Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes

Ryo Katsuma*, Yoshihiro Murata†, Naoki Shibata‡, Keiichi Yasumoto* and Minoru Ito*
*Graduate School of Information Science, Nara Institute of Science and Technology (NAIST),
Ikoma, Nara 630-0192, Japan, Email: {ryo-k, yasumoto, ito}@is.naist.jp
†Graduate School of Information Science, Hiroshima City University,
Hiroshima, Hiroshima 731-3194, Japan, Email: yosihi-m@hiroshima-cu.ac.jp
‡Department of Information Processing and Management, Shiga University,
Hikone, Shiga 522-8522, Japan, Email: shibata@biwako.shiga-u.ac.jp

***Abstract*—One of the important issues in wireless sensor network (WSN) is to k-cover the target sensing field and to extend its lifetime. We propose a method to k-cover the field and maximize the WSN lifetime by moving mobile sensor nodes to appropriate positions for a WSN consisting of both static and mobile sensor nodes which periodically collect environmental information. Our target problem is NP-hard. So, we propose a genetic algorithm (GA) based scheme to find a near optimal solution in practical time. In order to speed up the calculation, we devised a method to check a sufficient condition of k-coverage of the field. For the problem that nodes near the sink node have to forward the data from farther nodes, we make a tree where the amount of communication traffic is balanced among all nodes, and add this tree to the initial candidate solutions of our GA-based algorithm. Through computer simulations, we confirmed that our method achieves much longer k-coverage lifetime than conventional methods for 100 to 300 node WSNs.**

I. Introduction

Wireless Sensor Network (WSN) is a network consisting of many small sensor nodes capable of wireless communication, and it is used for environmental monitoring, border guards, and so on. One of the typical applications of WSN is to periodically collect environmental information such as temperature or brightness from many sensors in a large agricultural field or a forest to a *sink node*. In order to make such a WSN operate for a long term, we have to carefully schedule and adjust parameters of sensor nodes for sensing data, receiving and transmitting the data. Since each sensor node consumes battery amount not only for sensing data but also for receiving/transmitting the sensed data from/to other sensor nodes, the battery lifetime varies depending on the initial amount, the data size for transmission, the sensing frequency, and the radio transmission distance.

Many research efforts have extended the WSN lifetime. Tang et al. reduced power consumption by regulating communication frequency among sensor nodes[1]. Heinzelman et al. reduced total data transmission by merging the data received from multiple sensor nodes[2]. Most of the existing studies suppose *static sensor nodes* that are not movable after being deployed in the field. WSNs consisting of only static sensor nodes cannot automatically recover unsensable areas generated by failure or battery exhaustion of the corresponding sensor nodes.

Recently, some studies have utilized *mobile sensor nodes* with wheels and motors for WSNs. In [3], Mei et al. repaired unsensable areas (due to node failure or battery exhaustion) by moving mobile sensor nodes to those areas. In [4], Wang et al. proposed a method to guarantee k-coverage of the target field for a WSN constructed by mobile and static sensor nodes. Here, a given geographical field is *k-covered* if any points in the field is included in the sensing ranges of at least k sensor nodes. These studies aimed to k-cover the whole target field by moving mobile nodes to places where static nodes failed or were not placed. The existing studies do not try to extend the WSN lifetime using mobile nodes while maintaining k-coverage of the target field.

In this paper, we first formulate the problem to find the positions of mobile sensor nodes and the tree (called a *data collection tree*) connecting all nodes so that the target area is k-covered and the WSN lifetime is maximized, from a given target field, the positions of the sink node and all sensor nodes, and the power consumption parameters for sensor nodes. This problem is NP-hard since it implies a Minimum Geometric Disk Cover Problem (GDC)[5] as a special case.

Secondly, we propose a heuristic algorithm based on genetic algorithm (GA). For better search speed and solution quality, we make a good initial candidate solution for GA considering the fact that nodes near the sink node have to consume battery amount for forwarding data from other nodes. We also devised a fast decision method for a sufficient condition for k-coverage of the target field. In this method, we check for all grid points if at least k sensors are placed within the circle whose center is each grid point with a radius smaller than the sensing radius.

Through computer simulations, we confirmed that the lifetime achieved by our method is much longer than the other conventional methods on WSNs with several hundreds of nodes. We also confirmed that our k-coverage judgment technique can much more accurately judge k-coverage of the target field than other method [4] with reasonable computation time.

II. Problem of WSN Lifetime Maximization with Mobile Nodes

In this section, we present the WSN model and formulate the problem of maximizing lifetime of a WSN which k-covers

 DOI:10.1109/WiMob.2009.18

the specified target field with mobile sensor nodes.

A. WSN Model, Assumptions, and Definitions

1) Assumptions on Target WSN: We suppose a WSN in which a massive number of small battery-driven sensor nodes are deployed in *target field*. Sensor nodes periodically sense environmental information such as temperature, humidity or sunlight, and send it by multi-hop communication to a station called a *sink node*. We denote the set of points in the target field, the sink node, and the sensing frequency as $Field$, Bs, and I, respectively.

Both *static nodes* and *mobile nodes* are used as sensor nodes. Static nodes cannot be moved from their originally placed locations, while mobile nodes can move by wheels. Robomote [6] is an example of a mobile node. We denote the sets of static and mobile sensor nodes by $P = \{p_1, ..., p_l\}$ and $Q = \{q_1, ..., q_m\}$, respectively. We assume that there is no obstacle in $Field$, and a mobile node can move straight to an arbitrary position in $Field$.

Each static or mobile sensor node covers a disk with radius R centered on it as its sensing range. We denote the sensing range of sensor node s by $s.range$. Each sensor obtains data by sensing. We assume that the size of data obtained by each time of sensing is fixed and the data are sent to the sink node without compression or unification by a multi-hop path to the sink node. We denote the data size by D.

Each sensor node has wireless communication capability and its radio transmission range is a disk with a certain radius centered on it. Each sensor node can freely change its transmission power so that the radio transmission radius can be adjusted depending on the distance to the next-hop node. We assume no interference of wireless communication between nodes since the supposed sensing frequency is not very high. The transmission success rate is 100% if the receiving node is within this disk, and 0% if outside of the disk.

We assume that each mobile or static sensor node knows its location and informs sink node Bs of its location by single-hop or multi-hop communication at their deployment. For each sensor node s, we denote its location by $s.pos$. Similarly, we denote the location of the sink node by $Bs.pos$. Based on the locations of all sensor nodes, the sink node calculates the appropriate positions of the mobile nodes as well as communication paths connecting all sensor nodes to the sink node and informs all the nodes of the new positions and/or new paths by single-hop or multi-hop communication.

2) Assumptions of Power Consumption: Each sensor node s is assumed to have a battery with finite energy denoted by $s.energy$. Powers $Trans(k, d)$ and $Recep(x)$ required to transmit x[bit] for d[m] and receive x[bit] conform to the formulas (1) and (2), respectively[2].

$$Trans(x, d) = E_{elec} \times x + \epsilon_{amp} \times x \times d^n \tag{1}$$

$$Recep(x) = E_{elec} \times x \tag{2}$$

Here, E_{elec} and ϵ_{amp} are constants representing the power required by information processing and the power for amplification, respectively. The value of $n(\geq 0)$ is defined by the antenna properties. The value is specified somewhere between 0 and 2.

Powers $Sens()$ and $Listen(y)$ required to sense the information which is D[bit] data and listen to whether radio messages come or not for y[s] conform to the following formulas (3) and (4), respectively.

$$Sens() = E_{elec} \times D + E_{sens} \tag{3}$$

$$Listen(y) = E_{listen} \times y \tag{4}$$

Mobile nodes consume battery power not only by communication but also by movement. Power $Move(d)$ required to move d[m] conforms to formula (5) [7].

$$Move(d) = E_{move} \times d \tag{5}$$

Here, E_{move} is a constant. Each mobile node can move at V [m/s] where V is a constant value.

B. Problem Definition

When a WSN operates for a long time, batteries of some sensor nodes will be exhausted and k-coverage will be broken. Then, it is necessary to move mobile nodes one after another. So, we formulate a problem to derive the schedules of when and where each mobile node should move at each time during WSN operation time.

Let t_0 denote the initial WSN deployment time. Let t_{end} denote the time after k-coverage is no longer satisfied due to battery exhaustion. For each $q \in Q$ and each $t \in [t_0, t_{end}]$, let $Run(q, t)$ denote the speed (0 or V) and direction of q at time t. Then, for each $q \in Q$, we denote a speed-direction schedule for q's movement during time interval $[t_0, t_{end}]$ by formula (6).

$$schedule(q, [t_0, t_{end}]) = \bigcup_{t \in [t_0, t_{end}]} \{Run(q, t)\} \tag{6}$$

Given the information on the target field $Field$, a sink node Bs and its position $Bs.pos$, $s.pos$, $s.energy$, and $s.range$ for each sensor node $s \in P \cup Q$, and constants E_{elec}, ϵ_{amp}, n, E_{sens}, E_{listen}, E_{move}, V, D, and I, our target problem for maximizing the WSN lifetime denoted by t_{life} is to decide the schedule $schedule(s, [t_0, t_{end}])$ for each node $s \in P \cup Q$ that satisfies condition (7).

$$\forall t \in [t_0, t_{end}], \forall pos \in Field, |Cover(pos, t)| \geq k. \tag{7}$$

where

$$Cover(pos, t) = \{s | pos \in s.range(t) \wedge s \in P \cup Q \wedge s.energy(t) > 0\}. \tag{8}$$

Here, $s.energy(t)$ and $s.energy(t)$ represent s's sensing range and remaining battery at time t.

We define the WSN lifetime t_{life} as the time from initial deployment to the time when condition (7) is no longer satisfied. We assume that for any positions of all sensor nodes which achieve k-coverage of the field, multihop communication paths which connect all sensor nodes to the sink node and maximize the WSN lifetime can be decided. Then, our target problem is to obtain schedules of all sensor nodes' speed and direction that maximize the WSN lifetime to satisfy condition (7). Therefore, we define the following objective function (9).

$$\textbf{maximize } (t_{life}) \textbf{ subject to } (7) \tag{9}$$

C. Modified Target Problem

Our target problem formulated in section II-B is to decide speed-direction schedule of q during time interval $[t_0, t_{end}]$, for each $q \in Q$. Then, we must decide a data collection tree including all sensor nodes whenever the positions of mobile nodes change. Solving the problem is considered to be very difficult because of the wide solution space. Therefore, we adopt a heuristic method to solve this problem stepping on the several stages as the following procedures:

1) Solving the problem to find the positions of mobile nodes and the data collection tree for maximizing *the WSN forecast endtime* (defined later) satisfying condition (7).
2) Whenever the battery of any sensor node is newly exhausted, go to step 1.

In the problem of step 1, its input is the same as the original problem. Its output is the new position of each mobile node $q \in Q$ denoted by $q.newpos$ satisfying condition (10) and the parent node of each sensor node $s \in P \cup Q$ denoted by $s.send$. We have the following constraint on $q.newpos$.

$$|q.pos - q.newpos| < \frac{V}{I} \tag{10}$$

Here, the new position of each mobile node is in the area where each mobile node can move in $\frac{1}{I}$ seconds.

The WSN lifetime t_{life} is the time of WSN termination considering the movement of mobile nodes in the future. It is difficult to calculate t_{life} strictly. So, we define the *WSN forecast endtime* when the battery of some sensor node is newly exhausted in the objective function (condition (9)) instead of t_{life}. Thus, we use the following objective function.

$$\textbf{maximize}\left(t_{now} + \min_{s \in P \cup Q}\left(\frac{s.energy}{C(s)} - \frac{Move(|s.pos - s.newpos|)}{C(s)}\right)\right) \tag{11}$$

where t_{now} is current time, and $C(s)$ is the energy consumption of sensor node s per second. If $s \in P$, $|s.pos - s.newpos| = 0$. So, $\frac{s.energy}{C(s)} - \frac{Move(|s.pos - s.newpos|)}{C(s)}$ means the time from present until the battery of the sensor node $s \in P \cup Q$ is exhausted.

The energy consumption of sensor node s per unit of time $C(s)$ is defined, as follows:

$$\begin{aligned} C(s) = (&Sens() + Recep((D+H) \times s.desc) \\ &+Trans((D+H) \times (s.desc+1), |s - s.send|)) \\ &\times I + Listen(1) \end{aligned} \tag{12}$$

where $s.desc$ is the number of sensing nodes except for s in the subtree of the data collection tree rooted on s, $s.send$ is the destination node of data transmission of s, H is a header length of a packet. Formula (12) represents that every unit of time, each node consumes power for sensing, receiving, and transmitting data I times and for listening packet reception. The node consumes $Recep((D+H) \times s.desc) \times I$ power for receiving packets since it has $s.desc$ nodes as its descendants and the each packet's size is $D+H$. It consumes $Trans((D+H) \times (s.desc+1), |s - s.send|)) \times I$ power for transmitting data to node $s.send$.

III. Algorithm

In this section, we describe the algorithm to solve the problem defined in Section II-C.

A. Overview

Our algorithm decides the destinations of mobile sensor nodes and a tree called a *data collection tree* that connects all sensor nodes to sink node Bs by multi-hop paths for data collection. Whenever the battery of any sensor node is newly exhausted, our algorithm is applied, as shown in Section II-C.

The proposed GA-based algorithm for calculating the positions of mobile nodes and a data collection tree is supposed to be executed at the initial deployment time. The lifetime of the whole system ends when k-coverage of the target field is unable to be maintained.

B. Algorithm details

GA is a well-known meta-heuristic algorithm[8]. The following is its basic procedure.

1) **Generation of initial candidate solutions**: N candidate solutions are randomly generated.
2) **Evaluation**: Objective function for each candidate solution is evaluated to grade each candidate solution.
3) **Selection**: N candidate solutions with better evaluations are selected.
4) **Crossover**: New candidate solutions are generated by mixing two randomly selected candidate solutions.
5) **Mutation**: Part of candidate solutions are randomly mutated.
6) **Check termination**: If the termination condition is met, the candidate solution with the highest evaluation is output as the solution. Otherwise, go to Step 2.

Below, we show our algorithm for each GA operation.

Encoding of candidate solution: To apply a GA, each candidate solution has to be encoded, and the way of encoding sometimes greatly affects the algorithm performance. The coding in the proposed algorithm is shown in Fig. 1. Each candidate solution contains positions for $|Q|$ mobile nodes and

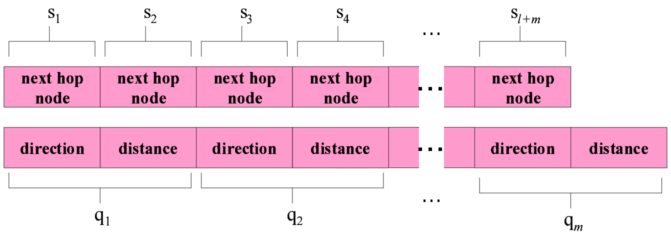

Fig. 1. Encoding of Candidate Solution

the structure of the data collection tree consisting of $|P \cup Q|$ sensor nodes. The positions for the mobile sensor nodes are represented in polar coordinates to avoid generating impossible destinations of mobile sensor nodes. A data collection tree is represented by a set of node IDs.

Generation of initial candidate solutions: Initial candidate solutions are made from random variables. Angles and distances of mobile nodes are uniformly assigned distributed random values between 0 and 2π, and 0 and $Dist$, respectively (here, $Dist$ is a constant and typically set to the longest movable distance in the target field). As an initial parent node for each node, a node geographically closer to the sink node is randomly selected. For efficiency, three candidate solutions are added to the initial candidate solutions whose collection trees are made using the minimum cost spanning tree method where an edge cost is the square of the distance, the balanced edge selection method proposed in Sect.III-D, and a method that directly connects all sensor nodes to the sink node.

Evaluation: The evaluation of each candidate solution verifies how long the target sensing field is k-covered by a simulation of WSN data transmission. The k-coverage duration is between the time when all mobile sensor nodes arrive at their new positions and the time when k-coverage cannot be maintained due to battery exhaustion of some nodes. If the decoded data collection network does not form a tree, the resulting evaluation is 0.

Strictly checking the k-coverage of the field is very expensive, and in the proposed algorithm, a sufficient condition for k-coverage is verified as described in Section III-E.

Genetic operators: In our proposed method, we adopted roulette selection, an elite preservation strategy, uniform crossover, and mutation per locus. For uniform crossover, we treated each combination of angle and distance for a mobile sensor node as a gene. For mutation, random value is overwritten to a randomly selected locus.

Termination condition: The algorithm stops after a constant number of generations (one generation corresponds to one iteration of the GA algorithm in Section III-B). In the experiment, we set 20 generations as the constant.

C. Local search technique

Our proposed method uses the local search technique in addition to GA to improve the quality of solution.

For each mobile node $q \in Q$, we give moving destination randomly in a circle (radius is $1[m]$) centered on q. If WSN lifetime improves when all mobile nodes move to the destination, they move actually and are given new destinations. If it is not improved, this algorithm terminates.

D. Balanced edge selection method

The nodes near the sink node tend to consume more battery by forwarding the data transmitted from other nodes. In the balanced edge selection method, we first decide the set of nodes called *first-level nodes* directly connecting to the sink node. Next, we connect the remaining nodes to the first-level nodes one by one. The idea to select the first level nodes is as follows.

Step-1: The first level nodes is decided by testing Step-2 for every number of the nodes from 1 to $|P \cup Q|$ so that the maximum power consumption by all the first level nodes is minimized. Here, we select each node in the increasing order of the distance from the node to the sink node.

Step-2: Data sent from the remaining nodes (other than the first-level nodes) must be forwarded through one of the first-level nodes to the sink node. Thus, the remaining nodes are distributed among the first-level nodes so that the power consumption is balanced among the first-level nodes. Here, the power consumption of each first-level node is estimated by the number of assigned nodes and the distance to the sink node.

Next, for each of the first-level nodes and the remaining nodes assigned to the node, we apply the above Step-1 and Step-2, recursively.

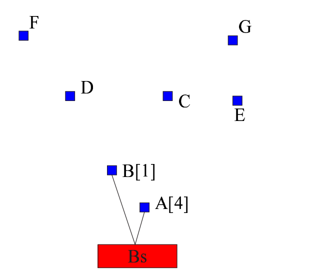

Fig. 2. Balanced Edge Selection Algorithm Repeats 1 time

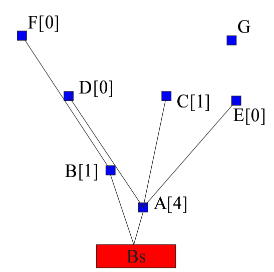

Fig. 3. Balanced Edge Selection Algorithm Repeats 3 times

We will explain how the algorithm works using an example. Fig. 2 depicts the situation just after the first-level nodes A and B have been decided. In the figure, 'A[4]' means that the node A has been assigned 4 remaining nodes. Here, node A is closer to the sink Bs than node B, A has been assigned more remaining nodes. We suppose that A and B have been assigned {C, D, E, G} and {F}, respectively.

Next, the algorithm is recursively applied, and the second-level nodes are decided as shown in Fig. 3. Among nodes C, D, and E, D is closest to node A. Then, finally node G is assigned to node C, and the data collection tree completes.

E. Algorithm for checking k-coverage

Geometrically verifying whether any points of the target sensing field is contained by at least k sensor nodes' sensing

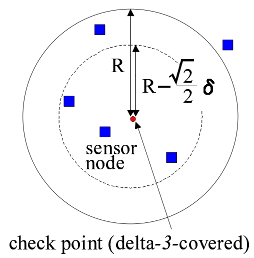

Fig. 4. Condition of delta-k-coverage

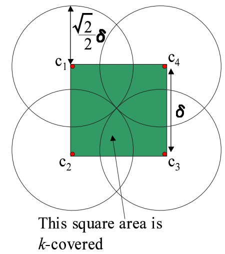

Fig. 5. Checking k-coverage by delta-k-coverage

ranges is very difficult.

In [4], Wang et al. proposed a sufficient condition for k-coverage, where the target field is divided into squares whose diagonals have the same lengths as the sensing radius to check if there is at least k sensor nodes in each square.

We propose a looser sufficient condition for the k-coverage of the target sensing field. In our method, we put checkpoints on grid points at intervals of δ in the target sensing field, and only check if each checkpoint is k-covered. However, even if all checkpoints are k-covered, some points between checkpoints may not be k-covered. The smaller δ is, the more the judgement accuracy improves. The judgment accuracy worsens when δ is too large. For $\delta < \sqrt{2}R$, we define delta-k-coverage which is a sufficient condition of k-coverage of the target field.

Definition 1

Checkpoint c is delta-k-covered if a circle whose center and radius are c and $R - \frac{\sqrt{2}}{2}\delta$, respectively, includes at least k nodes.

Fig. 4 shows that a checkpoint is delta-3-covered.

Theorem 1

Given checkpoints on grid points at intervals of δ ($\delta < \sqrt{2}R$) in a given field[1], if each checkpoint is delta-k-covered, then the field is k-covered.

Proof

As shown in Fig. 4, if checkpoint c is delta-k-covered, then any points in the circle with radius $\frac{\sqrt{2}}{2}\delta$ centered at c are k-covered. Thus, as shown in Fig. 5, for neighboring checkpoints c_1, c_2, c_3, and c_4, if all are delta-k-covered, any points in the square formed by those checkpoints are k-covered. Therefore, Theorem 1 holds.

Theorem 1 only provides a sufficient condition as k-coverage. If we use a smaller value for δ, the condition is closer to the necessary and sufficient condition for k-coverage. However, the smaller value of δ will cause more checkpoints to be checked by delta-k-coverage. In our experiment in Section IV, $\frac{\delta}{R} = \frac{1}{10}$ is used.

IV. Experimental validation

In this section, we show simulation results to validate the usefulness of our proposed method.

First, in order to evaluate the overall performance of our proposed method, we have measured the WSN operation time during which the whole target field is k-covered (we call the time *k-coverage duration*, hereafter), and compared it with the performance of other conventional methods including Wang's method[4], for several simulation configurations.

Next, in order to evaluate the efficiency of our delta-k-coverage judgment method, we measured and compared the performance of our method with Wang's method [4] in terms of accuracy of k-coverage judgment and its calculation time.

As a common configuration among the simulations, we used the parameter values shown in Table I by referring to existing literature.

TABLE I
Common configuration for experiments

Parameter	Value
Initial energy amount of each node	$s.energy$ = 32400 J (two AA batteries)
Power consumption coefficient for data processing	E_{elec} = 50 nJ/bit (by referring to [7])
Power consumption coefficient for signal amplification	ϵ_{amp} = 100 pJ/bit/m^2 (by referring to [7])
Power consumption exponent	n = 2 (by referring to [7])
Power consumption coefficient for moving	E_{move} = 8.267 J/m (by referring to [9])
Power consumption coefficient for sensing	E_{sens} = 0.018J/bit (by referring to [7])
Power consumption on idle time	E_{listen} = 0.025 J/s (by referring to [10])
Radius of sensing range of each sensor	R= 20m (by referring to [10])
Degree of coverage in the target field	k = 3
Size of data for sensed information	D= 128bit (by referring to [11])
Sensing frequency	0.1Hz (by referring to [11])

Parameters of GA are determined by preliminary experiments as follows: the number of solution candidates, the number of generations, crossover rate and mutation rate are 20, 20, 1, and 0.01, respectively.

A. *k-Coverage duration*

We have compared k-coverage duration of our proposed method with conventional methods named as follows: (i) *Proposed Method* which uses all techniques in Section III; (ii) *No Balancing Method* which randomly generates data collection trees as initial solution candidates in our method; (iii) *Static Method* which prohibits movement of mobile nodes in our method; and (iv) *Wang+Balancing Method* which decides the new positions of the mobile nodes by Wang's Method[4] and constructs a data collection tree by the balanced edge selection method and GA.

The configuration of this experiment other than Table I is provided as follows.

- Field size: 50m × 50m, 100m × 50m, and 100m × 100m
- Position of the sink node : around the south (bottom) end in the field
- Number of sensor nodes: 100, 200, and 300
- Proportion between numbers of static and mobile nodes: 25% and 75%
- Coverage degree: $k = 3$

[1]Note that outermost checkpoints must surround the target field.

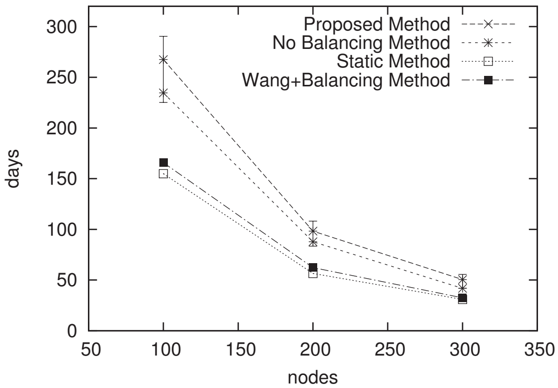

Fig. 6. 3-coverage duration

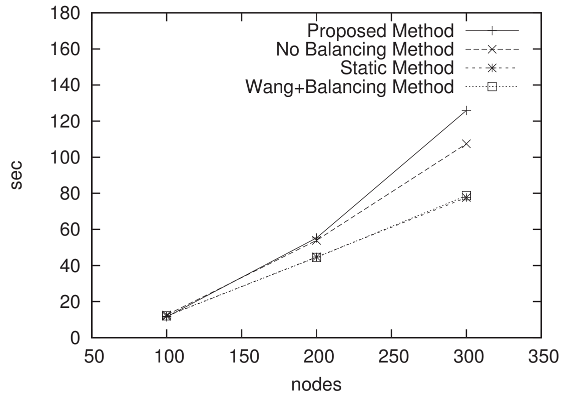

Fig. 7. 3-coverage computation time

Note that the size of the target field should be appropriately decided so that the field can be sufficiently k-covered by a given number of nodes and coverage degree k. Thus, we used field size $50\text{m} \times 50\text{m}$ with 100 nodes for the basic case, and enlarged the field size proportionally to the number of nodes. In the experiment, the initial positions of nodes are given by uniform random variables.

We show simulation results in Figs. 6 and 7 for 3-coverage. These results are average of 30 trials.

Fig.6 shows that two Proposed Methods (Balance and Random) outperform Static Method to a great extent, independently of the number of nodes. The reason is that finding the appropriate positions of mobile nodes in a wide area greatly affects the performance. Wang+Balancing Method was not so different from Static Method. Initially, the field was k-covered by sensor nodes in all methods. In many cases, however, Wang's k-coverage condition was not satisfied. Then, Wang+Balancing Method moved mobile nodes to the new positions so as to satisfy Wang's k-coverage condition. When a node exhausted its battery, Wang+Balancing Method often could not find the new positions of mobile nodes satisfying Wang's k-coverage condition.

The figure also shows that Proposed Method achieves better performance than No Balancing Method. Thus, our proposed balanced edge selection algorithm is effective in extending the k-coverage duration. In the figure, we see that the k-coverage duration of all methods decrease as the number of nodes increases. The reason is that the nodes that directly connects sink node Bs have to forward more data transmitted from their upstream nodes as the number of nodes increases, even though mobile nodes move closer to the sink node to help forwarding the data. In Fig. 6, the best and worst values of 30 trials by our algorithm were also shown. The difference of k-coverage duration of our algorithm was in the range from 84% to 109% compared with the average. We see that our algorithm does not output the solution with extremely bad performance.

Fig. 7 shows the computation time of each method. Proposed Method takes about 120 second in the case of 300 nodes for $k = 3$. This shows that it is possible to operate our method actually.

TABLE II
THE NUMBER OF OCCURRENCES THAT THE FIELD IS JUDGED AS k-COVERED (OUT OF 100 SIMULATION RUNS)

	k=1	k=2	k = 3
Wang's Method[4]	93	44	4
Proposed Method ($\delta = 0.5$m)	100	100	100
Proposed Method ($\delta = 1.0$m)	100	100	100
Proposed Method ($\delta = 2.0$m)	100	100	100
Proposed Method ($\delta = 4.0$m)	100	100	100
Proposed Method ($\delta = 8.0$m)	100	100	100
Proposed Method ($\delta = 12.0$m)	100	100	97
Proposed Method ($\delta = 16.0$m)	96	82	48
Proposed Method ($\delta = 20.0$m)	39	1	0

B. *Efficiency of k-coverage judgment algorithms*

We have measured and compared the accuracy and computation time of our delta-k-coverage judgment method and Wang's method[4]. Both methods are based on their own sufficient conditions for checking k-coverage. Thus, if one of the methods judges affirmatively, then the field is actually k-covered. Conversely, even if both methods judge negatively, it is not always the case that the field is not k-covered. The higher the ratio to judge that the field is k-covered is, the higher the judgment accuracy is.

In this experiment, 300 static nodes are randomly deployed in the $100\text{m}\times100\text{m}$ field. In this case, the field is almost always 3-covered. Therefore, it is expected to judge that 1, 2, and 3-coverage of the field are satisfied in all trials. We conducted the above simulation 100 times and measured the number of the occurrences that the field is judged to be k-covered out of the 100 simulations. Note that on some occurrences, the whole field is not actually k-covered since node positions are randomly decided.

We conducted the above simulations by changing the value of δ from 0.5m to 23.5m by 0.5m step for our delta-k-coverage judgment method, while the diagonal length of all squares in Wang's method is fixed to $10\sqrt{2}$m, which is the sensing radius of sensor nodes, and cannot be changed.

The experimental results on measured accuracy is shown in Table II. Note that Table II shows part of the results for some important δ values.

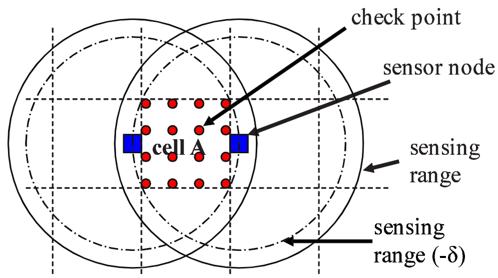

Fig. 8. Example of Misjudge by Wang's Method

Table II suggests that our delta-k-coverage judgment method is better than Wang's method for all numbers for k when δ is no bigger than 16m. The difference becomes bigger as k increases. Especially, when δ is no bigger than 12m, our algorithm almost perfectly judged k-coverage of the field, whereas Wang's method judged that only 4 occurrences out of 100 was 3-covered. Fig. 8 shows the example of node positions such that the difference of the judgement between our method and Wang's method is extreme. In Wang's method, the field is divided into grids at intervals of $\frac{R}{\sqrt{2}}$, and the number of coverage is the number of the sensor nodes in each grid. In Fig. 8, cell A is 2-covered actually. Wang's method judges that cell A is not covered because there is no sensor node in cell A. On the other hand, our method judges that cell A is 2-covered, since each check point is delta-2-covered. Wang's method takes a constant computation time around 0.13ms, while our method takes longer computation time, which is inversely proportional to δ, for example, 159ms for $\delta = 1$m, 2ms for $\delta = 8$m, and 1ms for $\delta = 12$m.

As a result, our algorithm takes longer computation time, however it is much more practical since it is adjustable depending on the required accuracy of k-coverage judgment within allowable computation time.

C. Influence of mobile nodes ratio for k-coverage lifetime

It is obvious that using n mobile nodes will achieve longer k-coverage lifetime than using n static nodes. However, a mobile node is much more expensive than a static node. In order to investigate the influence of mobile nodes ratio to all nodes, we measured k-coverage lifetime for 100, 200, and 300 sensor nodes, changing the mobile nodes ratio from 0% to 100% by 5% step.

We show the results in Fig. 9. The results are average values of 30 simulations. Fig. 9 suggests that the k-coverage lifetime increased sharply in the ratio from 0% to 25%, and loosely from 25% to 100%. That means about 25% ratio of mobile nodes will be the best when we consider the deployment cost.

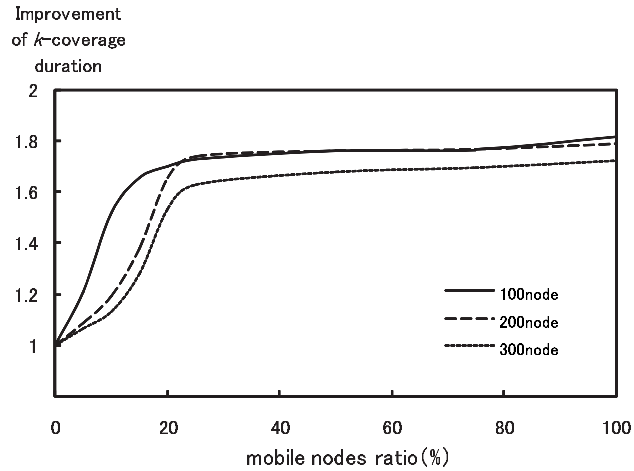

Fig. 9. Improvement of k-coverage Duration for Mobile Nodes Ratio

V. CONCLUSION

In this paper, we formulated a problem of maximizing the k-coverage lifetime of the field in a WSN environment with mobile and static sensor nodes. We proposed a GA-based algorithm to decide the positions of mobile sensor nodes and to construct a data collection tree with balanced power consumption for communication among nodes. We also defined a new sufficient condition for k-coverage based on checkpoints and proposed an algorithm to accurately judge k-coverage in reasonably short time.

Through computer simulations, we confirmed that our method improved k-coverage lifetime to about 140% to 190% compared with other conventional methods for 100 to 300 nodes. Also, we confirmed that the best cost-performance is achieved when the mobile nodes ratio is about 25%.

REFERENCES

[1] Tang, X. and Xu, J.: "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," *Proc. of Infocom2006*, pp. 1–12 (2006).

[2] Heinzelman, W.R., Chandrakasan, A., and Balakrishnan, H.: "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of the 33rd Hawaii Int'l. Conf. on System Sciences (HICSS 2000)* (2000).

[3] Mei, Y., Xian, C., Das, S., Charlie Hu, Y., and Lu, Y.: "Repairing Sensor Network Using Mobile Robots," *Proc. of IEEE Workshop on Wireless Ad hoc and Sensor Networks (WWASN 2006)*(2006).

[4] Wang, W., Srinivasan, V., and Chua, K. C.: "Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks," *Proc. of MobiCom 2007*, pp. 39–50 (2007).

[5] Srinivas A., Zussman G. and Modiano E.: "Mobile Backbone Networks - Construction and Maintenance," Proc. of ACM MobiHoc'06, pp.166–177 (2006).

[6] Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A. and Sukhatme, G. S.: "Robomote: enabling mobility in sensor networks," *Proc. of Fourth Int'l. Symp. Information Processing in Sensor Networks (IPSN 2005)*, pp. 404–409 (2005).

[7] Wang, G., Cao, G., La Porta, T., and Zhang, W.: "Sensor Relocation in Mobile Sensor Networks," *Proc. of Infocom2005*, Vol. 4, pp. 2302–2312 (2005).

[8] Goldberg, D. E.: "Genetic Algorithms in Search, Optimization, and Machine Learning," *Addison-Wesley Publishing Company, Inc.* (1989).

[9] Sibley, G. T., Rahimi, M. H. and Sukhatme, G. S.: "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," *Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA 2002)*, Vol. 2, pp. 1143–1148 (2002).

[10] Ganeriwal, S., Kansal, A., and Srivastava, M. B.: "Self aware actuation for fault repair in sensor networks," *Proc. of Int'l Conf. on Robotics and Automation (ICRA2004)* (2004) .

[11] Kamimura, J., Wakamiya, N., and Murata, M.: "Energy-Efficient Clustering Method for Data Gathering in Sensor Networks," *Proc. of First Workshop on Broadband Advanced Sensor Networks (BaseNets2004)* (2004).