Distributed Clustering Method for Energy-Efficient Data Gathering in Sensor Networks

Abstract: By deploying wireless sensor nodes and composing a sensor network, one can remotely obtain information about the behavior, conditions, and positions of entities in a region. Since sensor nodes operate on batteries, energy-efficient mechanisms for gathering sensor data are indispensable to prolong the lifetime of a sensor network as long as possible. In this paper, we proposed a novel clustering method where energy-efficient clusters are organized in a distributed and self-organizing way through local communication among sensor nodes. Our method is based on an idea of ANTCLUST, a clustering algorithm which applies a colonial closure model of ants. Through simulation experiments, we showed that our method could gather data from more than 80% of the sensor nodes longer than other clustering methods by over 30%.

Keywords: sensor network, data gathering, clustering, energy-efficiency, biological system

1 Introduction

With recent advancements and developments in Micro Electro Mechanical System (MEMS) technologies, low-cost and low-power consumption wireless micro sensor nodes have become available. A sensor node has one or more sensors, a general purpose processor with limited computing power and memory, a radio transceiver, that operates on batteries. By deploying sensor nodes and composing a sensor network, one can remotely obtain information about behavior, conditions, and the position of entities in a region through a sink, called base station, where sensor data are gathered [1].

Since sensor nodes derive power from batteries, an energy-efficient data gathering mechanism is indispensable to observe the region as long as possible. A sensor node consumes its energy in monitoring its environment and receiving and sending radio signals. The amount of energy consumed in a radio transmission is proportional to the k-th power of the range of the radio signal propagation [2, 3]. Since the distance from sensor node to sensor node is shorter than from sensor node to the base station, it is energy-inefficient for all sensor nodes to send their data directly to a distant base station. Therefore, cluster-based data gathering mechanisms effectively save energy [2-6]. In cluster-based mechanisms, groups of neighboring sensor nodes form clusters. In each cluster, one representative node called a cluster-head gathers sensor data from its members and sends the collected data to a base station. Since cluster-heads consume more energy than cluster members in receiving sensor data from their members, processing received data, and sending aggregated data to the base station, the role of cluster-head must be rotated among sensor nodes.

There are several demands to a clustering method. First, a clustering method should be completely distributed because central control of hundreds or thousands of sensor nodes is not feasible. Second, clusters are needed to be geographically well distributed for well-balanced energy consumption among sensor nodes. Third, of course, a clustering method itself should be energy-efficient. Fourth, since sensor nodes are dynamically deployed, moved, and halted, a clustering method should be able to adapt to changes of sensor networks. Our goal is to propose a new clustering method to satisfy the above mentioned features, where sensor nodes autonomously form appropriate clusters through local communications among neighboring sensor nodes.

In biology, ants and other social insects construct clusters, i.e., colonies, parties, and cemeteries in self-organizing ways [7]. Taking inspiration from such biological systems, specifically based on an ant model of colonial closure, [8] proposed an algorithm, ANTCLUST, to solve data clustering problems. Ants recognize each other by exchanging a chemical substance. If they are similar, the ant is welcomed and treated as a member of the same nest. In ANTCLUST, two randomly chosen objects meet. Based on their similarity, a cluster is created, merged, or deleted. By repeating meetings, an appropriate set of clusters is eventually formed so that similar objects are accommodated in the same cluster.

In this paper, based on ANTCLUST, we propose a novel clustering method that organizes energy-efficient clusters through local interactions among sensor nodes. In our method, sensor nodes with more residual energy independently become cluster-heads. Sensor nodes meet through local radio communications and find other clusters. Each sensor node with less residual energy chooses a cluster based on the residual energy of the cluster-head, distance to the cluster-head, and an estimation of cluster size. Energy-efficient clusters are eventually formed that extend the life of the sensor network.

The paper consists of the following sections. Section 2 explains the hypotheses of sensor networks considered in this paper. Section 3 introduces ANTCLUST, a clustering algorithm on which our method is based. In Section 4, we propose a new clustering method for energy-efficient data gathering in sensor networks. Results of simulation experiments are given in Section 5. Finally, Section 6 concludes the paper and describes future research.

2 Sensor Network

We consider an application where sensor data are gathered from all sensor nodes to a base station at regular intervals and/or on demand. To avoid installation cost and the need for careful planning, sensor nodes are deployed in the region to monitor in an uncontrolled and unorganized way. Sensor nodes operate on energy-limited, irreplaceable batteries. The capacity of batteries can differ among sensor nodes. Sensor nodes stop due to starvation of battery power, move from one place to another, and are deployed later. Sensor nodes have a wireless transmitter and receiver. The range of radio signals can be adjusted.

Sensor nodes can aggregate or fuse multiple data into single-sized data [9]. Sensor nodes can determine distance to other sensor nodes and the base station in accordance with their absolute or relative positions. Control phases are synchronized among sensor nodes. Although other clustering methods or cluster-based data gathering methods make similar or even stronger assumptions [3], we should note here that the number of applications where these assumptions hold would be limited. In our next work, we consider to adopt our method to sensor networks where some of these assumptions are not valid.

3 ANTCLUST

Ants synthesize a chemical substance called colony odor which differs by individuals, species, and environment; they spread it on their cuticles [10, 11]. When two ants meet, they recognize whether they belong to the same nest by exchanging and comparing these chemical substances, which is updated at each meeting. After spending some time in the nest and repeatedly meeting other ants, a young ant can prepare an appropriate chemical substance to recognize its mates.

ANTCLUST is a clustering algorithm which applies a colonial closure model and regards an object as an ant and a cluster as a nest [8]. A similarity $Sim(i, j) = [0, 1]$ is defined between a pair of objects i and j. Each object i has a cluster identifier, $Label_i$, an acceptance threshold of similarity, $Template_i$, an estimator of cluster size, $M_i = [0, 1]$, and an estimator, $M_i^+ = [0, 1]$, which measures how well the object is accepted in the cluster. They are initialized as $Label_i = 0$, $M = 0$, and $M^+ = 0$. $Template_i$ is defined through a learning phase where object i experiences random meetings.

$$(1) \qquad Template_i \leftarrow \frac{\overline{Sim(i,\cdot)} + Max(Sim(i,\cdot))}{2}.$$

$\overline{Sim(x,\cdot)}$ and $Max(Sim(x,\cdot))$ represent the average and the maximum value of similarity between object x and all object that object x has met, respectively. In ANTCLUST, two randomly chosen objects meet. Based on their similarity, threshold values, and clusters, they create, merge, or delete clusters. By repeatedly conducting random meetings, clusters are appropriately organized so that objects in the same cluster become more similar with one another than those in different clusters.

We consider here the case when two objects i and j meet. First, two objects i and j decide whether they accept their counterpart according to similarity $Sim(i,j)$ and threshold values $Template_i$ and $Template_j$.

$$(2) \qquad \begin{aligned} Acceptance(i,j) \Leftrightarrow\ & (Sim(i,j) > Template_i) \\ & \wedge\ (Sim(i,j) > Template_j). \end{aligned}$$

Then $Template_i$ and $Template_j$ are updated by Eq. (1). Next, their $Label$s are compared. There are five conditions to consider. When neither of them belongs to any cluster and they accept each other (condition 1), a new cluster is created as $Label_i \leftarrow Label_{NEW}$ and $Label_j \leftarrow Label_{NEW}$. If one of two objects, say object i, does not belong to any cluster, and if they accept each other (condition 2), object i joins the cluster of the other as $Label_i \leftarrow Label_j$. When two objects belong to the same cluster and they accept each other (condition 3), they increase their size estimate of their cluster as $M_i \leftarrow (1-\alpha)M_i+\alpha$, $M_j \leftarrow (1-\alpha)M_j+\alpha$, $M_i^+ \leftarrow (1-\alpha)M_i^+ +\alpha$, and $M_j^+ \leftarrow (1-\alpha)M_j^+ +\alpha$. Here, α is a constant between 0 and 1. When two objects belong to the same cluster and they reject each other (condition 4), they first update the size of their estimates as $M_i \leftarrow (1-\alpha)M_i + \alpha$, $M_j \leftarrow (1-\alpha)M_j + \alpha$, $M_i^+ \leftarrow (1-\alpha)M_i^+$, and $M_j^+ \leftarrow (1-\alpha)M_j^+$. Then, an object x with a smaller estimate loses its cluster as $Label_x \leftarrow 0$, $M_x \leftarrow 0$, and $M_x^+ \leftarrow 0$, where $x = (x|M_x^+ = \min_{k\in[i,j]} M_k^+)$. Finally, when two objects belong to different clusters and they accept each other (condition 5), they first estimate how the size of their clusters will be changed as $M_i \leftarrow (1-\alpha)M_i$, and $M_j \leftarrow (1-\alpha)M_j$, Then, an object x in a smaller cluster changes its cluster as $Label_x \leftarrow Label_{(k|k\in(i,j),k\neq x)}$, where $x = (x|M_x = \min_{k\in[i,j]} M_k)$. When none of the above conditions holds, nothing happens.

4 ANTCLUST-based distributed clustering method

4.1 Proposal Outline

In our method, we regard a sensor node as an ant and a cluster as a nest. The similarity of one sensor node to another corresponds to the distance from the sensor node to the cluster-head of another. Sensor nodes meet through wireless communications. Since a sensor node usually has an omni-antenna, a radio signal is broadcast nearby, and it is received by all sensor nodes within its transmission range. In addition, it is a one-way communication, while in ANTCLUST both encountered objects adjust their clusters.

A cycle of data gathering, called a round, consists of four phases [3]: (i) cluster-head candidacy, (ii) cluster formation, (iii) registration, and (iv) data transmission. In the cluster-head candidacy phase, all sensor nodes initially consider themselves as candidates for cluster-head. A sensor node with more residual energy has a chance to advertise its candidacy earlier than others. It becomes a cluster-head by broadcasting an advertisement within a limited range R. Those sensor nodes that receive advertisements from other sensor nodes abandon their candidacy and join a cluster. Details of the cluster-head candidacy phase will be given in 4.2. In the cluster formation phase, sensor nodes meet through radio communications. A percentage P_{ex} of sensor nodes that are not cluster-heads broadcast information about their clusters within a limited range r ($r < R$). Each of the neighboring sensor nodes which receive broadcast messages determines which cluster to join based on information about its own cluster and the newly advertised clusters. Further details will be given in 4.3. Next in the registration phase, each sensor node registers itself as a cluster member by sending a registration message to a cluster-head. In the data transmission phase, cluster members send their data to the cluster-head. The cluster-head receives its members' data, aggregates them into one, and sends it to a base station. The beginning of each round and the timing of phase-changes are synchronized among sensor nodes.

For constructing clusters, a sensor node maintains information about itself and its cluster-head listed in Table 1. Among the parameters in Table 1, the first three are static. $Template_i$ and P_i are initialized when sensor node i is deployed, and they are updated every round. Information about a cluster is initialized at the beginning of each round. The last three parameters are initialized at deployment, but they can be adjusted according to conditions surrounding sensor node i.

4.2 Cluster-head candidacy phase

At the beginning of a round, all sensor nodes consider themselves candidates for cluster-head. Parameters are initialized as:

$$head_i \leftarrow i,\ E_i \leftarrow e_i,\ C_i \leftarrow c_i,\ M_i \leftarrow 1. \tag{3}$$

Assuming that the cluster-head candidacy phase has a T time unit duration, sensor node i announces its candidacy within the radius of R at $T \times (1 - P_i) + K$, where K is a random value $[0,\ (Tp - 1)]$ to reduce the possibility of collisions among sensor nodes with identical P_i and weaken the assumption of the synchronization among sensor nodes. To prolong the lifetime of a sensor network, energy consumption among sensor nodes must be balanced. By adjusting P_i in accordance with the residual energy of neighboring sensor nodes as explained in the next subsection, sensor nodes with more residual energy are more likely to become cluster-heads. p is the constant value used when increasing or decreasing P_i in eq. (5). An advertisement contains an identifier $head_i$, residual energy E_i, coordinates C_i, an estimator M_i of the number of cluster members, and its own residual energy e_i. When a candidacy is announced, E_i is obviously identical to e_i.

When a sensor node that has not yet announced its candidacy receives an advertisement message from another sensor node, it abandons its candidacy and becomes a member of the cluster. Furthermore, when a sensor node that already belongs to a cluster receives another advertisement message, it considers the offer and conducts the same procedure as in the next cluster formation phase to determine which cluster it should join.

4.3 Cluster formation phase

At the end of the cluster-head candidacy phase, every sensor node belongs to a cluster as either a cluster-head or as a member. A percentage P_{ex} of sensor nodes decide to be social and broadcasts information about their clusters within a radius of r. On receiving an advertisement, sensor nodes within radio signal range meet the sensor node and find the cluster. The format for a meeting advertisement is the same as for candidacy. If a sensor node is a cluster-head, it does not cause a meeting. Hereafter we describe a case where sensor node i received an advertisement from sensor node j.

If sensor node i is not a cluster-head, then it adjusts its cluster.

First, sensor node i decides whether to accept cluster-head $head_j$ to which sensor node j belongs by comparing the distance to $head_j$ with threshold $Template_i$.

$$(4) \qquad Acceptance(i,j) \Leftrightarrow \big(d(i, head_j) \leq Template_i\big).$$

Here, $d(i, head_j)$ represents the distance between cluster-head $head_j$ and sensor node i derived from their coordinates c_i and C_j. When sensor node i accepts cluster-head $head_j$, that is, sensor node i considers that cluster-head $head_j$ is close enough, sensor node i compares the two clusters. If sensor nodes i and j belong to the same cluster, i.e., $(head_i = head_j) \wedge (Acceptance(i,j) = True)$, sensor node i increases its estimate of size as $M_i \leftarrow M_i + 1$. If they belong to different clusters it implies that there is another cluster close to sensor node i, as illustrated in Fig. 1. Cluster-head $head_j$ is in sensor node i's $Template_i$, but its advertisement has not been heard by sensor node i. For energy-efficient data gathering, it is effective for a sensor node i to choose a cluster that is closer to sensor node i since sensor node i can save energy by sending sensor data to a closer cluster-head. In addition, sensor node i should choose a cluster-head with more residual energy to avoid driving an energy-poor sensor node to starvation. Finally, a cluster with fewer members is preferred, since energy expended in gathering sensor data to a cluster-head is proportional to the number of cluster members. Thus, sensor node i changes its cluster as $head_i \leftarrow head_j$, $E_i \leftarrow E_j$, $C_i \leftarrow C_j$, and $M_i \leftarrow M_j + 1$, if $(head_i \neq head_j) \wedge \big(Acceptance(i,j) =$

$True) \wedge \left(\frac{E_j}{M_j \cdot d^2(i, head_j)} \geq \frac{E_i}{M_i \cdot d^2(i, head_i)}\right)$ holds. Except in the above conditions, sensor node i does nothing for cluster formation.

Regardless of whether sensor node i is a cluster-head, it updates probability P_i of its cluster-head candidacy to reflect the relationship among its own residual energy e_i and that of sensor node j, e_j:

$$(5) \qquad P_i \leftarrow \begin{cases} \min(1,\ P_i + p), & \text{if } e_i > e_j \\ \max(0,\ P_i - p), & \text{if } e_i < e_j \\ P_i, & \text{if } e_i = e_j. \end{cases}$$

Here, p is a constant value which satisfies $p = [0, 1]$. Thus, the probability of a candidacy is determined in relation to the residual energy of surrounding sensor nodes, not by its absolute amount. Next, sensor node i updates threshold $Template_i$:

$$(6) \qquad Template_i \leftarrow \frac{\overline{d(i, \cdot)} + \max(d(i, \cdot))}{2}$$

where, $\overline{d(i, \cdot)}$ and $Max(d(i, \cdot))$ give the mean and maximum distance between sensor node i and all cluster-heads that sensor node i recognizes through receiving advertisements.

5 Simulation Experiments

We evaluated the effectiveness of our method through simulation experiments. We considered sensor networks of 100 sensor nodes randomly arranged at lattice points in a 50×50 region. A base station was located at (25, 150). In simulation experiments, all communications consumed energy. We used the same energy consumption model as [2]. A sensor node consumes E_{elec} (nJ/bit) in transmitter or receiver circuitry and ε_{amp} (pJ/bit/m^2) in transmitter amplifier. A sensor node expends energy $E_{T_x}(k, d) = k \cdot (E_{elec} + \varepsilon_{amp} \cdot d^2)$ or $E_{R_x}(k) = k \cdot E_{elec}$ in transmitting or receiving a k-bit message to or from distance d. A sensor node also consumes E_{fuse} (nJ/bit/message) in aggregating multiple sensor data into one. We set E_{elec} at 50 (nJ/bit), ε_{amp} at 100 (pJ/bit/m^2), and E_{fuse} at 5 (nJ/bit/message) [2]. A message to advertise a cluster information was set at 60 bits long. Each cluster member sent a 16 bits-long message to a cluster-head for registration. The size of sensor data was 2000 bits. In the following figures and tables, average values over 100 simulation experiments are depicted.

Figures 2 through 5 show the number of sensor nodes that remained alive and the number of rounds for different settings of radius R for reporting candidacy from 10 to 70 and the number of sensor nodes from 50 to 400 for the same size region. The broadcasting radius r for meetings is set at 20, and the percentage P_{ex} of social sensor nodes is set at 10%. All sensor nodes initially have energy of 0.5 J. These figures indicate that there is a trade-off between radius R and the lifetime of a sensor network. As R increases, the number of clusters decreases. Consequently, the number of sensor nodes that become energy-consuming cluster-heads decreases. On the other hand, the number of cluster members increases, it requires more energy to collect sensor data in a cluster. In addition, the diameter of a cluster expands, so cluster members need more energy to send their sensor data to the cluster-heads. When R is small, many clusters are organized into a sensor network. The energy consumption of cluster members is reduced, whereas the number of cluster-heads increases, and much energy is lost.

Several factors affect the desired radius R, but we expect that each sensor node can determine the appropriate radius R by observing its environment and estimating the density. Table 2 summarizes the desired R against the density of sensor networks. In parentheses, the resultant number of clusters are shown. To determine the desired R, we focus on the maximum number of rounds which results in more than 80% of sensor nodes remaining alive. It can be seen that a larger radius is preferable to a sparse network, and vice versa. Although a scheme for nodes to dynamically and autonomously determine an appropriate set of control parameters remains as a future work, we would show a simple idea. Since a cluster-head which is closer than $Template_i$ is considered by sensor node i, it is efficient and effective to consider $Template$ in deciding the radius R. When a sensor node becomes a cluster-head, it receives registration messages from its cluster members. By dividing the number of members by πR^2, it can easily estimate the density ρ around itself. One possible way is to adjust the radius R_i for candidacy of sensor node i is to apply $Template_i + f(\rho)$, where $f(\rho)$ is a monotonically decreasing function of density ρ.

Figures 6 and 7 show the results when the percentage P_{ex} of social sensor nodes changes from 0% to 20% for $R = 20$ and 40, respectively. As shown in Fig. 6, when the number of meetings increases with a larger percentage P_{ex}, more energy-efficient clusters are organized so that more than 80% of sensor nodes remain alive longer, but increased energy consumption is sacrificed in meetings. On the other hand, in Fig. 7, where a cluster-head advertises its candidacy to larger extent, exchanging cluster information results in a shorter network lifetime of a sensor network. When the radius R for broadcasting candidacy grows, the possibility increases that a sensor node close to a cluster-head decides to advertise for cluster-information. Since the radius r is smaller than R, the broadcast signal does not effectively reach sensor nodes in the other clusters, and so no cluster changes occur.

In Figs. 8 and 9, we show the comparison results to other clustering methods, i.e., LEACH [2], a distributed-version of LEACH-C [3], which we called e-LEACH, and HEED [6]. In LEACH, a percentage P_{opt} of sensor nodes advertise their candidacy to the whole of a sensor network. Hearing advertisements, each sensor node chooses the closest cluster-head and registers itself as a cluster member. In e-LEACH, each sensor nodes advertise their candidacy based on its residual energy. The probability H_{prob} of candidacy is calculated by multiplying the pre-determined optimal number of clusters and its residual energy and further divided by the sum of residual energy of cluster members. HEED also takes into account residual energy of sensor nodes in electing cluster-heads. The probability H_{prob} of candidacy is given by multiplying the pre-determined constant probability and the percentage of residual energy against the maximum capacity.

Figure 9 illustrates the results in terms of the total amount of sensor data received at the base station. We considered sensor networks of 100 sensor nodes randomly arranged at lattice points in a 100×100 region. A base station was located at (50, 175). We employed another model of energy consumption in message transmission as [3], where $E_{T_x}(k, d) = k \cdot (E_{elec} + \varepsilon_{mp} \cdot d^4)$ for $d \geq d_0$. The threshold d_0 was introduced to take into account the effect of multi-path fading. All sensor nodes had the same initial residual energy of 0.5 J. We set ε_{amp} at 10 (pJ/bit/m^2) and ε_{mp} at 0.0013 (pJ/bit/m^4). The threshold d_0 was set at 75 (m). An advertisement message was 400 bits long and an registration message was 200 bits long. The size of sensor data was 1000 bits. Most of conditions of simulation experiments were the same as in [6]. In our proposal, R, r, and P_{ex} were set at 40, 20, and 10%, respectively. For the other methods, we chose such parameters that led to the best performance.

It is clearly shown that our proposal outperforms the others. In LEACH, every sensor has the same chance to become a cluster-head. Hence, a sensor node with insufficient residual energy occasionally becomes a cluster-head and halts due to battery depletion, even if there is a sensor node with rich battery power nearby. Moreover, in LEACH, since cluster-heads are chosen in a probabilistic way, the predetermined optimal number of clusters are not necessarily organized and clusters are not well distributed in a region. The reason why e-LEACH leads to shorter lifetime of a sensor network than LEACH is that the size of messages becomes longer for sensor nodes to advertise the amount of residual energy. Since HEED needs more broadcasting to form clusters than our proposal, it consumes more energy than ours.

6 Conclusions and Future Work

In this paper, based on ANTCLUST, we proposed a novel clustering algorithm for energy-efficient data gathering in sensor networks. Sensor nodes with more residual energy became cluster-heads, improving the organization of clusters by local interactions among sensor nodes. To summarize characteristics of our proposal, first, our method organizes clusters in a completely distributed way. Each sensor node determines whether to become a cluster-head or not, whether to be social or not, and which cluster to join by itself. Second, our method leads to a longer lifetime of a sensor network than other methods by equalizing the residual energy of sensor nodes. In our method, a sensor node with more residual energy has a larger chance to become a cluster-head. In addition, a sensor node chooses its cluster not only by the closeness but the residual energy of a cluster-head. Third, our method can gather more data under unstable radio environments. In our method, if a sensor node does not hear any candidacy messages, it becomes a cluster-head and send its own sensor data to a base station by itself. Simulation experiments verified that our proposed method prolonged the lifetime of sensor networks by more than 30% of the others.

We are now considering a further efficient clustering algorithm where sensor nodes autonomously adjust control parameters through observation of its surroundings. We also consider the coverage area of sensor networks for energy-efficient cluster-based data gathering. Furthermore, we extend our method to the case where sensor data are sent to a base station through communications among cluster-heads, i.e., multi-hop transmissions.

References and Notes

1 I. F. Akyildiz, W.Su, Y. Sankarasubramaniam, and E.Cyirci, "Wireless sensor networks: a survey," *Computer Networks*, pp. 393–422, March 2002.

2 W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of HICSS-33*, pp. 3005–3014, January 2000.

3 W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, pp. 660–670, October 2002.

4 S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," in *IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, pp. 924–935, April 2001.

5 K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC, 2003)*, March 2003.

6 O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proceedings of IEEE INFOCOM*, March 2004.

7 E. Bonabeau, G. Theraulaz, and M. Dorigo, *Swarm Intelligence*. New York : Oxford University Press, October 1999.

8 N. Labroche, N. Monmarché, and G. Venturini, "A new clustering algorithm based on the chemical recognition system of ants," in *Proceedings of ECAI 2002*, pp. 345–349, July 2002.

9 D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. CRC Pr I Llc, June 2001.

10 R. Yamaoka, "The communication and community of ants," *NATURE INTERFACE*, no. 6, pp. 58–61, 2001.

11 R. Yamaoka, "How do ants recognize their nestmates?," *Quarterly Journal Biohistory*, no. 23, 1999.

Table 1 Notations

Information about sensor node i	
i	node identifier
e_i	residual energy
c_i	coordinates
$Template_i$	threshold of similarity, initial value, initial value R
P_i	probability of candidacy [0, 1], initial value 0.5
Information about a cluster of sensor node i	
$head_i$	identifier of a cluster-head
E_i	residual energy of a cluster-head
C_i	coordinates of a cluster-head
M_i	estimator of the number of cluster members
System parameter	
R	radius for broadcasting candidacy for cluster-head
r	radius for broadcasting cluster information for meetings
P_{ex}	proportion of social sensor nodes that cause meetings [0, 1]

Table 2 Density of sensor nodes and desired radius

density	0.02	0.04	0.08	0.16
desired R (clusters)	50 (1～2)	50 (2～3)	40 (3～4)	20 (5～8)

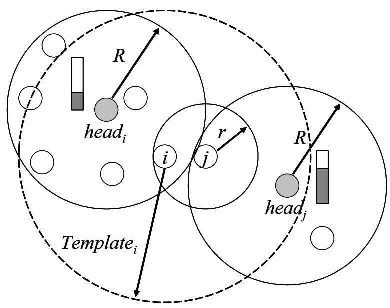

Figure 1 An example where $(head_i \neq head_j) \wedge (Acceptance(i, j) = True)$

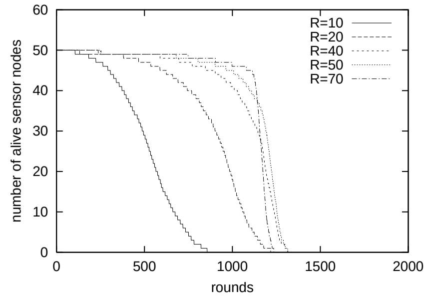

Figure 2 Number of alive sensor nodes (50 nodes, $R = 10-70,\ r = 20,\ P_{ex} = 10\%$)

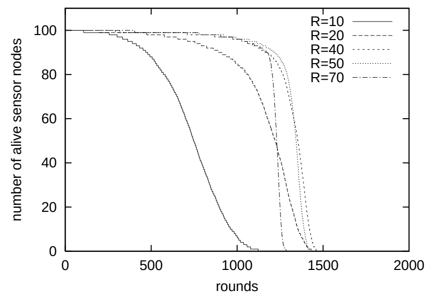

Figure 3 Number of alive sensor nodes (100 nodes, $R = 10-70,\ r = 20,\ P_{ex} = 10\%$)

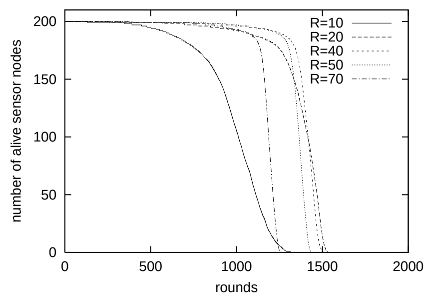

Figure 4 Number of alive sensor nodes (200 nodes, $R = 10-70,\ r = 20,\ P_{ex} = 10\%$)

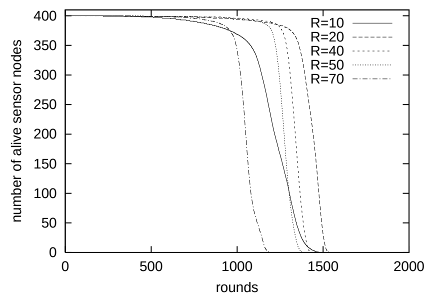

Figure 5 Number of alive sensor nodes (400 nodes, $R = 10-70,\ r = 20,\ P_{ex} = 10\%$)

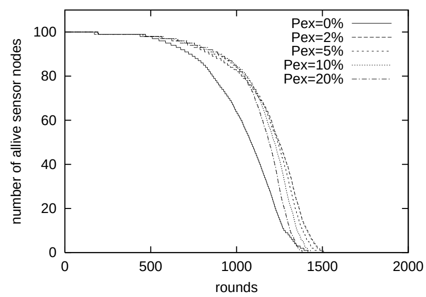

Figure 6 Number of alive sensor nodes (100 nodes, $R = 20,\ r = 20,\ P_{ex} = 0\% - 20\%$)

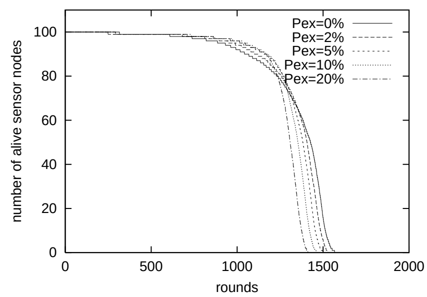

Figure 7 Number of alive sensor nodes (100 nodes, $R = 40,\ r = 20,\ P_{ex} = 0\% - 20\%$)

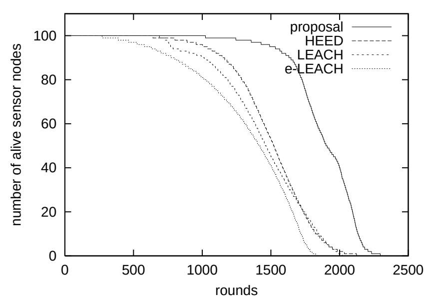

Figure 8 Comparison of the number of alive sensor nodes with the other clustering methods

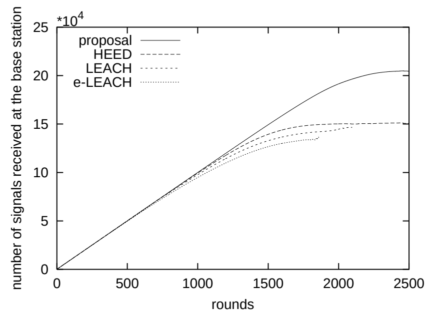

Figure 9 Comparison of the amount of data received at BS with the other clustering methods