-programming problem. Three heuristic-based algorithms are proposed for solving the optimization problem. The experimental results showed that the proposed efficient boundary monitoring algorithm (EBMA) gets a near optimization in the efficient boundary monitoring for grouping capabilities.**

Keywords- wireless sensor networks; boundary monitoring; grouping; full coverage

I. INTRODUCTION

In the past few years, from either practical or theoretical domain, the application and technique development of wireless sensor networks (WSNs) are important research issues [1][2]. Some interesting applications for WSNs have been investigated, e.g., surveillance, object positioning, object tracking, intrusion detection, anti-terror, and health care. In addition, under some applied circumstances, we just need to record the objects that enter or leave the boundary of monitored area [6][8][9], eg., the preservation area administrators must be notified when the hunters enter or leave the wildlife preservation area in order to take necessary action. Besides, intrusion detection of enemies and layered defense are also required to record whether the objects enter or leave the boundary of monitored area for further notification and following track.

In this paper, we focus on the sensor grouping problem to support boundary monitoring service. First, we try to find the boundary nodes from monitoring region. Second, we will deal with the problem of arrival and departure for the objects. Third, we want to find the maximum *K* groups of sensors to monitor a sensor field boundary. This mechanism can prolong the system lifetime.

In the prior studies [3][4][10], In [3], Sam, et al. propose a optimized communication and organization method called OCO to find the boundary nodes. In [4], Sahoo, et al. propose two boundary node selection algorithms, called SBNS and DBNS, to find out the boundary nodes. The two methods have three phases to find out the boundary nodes. In the initial phase, each sensor node in the monitoring region could be classified as boundary nodes or non-boundary nodes after the initial phase is executed. In the selection phase, the ring of boundary node can be found. In the pruning phase, the redundant boundary nodes are changed to non-boundary nodes. The DBNS approach tries to find out the boundary nodes by distributed method. In [10], P.L. Chiu, et al. construct the sensor network such that it includes *K* mutually exclusive sets (number *K* is given). These sets are called covers. The covers are disjoint covers. The method can find out the boundary nodes and prolong the system lifetime.

In this paper, we introduce the concept of check point and the check points can assist to check full coverage. Besides, it can save energy consumption because the concept can check full coverage more efficiently for arbitrary topology. And further, we find the maximum *K* sets of sensors to support boundary monitoring service on monitoring region. These sets can be joint or disjoint sets. Each of them is called a group, and can provide full coverage of the boundary of sensor field. Each group is activated in turn to monitor the boundary of monitoring regions. Generally, the power consumption for inactive sensors can be neglected, and the system lifetime can be effectively prolonged up to *K* times. We present a mathematical model to describe the optimization problem and three heuristic-based algorithms are proposed to solve the problem.

We formulate the problem as a 0/1 integer programming problem where the objective function is the maximization of the system lifetime of the boundary of monitoring region subject to full coverage, battery capacity, and variables integer constraints. We construct three heuristic-based algorithms to solve the problem.

The problem is formulated as a linear optimization-based problem with three different decision variables: wakeup sensors, covered check points, and full coverage in the round *r*. Wakeup sensors are 1 if sensor *s* is awake in the round *r*, and 0 otherwise. Covered check points are 1 if check point *a* at least is covered by one awake sensor in the round *r*, and 0 otherwise. Full coverage is 1 if full coverage boundary check points in the round *r*, and 0 otherwise. In the further computational experiments, our proposed boundary monitoring for grouping capabilities algorithm is expected to be efficient and effective in dealing with the optimization problem.

From papers review, we find that this study differs from prior works in several points. First, we consider both the energy conservation and lifetime extending during the sensor

978-1-4244-5849-3/10/$26.00 ©2010 IEEE

deployment phase for boundary monitoring. Second, we present a mathematical model to describe the optimization problem. Third, the relationship between the grouping capabilities of boundary node and the maximum extension of system lifetime is investigated. Fourth, we present a new concept of the check point. Fifth, we can find boundary nodes in user define disjoint monitoring region.

The rest of this paper is organized as follows. The problem and mathematic model are described in section II and III, respectively. Additionally, the solution procedure is presented in section IV. Furthermore, the computational results are discussed in section V, and conclusions are presented in section VI.

II. PROBLEM DESCRIPTION

A. Boundary Nodes Selection

In this section, we use the mathematical method to select boundary node. We particularly introduce novel concept of check point for full coverage check points. The monitoring region can be represented as a collection of 2D region. It includes check points and sensor nodes, as illustrated in Figure 1. This approach is called check point-based boundary node selection. The positioning resolution requirement of application determines the granularity of check point and sensing range.

We introduce the concept of check point. The check point can assist to check full coverage. Besides, it can save energy consumption because the concept can check full coverage more efficiently for arbitrary topology and disjoint monitoring regions.

Definition 1: The check points are virtual points. The distance of each neighboring check points is small or equal to minimum size of monitoring object.

Lemma 1: The boundary of monitoring region is full coverage if all check points are covered by sensors.

The proposed efficient boundary nodes selection (EBNS) algorithm is shown in Figure 2.

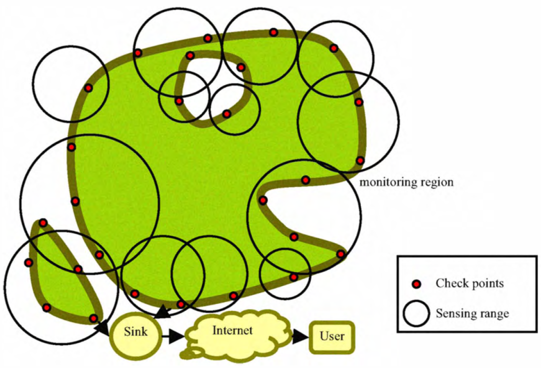

Figure 1. Check point-based boundary node selection.

Algorithm Efficient Boundary Nodes Selection

Input: Coordinate of check points and sensor nodes, and sensing radius of sensor nodes
Output: Boundary nodes (*BNSet*)

1: **begin**
2: *BNSet*= ∅ ;/* *BNSet*: the set of boundary nodes */
3: *UncoverSet*= ∅ ; /* *UncoverSet*: the set of uncovered check points */
4: **for** *a*=1 **to** *cp* **do** /* *cp*: number of check point */
5: $flag_a$=0;
6: **for** *a*=1 **to** *cp* **do**
7: **begin**
8: **for** *s*=1 **to** *sn* **do** /* *sn*: number of sensor node*/
9: **begin**
10: **if** check point *a* is covered by sensor node *s*
/* $\sqrt{(x_s - x_a)^2 + (y_s - y_a)^2} \le r_s$ */
11: **then** *BNSet*← sensor node *s* and $flag_a$=1
12: **end**
13: **if** $flag_a$=0
14: **then** *UncoverSet* ← check point *a*
15: **end**
16: **if** *Uncoverset* ≠ ∅
17: **then** boundary of monitoring region is not full coverage
18: **else** boundary of monitoring region is full coverage and boundary nodes=*BNSet*
19: **end**

Figure 2. Boundary nodes selection algorithm.

In this algorithm, from steps 2-5 set initialize values. Steps 6-15 are used to find boundary node set. Steps 16-18 check full coverage.

We use above EBNS algorithm to find out boundary nodes and check full coverage of boundary.

B. Arrival and Departure of Objects

We assume that $r_c \geqq 2\max r_s + w$ and $w > 2\max r_s$, where r_c is communication radius, r_s is sensing radius, and w is minimum size of monitoring object, as shown in Figure 3.

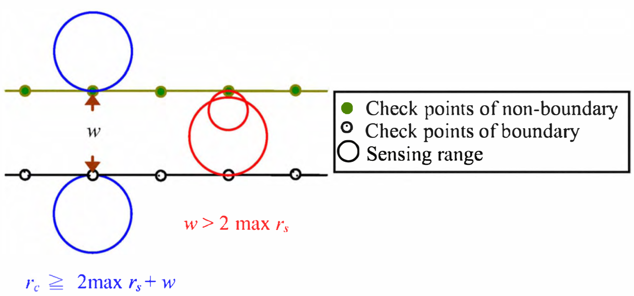

Figure 3. Assumption of communication and sensing radii.

We propose two algorithms, single ring algorithm (SRA) and double ring algorithm (DRA), to deal with the problem of arrival and departure of objects. In the single ring algorithm, an object is sensed by boundary nodes (*BNs*) while it touches the monitoring region, and *BNs* will wake up their neighboring non-boundary nodes (*non-BNs*). For the next moment, if *BNs* do not sense the object but neighboring *non-BNs* sense the object, the object is entering the monitoring region.

Similarly, the neighboring *non-BNs* of *BNs* detect the object. For the next moment, if *BNs* sense the object and soon after they do not sense the object, and neighboring *non-BNs* do not sense the object, the object is leaving the monitoring region, as shown in Figure 4.

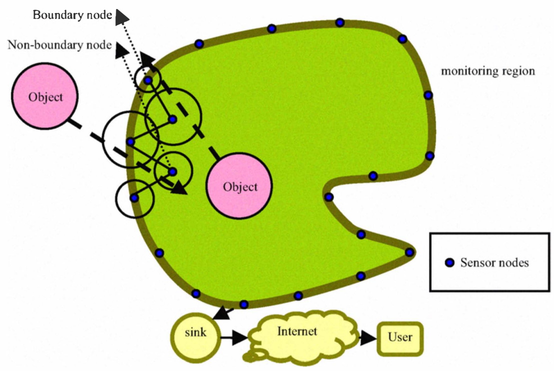

Figure 4. Single ring for arrival and departure of objects.

In the double ring algorithm, an object is sensed by *BNs* of outer ring while the object touches the monitoring region. For the next moment, if outer ring *BNs* do not sense the object and inner ring *non-BNs* sense the object, the object is entering the monitoring region.

Similarly, the inner ring *non-BNs* detect the object. For the next moment, if outer ring *BNs* sense the object and presently do not sense the object, and inner ring *non-BNs* do not sense the object, then the object is leaving the monitoring region, as shown in Figure 5.

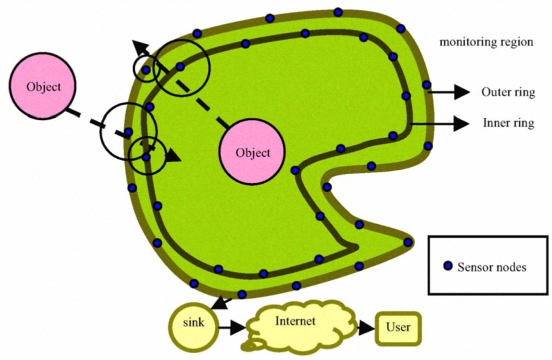

Figure 5. Double ring for arrival and departure of objects.

C. *Boundary Monitoring Algorithms for Grouping Capabilities*

We try to find maximum *K* sets of sensors to support boundary monitoring service on monitoring region, as shown in Figure 6. Each of them, is called a group, can provide full coverage of the boundary. Each group is activated in turn to monitor the boundary. Figure 7 shows the state transitions of the sensor network. From the network viewpoint, two operation states exist: the sleeping and active states. Only one group sensors are activated in turn to monitor the boundary, and the other group sensors are sleeping in one time. The system lifetime can be effectively prolonged up to *K* times.

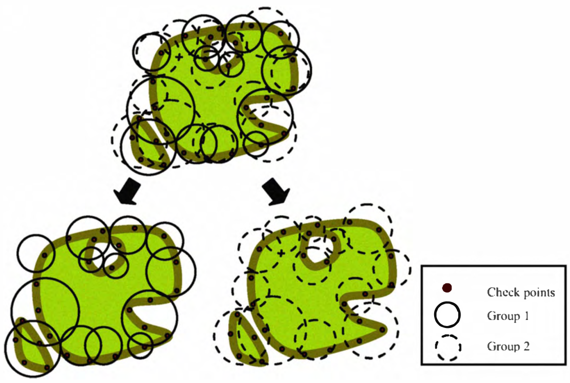

Figure 6. Boundary monitoring for grouping capabilities.

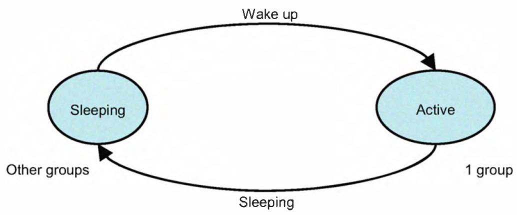

Figure 7. The state diagram of the sensor network

III. PROBLEM FORMULATION

In this section, we formulate the problem as a 0/1 integer programming problem where the objective function is the maximization of the amount of cover *K* required to full coverage under a given boundary of sensor networks. The problem is a variant of the set *K*-cover problem and thus is NP-complete [11].

The notations used to model the problem are listed in Table I and II.

TABLE I. NOTATIONS FOR THE GIVEN PARAMETERS.

Given Parameters	
Notation	Description
S	The set of all sensor nodes.
A	Index set of the service check points in the monitoring region boundary.
C_s	The initial energy level of sensor node s.
E_s	The energy consumption for aware sensor node s to sense data in each round.
R	The total number of rounds.
b_{sa}	The indicator function which is 1 if the check point a is in the sensing range of the sensor node s, and 0 otherwise.

TABLE II. NOTATIONS FOR THE DECISION VARIABLES.

Decision Variables	
Notation	Description
π_{sr}	1 if sensor s is awake in the round r, and 0 otherwise.
y_{ar}	1 if check point a at least is covered by one awake sensor in the round r, and 0 otherwise.
z_r	1 if full coverage boundary check points in the round r, and 0 otherwise.

Problem (IP):

$$\max \sum_{r\in R} z_r \quad \text{(IP)}$$

subject to:

The full coverage boundary check points constraints

$$y_{ar} \le \sum_{s\in S} b_{sa}\pi_{sr} \qquad \forall a\in A, r\in R \qquad (1)$$

$$z_r \le \frac{\sum_{a\in A} y_{ar}}{|A|} \qquad \forall r\in R \qquad (2)$$

The battery capacity constraint

$$\sum_{r\in R} \pi_{sr} E_s \le C_s \qquad \forall s\in S \qquad (3)$$

The integer constraints

$\pi_{sr} = 0$ or 1 $\qquad \forall s\in S,\ r\in R$ (4)

$y_{ar} = 0$ or 1 $\qquad \forall a\in A, r\in R$ (5)

$z_r = 0$ or 1 $\qquad \forall r\in R$. (6)

The objective function is to maximize the system lifetime of the monitoring region boundary. The lifetime is defined as the total number of rounds.

Constraints (1)-(2): Full coverage boundary check points constraint.

Constraint (3): For each sensor node s, the total sensing consumption can not exceed its initial energy level.

Constraints (4)-(6): The integer constraints for decision variables π_{sr}, y_{ar}, and z_r.

IV. SOLUTION APPROACH

The parameters and decision variables used to model our algorithms in this section are listed in Table III.

TABLE III. THE PARAMETERS AND DECISION VARIABLES IN OUR ALGORITHMS

Notation	Description
max_k	The upper bound of system lifetime.
cp	The number of check points.
sn	The number of sensor nodes.
cpc_no[*a*]	The number of covered rounds in each check point *a*.
cs[*s*]	The initial energy level of sensor node *s*.
es[*s*]	The energy consumption for aware sensor node *s* to sense data in each round.
max_round	The system lifetime.
c_bsa[*a*]	The number of covered times in check point *a* by waked sensors.
count[*s*]	The number of covered check points by waked sensor *s*.
c_s[*s*]	The number of covered check points under sensing range of sensor *s*.
t_cover	The number of full coverage in each iteration.
ub	The upper bound of number of finding the best fit sensor to cover check point.
bsa[*s*][*a*]	The indicator function which is 1 if the check point *a* is in the sensing range of the sensor node *s* and 0 otherwise.
f_coverage[*r*]	The decision variable which is equal to *cp* if full coverage boundary check points in the round *r*, and less than *cp* otherwise.
p[*s*][*r*]	The decision variable which is 1 if sensor *s* is awake in the round *r*, and 0 otherwise.
cover[*a*][*r*]	The decision variable which is 1 if check point *a* at least is covered by one awake sensor in the round *r*, and 0 otherwise.

A. *Upper Bound of the Maximum Rounds*

In this section, we study the upper bound of maximum rounds in boundary monitoring.

We can calculate the upper bound (*UB*) of system lifetime by follow algorithm in Figure 8.

Algorithm Upper Bound of the Maximum Rounds

Input: The initial energy level of sensor node *s*, the energy consumption for aware sensor node *s* to sense data in each round

Output: The upper bound of maximum rounds (*max_k*)

```
 1: begin
 2:   max_k=∞
 3:   for a=1 to cp do
 4:     cpc_no[a]=0;
 5:   for s=1 to sn do
 6:     for a=1 to cp do
 7:       if (bsa[s][a]=1)
 8:         then cpc_no[a]=cpc_no[a]+(cs[s]/es[s])
 9:   for a=1 to cp do
10:     if (cpc_no[a]<max_k)
11:       then max_k=cpc_no[a]
12: end
```

Figure 8. The algorithm of upper bound of system lifetime

In this algorithm, from steps 2-4 are setting initialize value, steps 5-8 are finding the maximum rounds value for each check point. Steps 9-11 are used to get system upper bound of the maximum rounds.

B. Simple Algorithm 1

We compare our proposed greedy-based algorithm with non-greedy-based algorithms (simple algorithm 1 and 2) that use the concept of "cover" to decide whether sensor *s* is awake in the round *r*. The *cover* is 1 if the check point *a* is in the sensing range of the sensor node *s* and 0 otherwise.

In each round, we first find sensor *s* to cover check point *a*, and then sensor *s* is awake in the round *r*, and repeat the assignment process until all check points have been covered.

C. Simple Algorithm 2

Simple algorithm 1 waste on energy consumption, because system has redundant waked up sensor nodes. Therefore, we propose simple algorithm 2 (SA2) to deal with the problem. For example, s_2 is redundant sensor node as shown in Figure 9.

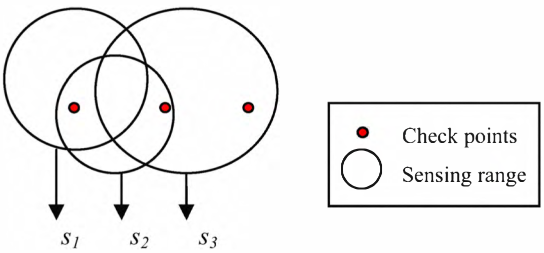

Figure 9. An example of deleting redundant sensor node.

D. Efficient Boundary Monitoring Algorithm

In this section, we present a greedy-based efficient boundary monitoring algorithm (EBMA) to improve SA1 and SA2.

To solve the original problem near-optimally. We use the *f_coverage*[*r*] to check full coverage in the round *r*. The decision variable which is equal to *cp* if full coverage boundary check points in the round *r*, and 0 otherwise. Then, in each round, we use different sensor node *id* to cover uncheck point *a* given minimum be cover check points and then sensor *s* is awake in the round *r*, and repeat the assignment process until all check points have been covered. For example, system prioritizes to select s_1 sensor node, because s_1 sensor node has not cover selected check points. If system can not find the s_1 sensor node, then second priority is s_2 sensor node, as shown in Figure 10.

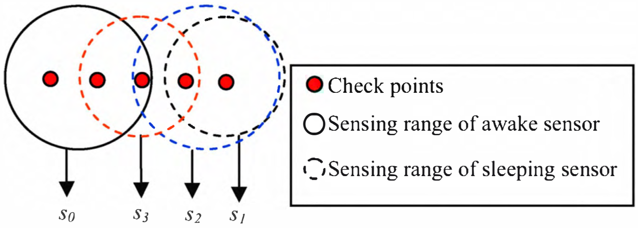

Figure 10. An example of greedy-based sensor node selection.

An efficient boundary monitoring algorithm is listed in Figure 11.

Algorithm Efficient Boundary Monitoring

Input: The initial energy level of sensor node *s*, the energy consumption for aware sensor node *s* to sense data in each round, and *max_k*

Output: The maximum rounds (*max_round*)

```
begin
  for r=1 to max_k do
  begin
    f_coverage[r]=0;
    for s=1 to sn do
      p[s][r]=0;
  end
  for r=1 to max_k do
  begin
    for i=1 to sn do
     for a=1 to cp do
     begin
      jump=0;
      for x=1 to ub do
       for s=1 to sn do
        if (((bsa[s][a]=1) and (cs[s]>=es[s]) and
          (cover[a]=0) and (x=ub)) or ((bsa[s][a]=1) and
          (cs[s]>=es[s]) and (cover[a]=0) and (bsa[s][a-
          x]=0))) then
```

```
17:         begin
18:          p[s][r]=1;
19:          cs[s]=cs[s]-es[s];
20:          for k=1 to cp do
21:           if (bsa[s][a]=1)
22:            c_bsa[a]=c_bsa[a]+1;
23:           if ((bsa[s][a]=1) and (cover[a][r]=0))
24:              then cover[a][r]=1 and
                       f_coverage[r]=f_coverage[r]+1;
25:          jump=1;
26:          break;
27:         end
28:         else
29:          continue;
30:         if (jump=1) then break;
31:       end
32:       if (f_coverage[r]=cp) then  /* delete redundant
                                            nodes */
33:         for s=1 to sn do
34:         begin
35:          for a=1 to cp do
36:           if ((p[s][r]=1) and (bsa[s][a]=1) and
                (c_bsa[a]>=2))
37:            then count[s]=count[s]+1;
38:           if (count[s]=c_s[s]) then
39:           begin
40:             cs[s]=cs[s]+es[s];  /* recovery energy */
41:             p[s][r]=0;
42:            for a=1 to cp do
43:            if (bsa[s][a]=1)
44:             then c_bsa[a]=c_bsa[a]-1;
45:           end
46:         end
47:        end
48:        t_cover=0;
49:        for r=1 to max_k do
50:         if (f_coverage[r]=cp)
51:          then round r is full coverage and
                 t_cover=t_cover+1;
52:        if (round< t_cover)
53:         then round= t_cover;
54:        if (max_round<round)
55:         then max_round=round;
56:        round=-∞;
57: end
```

Figure 11. The efficient boundary monitoring algorithm.

In the algorithm, from steps 2-7 are to set initial values. Steps 9-31 are to decide whether sensor s is awake in the round r. Steps 32-46 are used to delete redundant sensor nodes. Steps 48-56 are used to get system maximum rounds.

After solving the problem, a set of feasible solutions of the problem (IP) then can be obtained. The feasible solution is a lower bound (*LB*) of the problem (IP), and the *max_k* is the upper bound (*UB*) of the problem (IP). We get the *UB* and *LB*, respectively. The gap between *UB* and *LB*, computed by $|(UB - LB)/LB| * 100\%$, illustrates the optimality of problem solution. The smaller gap computed, the better the optimality.

V. COMPUTATIONAL EXPERIMENTS

To evaluate the performance of the proposed algorithm, we conduct an experiment. The performance is assessed in terms of total rounds.

A. Scenario

The proposed algorithm is coded in C under a dev C++ 4.9.9.2 development environment. All the experiments are performed on a Core 2 Duo 2.2G Hz PC running Microsoft Windows Vista. The algorithm is tested on a 2D sensor field. We distribute 100, 400, and 1600 sensor nodes and 36, 72, and 156 check points respectively in 2D sensor field. The radius of different sensors types (s_a, s_b, s_c, s_d) is (1, 2, 3, 4). The energy consumption of aware different sensor types (s_a, s_b, s_c, s_d) is (1, 4, 9, 16) in each round. The initial energy level of each sensor node s is 32.

B. Experimental Results

Table IV shows the maximum total number of rounds calculated by different algorithms. We can see that the EBMA outperforms the SA1 and SA2 algorithm.

The results show that the algorithm is better than the SA1 and SA2, and the gap is also small. In other words, when compared with SA1 and SA2, the proposed EBMA can improve the percentage of energy consumption from 11% to 61%. In the test problems, EBMA also achieves optimality since the gaps are 0%.

VI. CONCLUSION

This study proposes a boundary monitoring algorithm in wireless sensor networks. To our best knowledge, the proposed algorithm is truly novel and it has not been yet discussed in previous researches. This study first formulates the problem as a 0/1 integer programming problem, and then proposes a heuristic-based algorithm for solving the optimization problem.

The experimental results show that the algorithm is not only better than the other heuristic algorithms, such as SA1 and SA2, but the gap is also small. Compared with SA1 and SA2, the proposed EBMA can improve system lifetime and achieve the optimal solution since the gaps are 0% in the test problems. Therefore, the results show that the proposed algorithm can achieve boundary monitoring for grouping capabilities. Furthermore, the algorithm is very efficient and scalable in terms of the solution time.

As to the next step, we plan to further investigate mobile capabilities model based on boundary monitoring application requirements and heuristic algorithms [5][6][7]. In addition, we are looking into the tradeoff of total number of rounds with various system issues, such as mobile capabilities, layered defense, etc.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey on Wireless sensor network,"IEEE Communication Magazine, pp. 102-114, August 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor network: a survey,"Computer Networks , vol. 38, pp. 393-422, 2002.

[3] S. P. Manh Tran and T. A. Yang, "OCO: Optimized Communication & Organization for Target Tracking in Wireless Sensor Networks," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing 2006, pp. 428-435.

[4] P. K. Sahoo, K. Y. Hsieh, and J. P. Sheu, "Boundary Node Selection and Target Detection in Wireless Sensor Network," IFIP International Conference on Wireless and Optical Communications Networks, 2007, pp. 1-5.

[5] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," IEEE INFOCOM, vol. 4, pp. 2302-2312, March 2005.

[6] G. Zhen, G. Zhen, Z. MengChu, and J. Guofei, "Adaptive Sensor Placement and Boundary Estimation for Monitoring Mass Objects," Part B, IEEE Transactions on Systems, Man, and Cybernetics, vol. 38, pp. 222-232, 2008.

[7] Y. Yoo and D. Agrawal, "Mobile Sensor Relocation to Prolong the Lifetime of Wireless Sensor Networks," Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, 2008, pp. 193-197.

[8] Z. Chi, Z. Yanchao, and F. Yuguang, "Detecting Coverage Boundary Nodes in Wireless Sensor Networks," IEEE International Conference on Networking, Sensing and Control, pp. 868-873, 2006.

[9] J. Lian, L. Chen, K. Naik, Y. Liu, and G. B. Agnew, "Gradient Boundary Detection for Time Series Snapshot Construction in Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, 2007.

[10] P. L. Chiu and F.Y.S. Lin, "Sensor Deployment Algorithms for Target Positioning Services," Doctoral Thesis, Graduate Institute of Information Management of National Taiwan University, 2007.

[11] S. Megerian and M. Potkonjak, "Low power 0/1 coverage and scheduling techniques in sensor networks," UCLA Technical Reports 030001, January 2003.

TABLE IV. EVALUATION OF THE GAP AND IMPROVEMENT RATIO WITH DIFFERENT NUMBER OF NODES

Number of nodes (check points, sensor nodes)	Monitoring Region (m^2)	EBMA	UB	Gap	SA1	Improvement Ratio to SA1	SA2	Improvement Ratio to SA2
(36, 100)	10×10	29	29	0	18	0.61	21	0.38
(76, 400)	20×20	23	23	0	15	0.53	20	0.15
(156, 1600)	40×40	20	20	0	16	0.25	18	0.11