Maximizing Lifetime of Wireless Sensor Networks with 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
Ikoma, Nara 630-0192, Japan e-mail: {ryo-k,yasumoto,ito}@is.naist.jp
† Graduate School of Information Sciences, Hiroshima City University
Hiroshima 731-3194, Japan e-mail: yoshi-m@hiroshima-cu.ac.jp
‡ Department of Information Processing and Management, Shiga University
Hikone 522-8522, Japan e-mail: shibata@biwako.shiga-u.ac.jp
§ Japan Science and Technology Agency, CREST

Abstract

In this paper, we propose a new method utilizing mobile sensor nodes to prolong WSN lifetime while maintaining a large sensing area. We formulate this problem as a problem in finding the best positions for mobile sensor nodes to maximize the integral of the sensing area for the remaining lifetime of WSN. This problem implies the Euclidean Steiner Tree Problem, and is thus an NP-hard problem. We developed a GA (genetic algorithm)-based algorithm to find a near-optimal solution. Through simulations, we confirmed that the proposed method achieves better performance than a general local-search based approximation method for 100-node WSNs. We also confirmed that our method can find solutions within 5.1% difference from the optimal solution for 10-node WSNs.

1 Introduction

Recently, many efforts have been made to study wireless sensor networks (WSNs) and their applications such as collections of environmental information and tracking intruders. Typical WSN applications include environmental monitoring of temperature, humidity, acid degree and so on in agricultural areas, tracking of moving objects for border guard, and investigation of unexplored areas such as astronomical objects and abyssal areas. Sensor nodes used in the WSNs are expected to have a long lifetime, but they are battery-limited, and some of them may fail as time passes. Many research efforts have been made to extend lifetime of a system while covering a wide sensing area in WSN by reducing the frequency of communication [1], moving mobile sensor nodes to replace failed nodes [2, 3], and so on.

In this paper, we propose a new method to extend lifetime while maintaining a wide sensing area of WSN by introducing mobile sensor nodes. In the proposed method, we realize both long lifetime and wide sensing area by moving mobile sensor nodes to appropriate positions not only to cover the unsensed areas but also to help static sensor nodes send packets with lower power. We have formulated the problem as a combinatorial optimization problem. This problem is NP-hard, since it implies the Steiner Tree Problem as a special case. We have developed a heuristic algorithm based on GA to solve this problem in practical time.

We have evaluated the performance of our method by comparing the solutions of our GA-based algorithm with those of a general local-search based approximation method. As a result, we confirmed that our method outperforms the other method in terms of lifetime, sensing area, and calculation time for 100-node WSNs. We also confirmed that the solution calculated by our algorithm differs only 5.1% from the optimal solution for 10-node WSNs.

2 Related Work

In WSN, each sensor node not only senses environmental information nearby but also forwards the information sensed by other sensor nodes. Since sensor nodes work by battery, their computation power and the power used for sensing and communication are also limited [4]. In static WSN, the functionality of the whole network can be dam-
aged considerably by failure or battery shortage of a few nodes. Some existing studies aim at prolonging the lifetime of WSN as much as possible. On the other hand, when placing sensor nodes in a certain environment, overlaps of sensing ranges of multiple sensors, communication distance between nodes, and tolerance against node failure have to be taken into account.

In [1], Tang, et al. proposed a method to extend the lifetime of WSN by making nodes with less remaining battery amount transfer data less frequently.

In [5], Wang, et al. proposed a method to extend the lifetime of WSN by introducing mobile sensor nodes. Assuming that all sensor nodes are movable, this method moves each mobile node so that a node covers an unsensed area, or replaces another failure node (see Fig. 1). Since this method assumes all nodes to be mobile, the total cost tends to be high.

In [2], Mei, at al. proposed a method to repair a WSN partitioned due to some failure nodes, by utilizing mobile nodes. In this method, WSN nodes consist of both static and mobile nodes. However, this method does not address the lifetime extension problem.

In [3], Wang, et al. proposed a method to maximize the whole sensing area from the viewpoint of k-coverage in an environment where mobile nodes and static nodes exist. The purpose of this method is to maximize the whole sensing area by considering the cost of moving mobile nodes. However, packet transmission cost is not considered.

To the best of our knowledge, there is no study to improve both sensing area and lifetime of WSN simultaneously by utilizing mobile sensor nodes. Our contribution is a solution to this problem.

3 Target Problem and Formulation

In this section, we give the target WSN model and formulate the problem of moving mobile sensor nodes to appropriate positions to maximize lifetime of the WSN while maintaining a high coverage of the target field. The notations used in this section and the following sections are summarized in Table 1.

3.1 Application Model

A WSN consisting of multiple sensor nodes and a base station is considered, where the WSN deployment field (also called target field) has no obstacles. We denote a base station and its geographical position by Bs and Bs.pos, respectively.

We suppose an application in which a base station Bs periodically aggregates sensed values such as temperature or humidity from all sensor nodes by multi-hop communication along a tree (called data aggregation tree) spanning all the sensor nodes.

Sensing frequency is denoted F and given as a constant. The size of a data packet which contains data sensed by each sensor node is denoted Size.

In data aggregation by multi-hop communication, each intermediate node s in the data aggregation tree waits until receiving data from all of its child nodes, merges the received data and the data sensed by s itself into one data, and forwards the merged data to s’s parent node along the tree.

We assume that the size of the merged data does not exceeds Size and can be accommodated in a packet.
The sensing region covered by each node is a circle with radius $R$ whose center is the sensor node’s position. The sensing area covered by all the sensor nodes is defined as a union of the circles corresponding to all sensor nodes’ sensing regions.

We denote by $A_{\text{min}}$ the minimum area size which must be covered in the target field during the WSN operation time. If 100% coverage is required, $A_{\text{min}}$ should be set equal to the area of the target field.

The base station $Bs$ knows the coordinates of all sensor nodes in advance using GPS or aerial photographs.

### 3.2 Sensor Node Model

WSN nodes are usually powered by batteries, and in many applications, these batteries are difficult to recharge during WSN operation time. Thus, we assume that each WSN node has a battery with finite energy. Also, we assume that each node has a wireless communication device with variable transmission power so that transmission range can be changed. We define the power $\text{Trans}(k, d)$ required to transmit $k$[bit] for $d$[m] by expression (1) which is introduced in [6].

$$\text{Trans}(k, d) = E_{\text{elec}} \times k + \epsilon_{\text{amp}} \times k \times d^n$$  \hspace{1cm} (1)

Here, $E_{\text{elec}}$ is a constant representing the power required for information processing. $\epsilon_{\text{amp}}$ is also a constant representing the power for amplification. The value of $n (\geq 0)$ is defined by the properties of the antenna. If we could use a perfect directional antenna in a vacuum, $n$ would be zero, whereas if we use an omni-directional antenna, $n$ would be two. In reality, the value is somewhere between 0 and 2.

Sensor nodes can be classified into static nodes and mobile nodes. Static nodes cannot be moved from their original positions. Mobile nodes, such as Robomote [7] can move on their wheels.

Mobile nodes consume battery power not only by communication, but also by movement. We define the power $\text{Move}(d)$ required for $d$[m] to move by expression (2), which is introduced in [5]. Here, $E_{\text{move}}$ is a constant.

$$\text{Move}(d) = E_{\text{move}} \cdot d$$  \hspace{1cm} (2)

Each mobile node can move at up to $V$ [m/sec], where $V$ is a constant.

Let $P = \{p_1, p_2, \ldots, p_t\}$, and $Q = \{q_1, q_2, \ldots, q_m\}$ denote the set of static nodes and the set of mobile nodes, respectively. For each node $s \in P \cup Q$, $s_{\text{pos}}, s_{\text{energy}}$ and $s_{\text{range}}$, denote, respectively, the current position, the battery amount, and the sensing range of the node $s$. Note that $s_{\text{range}}$ corresponds to a circle with radius $R$ whose center is $s_{\text{pos}}$.

### 3.3 Network Model

The radio transmission range of a node is a circle whose center is the node’s position, but each node can change its transmission radius by deciding transmission power $\text{Trans}(k, d)$ appropriately. There is no interference between wireless communications by nodes, that is, the success rate of transmission is 100% if a receiving node is within the transmitting node’s transmission range, and 0% if the node is outside the range.

### 3.4 Problem Definition

Suppose that the best positions of mobile nodes are refreshed every $t$ seconds. Given $t$, $s_{\text{pos}}, s_{\text{energy}}$ and $s_{\text{range}}$ for each sensor node $s \in P \cup Q$, the position of a base station $Bs_{\text{pos}}$, and constants $E_{\text{elec}}, \epsilon_{\text{amp}}, n, E_{\text{move}}, V, \text{Size}$, and $F$, our target problem is to find the new position $s_{\text{newpos}}$ of each mobile sensor node $s \in Q$, which satisfies the conditions (3) and (4).

$$\forall s \in Q, |s_{\text{newpos}} - s_{\text{pos}}| \leq V \cdot t$$  \hspace{1cm} (3)

$$\bigcup_{s \in T} s_{\text{range}} \geq A_{\text{min}}$$  \hspace{1cm} (4)

The condition (3) means that mobile nodes can move for up to $V \cdot t$ meters in $t$ seconds and thus the new positions are limited to within this distance.

The condition (4) means that the sensing area covered by all sensor nodes must be no less than $A_{\text{min}}$.

According to the new positions of all sensor nodes, we compute a data aggregation tree $T$ which connects all the nodes with the smallest sum of link costs, where link cost is defined as $d^n$ for an edge $(u, v)$ with length $d$ in $T$.

Generally, there is more than one combination for positions of all mobile nodes which satisfy conditions (3) and (4). In this paper, we adopt the objective function defined by expression (5) to extend the lifetime of the WSN by maximizing the sum of the remaining battery amounts of all sensor nodes:

$$\max \sum_{(u, v) \in T} (v_{\text{energy}} - \text{Trans}(\text{Size}, |u_{\text{pos}} - v_{\text{pos}}|) \cdot t \cdot F)$$

$$- \sum_{s \in Q} \text{Move}(|s_{\text{pos}} - s_{\text{newpos}}|)$$  \hspace{1cm} (5)

subject to constraints (3) and (4).

### 4 Proposed Method

In this section, first we show that the mobile sensor location problem defined in Sect. 3 is an NP-hard problem.
Then, we propose a heuristic algorithm based on a genetic algorithm (GA) to find a near-optimal solution within practical time.

### 4.1 Euclidean Steiner Problem

Given a set of fixed vertices $V$ in a plane, the Euclidean Steiner Tree problem is the problem of finding the tree with minimal Euclidean length spanning all vertices in $V$, allowing for the addition of auxiliary vertices called Steiner vertices. This problem is known as an NP-hard problem [8].

If $n = 1$, $|Q| = |P| - 2$, and the battery amount required for movement of all mobile sensor nodes is much smaller than that required for packet transmission of all sensor nodes for $t$ seconds, our target problem defined in Sect. 3 corresponds to the Euclidean Steiner Tree problem of finding positions of $|Q|$ mobile sensor nodes which minimize the total Euclidean length of the tree. Therefore, our problem is also NP-hard since it implies the Euclidean Steiner problem as a special case.

### 4.2 Algorithm

The proposed algorithm consists of a data aggregation tree construction algorithm for given positions of all sensor nodes and a GA-based algorithm to decide the best positions of mobile sensor nodes.

#### 4.2.1 Construction of a Data Aggregation Tree

We use the Dijkstra method [9] to construct a data aggregation tree connecting all sensor nodes from the base station (root node), which has the minimum sum of the squares of the lengths of all links. We now briefly describe the algorithm to construct the tree, assuming $n$, the constant representing antenna property, to be 2.

We use two variables $S$ and $T$ representing a set of nodes and a set of links, respectively for the tree construction algorithm.

For two nodes $u$ and $v$ in a tree $T$, let $path(u, v, T)$ denote the set of links connecting $u$ and $v$ in $T$. For a path $path$, let $cost(path)$ denote the sum of $|l|^2$ for each link $l$ in $path$.

First, $T$ and $S$ are initialized to the empty set and the set of all sensor nodes except for the nodes with failure or battery exhaustion, respectively.

Second, we select a node $s \in S$ such that $|(s, Bs)|^2$ is the minimum, add a link $(s, Bs)$ to $T$, and remove $s$ from $S$.

Next, we select a node $s' \in S$ and a node $s''$ in $T$ such that $|(s', s'')|^2 + cost(path(s'', Bs, T))$ is the smallest, add a link $(s', s'')$ to $T$, and remove $s'$ from $S$.

This process is repeated until $S$ becomes empty.

### 4.2.2 Genetic Algorithm

GA is a meta heuristic algorithm which prepares multiple solution candidates called individuals and evolves the individuals by repeatedly applying genetic operators such as crossover and mutation to them and selecting only better individuals for the next generation[10]. The general procedure of GA is as follows:

1. Two sets with capacities of $N$ individuals, called current-generation and next-generation, are prepared.
2. As the initial population (the set of all individuals), $N$ individuals are generated in the current-generation set.
3. With the fitness function given in advance, the fitness value of each individual in the current-generation set is calculated.
4. For the next generation set, new individuals are generated by repeatedly applying the following GA operators to the current generation individuals until the number of the individuals becomes $N$.
   - Crossover: selecting two parent individuals and generating child individuals by mixing chromosomes (i.e., lists of genes) of the parents
   - Mutation: changing part of the chromosome of each individual at some probability
   - Selection and copy: selecting individuals with higher fitness values and copying them to the next generation set
5. All individuals in the next-generation set are copied to the current-generation set and the set is emptied.
6. The operations after step 3 is repeated until satisfying the termination condition given in advance. Finally, the individual with the highest fitness value in the current-generation set is output as a solution.

Below, we explain the details of our GA-based algorithm for solving the problem in Sect. 3.

#### Encoding

Each individual has a chromosome consisting of $|Q|$ genes, each of which corresponds to the target destination of a mobile node. We encode the destination of each mobile node as a polar coordinate. The polar coordinates are tuples consisting of distance and angle. So, it is easy to check the moving distance and suppress mobile nodes to move to unreasonably far places. We represent each angle and distance as a 32bit floating value.
Generating Initial Population

Initial population (set of all individuals) is generated based on random values. Moving angle is a random value between 0 and $2\pi$, and moving distance is a random value between 0 and $D$, where $D$ is a constant value. The number of individuals should be carefully set to the appropriate number considering the number of mobile nodes and other factors. In our preliminary experiments, we confirmed that 30 individuals are sufficient for convergence when the number of mobile nodes is less than 30.

Calculation of Fitness Value

In the proposed method, when failure or battery exhaustion of a node occurs, the data aggregation tree is re-constructed, and the whole sensing area and the required power for each node are re-calculated. The data aggregation tree is constructed by the method described in Sect. 4.2.1. We define that the fitness value of each individual is the objective function (5).

Genetic Operators

For efficient calculation of mobile node destinations, our algorithm uses the elite preservation strategy and the roulette selection technique. Elite preservation is a strategy which always preserves the individual with the highest fitness value for the next generation set. The roulette selection is a technique which repeatedly applies an operation for selecting an individual in the current generation set with a probability $\frac{\text{individual's fitness value}}{\text{sum of all individuals' fitness values}}$ for the next generation set until the set is filled with $N$ individuals. The roulette selection has an advantage in terms of computation power since it does not require sorting all the individuals in the order of their fitness values.

For the other genetic operators, our algorithm uses the uniform crossover technique, which generates child individuals by taking each gene from either of parents, and the mutation per gene technique, which changes the value of each gene with a random value at a specified probability.

Termination Condition

The termination condition of our algorithm is specified as the number of generations (i.e., the number of times we repeat the procedure of Sect. 4.2.2). In our preliminary experiments, we confirmed that 30 generations are sufficient for convergence.

4.2.3 Local Search

For each gene of the elite individual, our algorithm generates a new destination at random between 0 and $2\pi$ for angle and between 0 and 1 $[m]$ for distance, and checks if the fitness value with the destination improves or not. If the fitness value improves, the generated destination is overwritten in the corresponding gene of the elite individual. Our algorithm repeats this process 100 times.

4.3 Updating the Positions of Mobile Nodes and the Data Aggregation Tree: How and When

We assume that the calculation of destinations for mobile nodes and the data aggregation tree is performed at the base station, and the results are delivered to all nodes by one-hop communication using sufficiently large transmission power. In general, failure or unexpected battery exhaustion of nodes occurs, which makes the current data aggregation tree inefficient. So, we solve this problem periodically or whenever failure or battery exhaustion of a node occurs.

After this problem is solved, the data aggregation tree should be changed to new one. Basically, we let all nodes start using the new tree after all mobile nodes reach their target positions.

5 Experimental Results and Discussion

In this section, we provide experimental results to validate the usefulness of the proposed method and we discuss the geometrically best mobile node positions.

5.1 Experimental Validation

In order to evaluate the performance and the efficiency of the proposed method, we implemented a WSN simulator which simulates communication and movement of sensor nodes. With the simulator, we measured the time integral of the whole sensing area. In order to simplify the measurement of the area size covered by all sensor nodes, we used the following approximation method: (1) putting checkpoints on grid points at intervals of $\frac{R}{\sqrt{2}}$ where $R$ is sensing radius of a sensor node; and (2) counting the number of checkpoints that are covered by at least one sensor node.

We also measured the computation time of our method to calculate the destinations of mobile nodes.

The snapshots of the node positions and of the data aggregation tree are shown in Figs. 2 and 3, where the larger rectangle at the bottom represents the base station, and blue boxes (node numbers 0 to 13) and yellow boxes (numbers 14 to 19) represent static and mobile nodes, respectively. Fig. 2 indicates the WSN before mobile node movement, and Fig. 3 shows the WSN after movement. The figures show that the mobile nodes move to their new positions.
Figure 2. Positions of Nodes Before Mobile Nodes Move

Figure 3. Positions of Nodes After Mobile Nodes Move

which make distances between nodes shorter for power saving.

In the experiments, we determined the parameter values as follows, referring to [5, 6, 7, 11, 12, 13, 14].

- Initial energy: $s.energy = 32400\, [J]$
- Energy consumption to carry out data transmission: $E_{elec} = 50\, [nJ/bit]$
- Energy consumption to expand radio coverage: $\epsilon_{amp} = 100\, [pJ/bit/m^2]$
- Multiplier of energy consumption: $n = 2$
- Energy consumption for movement: $E_{move} = 8.267\, [J/m]$
- Idle cost $e_{idle} = 0.025\, [J/s]$
- Maximum speed of mobile node: $V = 0.5\, [m/s]$
- Covered sensing region of one node: $s.range = 400\pi\, [m^2]$
- Sensing radius: $R = 20\, [m]$
- Amount of sensing data of one node: $data\_size = 128\, [bit]$
- Mean time between failure (MTBF): $8.2 \times 10^8\, [s]$
- Sensing frequency: $0.1\, [Hz]$

The initial amount of energy of each sensor node is set to the same value for all static and mobile nodes.

According to the assumptions in Sect. 3, each node can raise radio transmission power freely, and thus each node is capable of communicating with any other node in the target field.

Regarding data collection, data sensed by each node is collected along the data aggregation tree to the base station $B_s$. Even if a sensor node was in an idle state such as no transmission and no moving, it consumes the energy of $0.025\, [J/s]$ ($= e_{idle}$).

When failure occurs or a battery has run out at a node, destinations of mobile nodes are recalculated.

In the first experiment, we compared the proposed method with three other methods: (1) a static WSN method which uses only static nodes; (2) a local search method which uses only the local search algorithm in Sect. 4.2.3; and (3) a circumference search method which randomly generates 1000 destinations satisfying $|q_{newpos} - q_{pos}| \leq D$ (for each $q \in Q$), and outputs the one with the best fitness value. The circumference search method executes more evaluations than our proposed method, that is, 30 individuals $\times$ 30 generations $= 900$ evaluations.

We conducted the first experiment under the following configuration.

- Field size: $250\, [m] \times 250\, [m]$
- Area to be covered in the target field: $A_{min} = 125 \times 125\, [m^2]$
- Coordinates of base station: $B_s.\text{pos} = (125, 250)$
- Number of static nodes: $l = 75$
- Number of mobile nodes: $m = 25$

Initial coordinates of sensor nodes were given randomly within the field. Parameters given to GA are as follows: population size is 20, number of generations until termination is 30, crossover rate is 1, and mutation rate is 0.2. The averaged results over 30 trials are shown in Fig. 4.

Fig. 4 shows that our proposed method achieved about a 19.1% larger time integral of the sensing area than the static WSN method, and about an 8.5% better value than the
In terms of computation time, the local search method is the fastest, but the evaluation value of the obtained solution is about 14.2% worse than our proposed method.

In the second experiment, we compared the proposed method with an exhaustive search method in order to check how close the approximate solution obtained by our method is to the optimal solution. The exhaustive search method first checks all candidate solutions whose coordinates are integers, and uses the local search method to improve the best solution.

Configurations of the second experiment are as follows:

- **Field:** 50[m] × 50[m]
- **Area to be covered in the target field:** \(A_{min} = 25 \times 25\) [m²]
- **Coordinates of base station:** \(Bs.pos = (25, 50)\)
- **Number of static nodes:** \(l = 6\)
- **Number of mobile nodes:** \(m = 4\)

Initial coordinates of sensor nodes were given randomly within the field. Results are shown in Table 2.

Table 2 shows that the solution found by the proposed method is only 5.1% smaller than the optimal solution. We think that this is reasonable enough when we consider the calculation time of the exhaustive search method.

### 5.2 Discussion

From the viewpoint of minimizing power for communication, one may think that mobile nodes should move to, for example, the exact midpoint between two static nodes. However, if we also take into account power for movement, there are cases in which the exact midpoint is not the optimal position. As the distance between the midpoint and the position of a mobile node reduces, the ratio of the saved communication power to the power required for movement grows smaller. For example, in Fig. 5(b), moving for distance \(d\) shortens the communication distance by \(2(\sqrt{5}d - \sqrt{2}d) = 1.64d\), whereas in Fig. 5(a), moving...
the same distance shortens the communication distance by
only $2(\sqrt{2}d - d) = 0.83d$.

In Fig. 3, one may think that node 16 moved to a very
unlikely position, but actually this position is near optimal.

6 Conclusion

In this paper, we proposed a method to find locations
to which mobile sensor nodes are moved in a WSN de-
ployment field, in order to allow the WSN to collect data
sensed in a wide area for long duration. The distinctive
feature of the proposed algorithm is that it takes into ac-
count battery usage for data transmission, it adjusts a trans-
mission path by changing the positions of mobile nodes,
and it prevents/repairs disconnection of paths due to bat-
ery exhaustion. Through experimental validation, we con-
firmed that the proposed method shows only a
5.1% difference from the optimal solution.

As part of our future work, we will improve the al-
gorithm for computing optimal positions of mobile sen-
sor nodes and develop a method to move mobile nodes
by predicting future failure or battery exhaustion of nodes.
We will also develop a distributed algorithm to apply
our method to larger WSNs and regulate the move-
ment/positions of mobile nodes so that mobile nodes and
base station can communicate through multi-hop paths
within a certain radio transmission power.

We are also planning to test our method with actual de-
vices. We are now developing a wireless communication
testbed using LEGO mindstorms NXT [15].

References

Precision-Constrained Data Aggregation in Wireless
Sensor Networks,” Proc. of 25th IEEE Int’l Conf. on
Computer Communications (Infocom2006), pp. 1–12

“Repairing Sensor Network Using Mobile Robots,”
Proc. of IEEE Workshop on Wireless Ad hoc and Sen-
sor Networks (WAWSN 2006)(2006).

offs Between Mobility and Density for Coverage in
Wireless Sensor Networks,” Proc. of The Thirteenth
Annual Int’l Conf. on Mobile Computing and Net-

“MICA2: Wireless Measurement System,”
http://www.xbow.com/Products/Product_pdf_files/
Wireless_pdf/MICA.pdf.

[5] Wang, G., Cao, G., La Porta, T., and Zhang, W.; “Sen-
or Relocation in Mobile Sensor Networks.” Proc. of
24th Annual Joint Conf. of the IEEE Computer and
Communications Societies (Infocom2005), Vol. 4, pp.

nan, H.; “Energy-Efficient Communication Protocol
for Wireless Microsensor Networks,” Proc. of the 33rd
Hawaii Int’l. Conf. on System Sciences (HICSS 2000)
(2000).

[7] Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal,
A. and Sukhatme, G. S.; “Robomote: enabling mobil-
Information Processing in Sensor Networks (IPSN

for the Euclidean Steiner Tree Problem,” Journal of

[9] Dijkstra, E.W.; “A Note on two Problems in Connect-

timization, and Machine Learning,” Addison-Wesley

“Robomote: A Tiny Mobile Robot Platform for Large-
Scale Sensor Networks,” Proc. of IEEE Int’l Conf. on


“Energy-Efficient Clustering Method for Data Gath-
ering in Sensor Networks,” Proc. of First Work-
shop on Broadband Advanced Sensor Networks

“Self aware actuation for fault repair in sensor net-
works,” Proc. of Int’l Conf. on Robotics and Automa-

[15] The LEGO Group; “LEG0 Mindstorms NXT,”
http://mindstorms.lego.com/.