Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor Networks

Ying Liao, Huan Qi, and Weiqun Li

Abstract— **Wireless sensor networks (WSNs) are composed of a large number of inexpensive power-constrained wireless sensor nodes, which detect and monitor physical parameters around them through self-organization. Utilizing clustering algorithms to form a hierarchical network topology is a common method of implementing network management and data aggregation in WSNs. Assuming that the residual energy of nodes follows the random distribution, we propose a load-balanced clustering algorithm for WSNs on the basis of their distance and density distribution, making it essentially different from the previous clustering algorithms. Simulated tests indicate that the new algorithm can build more balanceable clustering structure and enhance the network life cycle.**

Index Terms— **Cluster, connectivity density, life cycle, random distribution, wireless sensor networks.**

I. Introduction

THE DEVELOPMENT of tiny, low-cost and intelligent sensors with communication capabilities has prompted the emergence of wireless sensor networks (WSNs) [1]. Since, WSNs have attracted considerable attention because of their extensive applications in many areas such as object tracking, intrusion detection, environmental monitoring, traffic control, inventory management in factory and health related applications [2], [3] and so on.

In WSNs, the energy of network node is often limited, so the efficient use of energy is a must in topology control. At the same time, since there are usually a large number of nodes in WSNs, the node can only get part of the network topology information. As a result, the clustering algorithms are needed to select the appropriate sub-cluster in local network on the basis of the partial information.

The topology of WSNs often changes dynamically for a variety of reasons, including sensor node failure as battery runs out, destruction of environmental factors, location change of sensor nodes [4], wireless communication link signal alteration due to node power control or environmental factors, the adding of new nodes into the network to enhance the monitoring accuracy, etc. Thus, WSNs should be able to get adapted to these variations and reconfigure to satisfy user's task dynamically. Meanwhile, a big concern in WSNs is the acquisition of regional data information rather than specific node information, and thus the topology control mechanism of traditional networks is frustrated.

Due to energy constraints, a sensor node can but communicate directly with other sensors within a limited distance. In order to enable communication between sensors out of each other's communication range, sensors form a multi-hop communication network. Utilizing clustering algorithm to form a hierarchical network topology is the accustomed realizing scenario of network management and data aggregation for WSNs, and clustering facilitates the distributed control of the network. Moreover, clustering nodes into groups not only saves energy but also reduces network contention when nodes communicate their data over shorter distances to their respective cluster-heads.

Motivated by the former research of clustering algorithm, the objective of this paper is to study a Balanced Clustering Algorithm with Distributed Self-Organization for Wireless Sensor Networks (DSBCA), which can deal with stochastic distribution of sensor nodes. In the scenario, we could calculate the radius of cluster based on distance and distribution, and take connectivity density and residual energy of nodes into account to form clusters.

The remainder of this paper is organized as follows. Section 2 reviews several cluster based algorithms proposed previously. In Section 3, we put forward the system model of our scheme. In Section 4, we investigate our new balanced clustering algorithm. Then, the performance analysis of the proposed algorithms is presented in Section 5. Finally, conclusions are given in the last section of the paper.

II. Related Work

LEACH [5] proposed initially is a distributed and single-hop clustering algorithm. Each clustering cycle consists of cluster forming phase and data communicating phase. Simultaneously, LEACH can guarantee not only the equal probability of each node as cluster head, but also the relatively balanced energy consumption of the network nodes. Many follow-up algorithms are put forward relying on the work of LEACH.

Manuscript received August 14, 2011; revised October 14, 2012; accepted November 4, 2012. Date of publication November 15, 2012; date of current version March 26, 2013. This work was supported in part by the National Natural Science Foundation of China under Grant 60774036, the Natural Science Foundation of Hubei Province of China under Grant 2008CDA063, and the Research Fund for Central Colleges under Grant C2009Z025Y. The associate editor coordinating the review of this paper and approving it for publication was Dr. V. R. Singh.

Y. Liao is with the Zhengzhou Institute of Information Science and Technology, Zhengzhou 450007, China (e-mail: liaoying_earth@hotmail.com).

H. Qi is with the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: qihuanster@gmail.com).

W. Li is with Huawei Company, Shenzhen 518000, China (e-mail: tieyun371@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2012.2227704

In the cluster forming phase, the cluster head nodes will inform other nodes once they are selected. The ordinary nodes can choose which cluster to join into automatically according to their distance from head nodes. In general, the election of cluster head depends on the selected threshold $T(n)$ as follows.

$$T(n) = \left\{ \begin{array}{ll} \frac{P}{1-P^{*}[r \mod (1/p)]}, & n \in G \\ 0, & \text{else} \end{array} \right\} \tag{1}$$

where P is the percent of cluster head nodes in all nodes, n is the token of the node, and r is the number of rounds for the election. r mod $(1/p)$ is the number of nodes elected as cluster head in a cycle, and G is the set of nodes not elected as a cluster head, while each node is elected as the cluster head with the same probability. In each circle, first the threshold $T(n)$ of the node elected as cluster head is set to 0, and then the probability of the remaining nodes increases. Thus this process can guarantee that the node with the equal probability is elected as the cluster head.

LEACH algorithm only considers the probability for each node to become cluster head. However, the specific geographical location of nodes is left out, so the election of cluster heads is heterogeneous, as proved in a large number of experiments. Additionally, LEACH allows building single-hop cluster, and a large number of clusters might be formed consequently, causing great energy consumption of communications. Further studies on the LEACH algorithm could be found in [6], [7].

By taking overhead of communication within a cluster into account, HEED [8] algorithm overcomes the shortcoming of unevenly distributed cluster heads as enjoyed by the LEACH algorithm. And the residual energy of node as a parameter is introduced to elect cluster head with more energy to undertake data forwarding tasks. In initialization phase, nodes send the messages to compete with the initialized probability of CH_{prob}. When the election of cluster head is completed, other nodes join into clusters by means of the information gathered in competing phase. Here, CH_{prob} is described as

$$CH_{prob} = \max(C_{prob} + E_{resident}/E_{\max}, p_{\min}) \tag{2}$$

where C_{prob} and $p_{\min}$ are the whole network parameters affecting the convergence speed of the algorithm, the recommended values of which are given in simulation experiments; $E_{resident}/E_{\max}$ is the ratio of the node residual energy and initial energy.

HEED algorithm is distinguished from the LEACH in cluster head selection mechanism. Meanwhile, the speed of clustering in the HEED algorithm is also improved and the cost of communication within the cluster has been taken into account. Moreover, the residual energy of node is as an important parameter to be introduced in cluster head selection mechanism, since it makes cluster heads competent for data aggregating and forwarding to form a more rational network topology. Even so, there are also some problems in HEED algorithm. For instance, the competition of cluster head may exclude some nodes from joining into any clusters [9].

GAF [10] is a location-based node clustering algorithm. In this algorithm, the monitored area is divided into virtual cells, and the nodes are assigned to cells as per their geographic locations. Each cell will generate a cluster head node periodically. Cluster head nodes remain active, while other nodes could be in sleep mode to reduce energy consumption. GAF is the routing algorithm proposed in ad hoc network at first, and the method of virtual division of the cell is used for WSNs. The main idea of GAF is to get the nodes into sleep state as long as possible to reduce energy consumption in listening state. Though, the resource of sensor nodes is limited, and the clusters based on geographical location algorithm are exigent for sensor nodes while paying no attention to the node residual energy problems. So GAF may lead to death of nodes due to premature depletion of energy, thus incurring unstable network Topology.

WCA [11] is a classical algorithm based on node degree, the number of single-hop neighbors. The election of cluster head relies upon the factors of node degree, send-receive energy and residual energy. Meanwhile the size of cluster (the communication consumes large amounts of energy when cluster is too large) is limited in order to save energy. In contrast, the WCA clustering algorithm is more comprehensive than the previously proposed algorithms, and some experiments show that the performance is more superior. Additionally, the main drawback of WCA is that it needs to obtain the weight of the node and require each node to save all the information of nodes before initializing network, so excessive amounts of computing and communications may cause excessive consumption in clustering directly. In the process of aggregating and forwarding, the overhead may also give rise to excessive energy consumption and rapid death of cluster head node, incurring instability in the network topology.

K-clustering [12] algorithm can constitute maximum k-hop non-overlapping clusters with partial networks topology information rather than the whole network topology. At the same time, it can also save energy to prolong network survival time. Furthermore, owing to dynamic network topology changes, it is of significance studying clustering based on local information. Nevertheless, it doesn't consider cluster size and may form unbalanced cluster. For example, some clusters contain tremendous number of nodes, which results in too large overhead of inter communication.

Different improved K-clustering algorithms have been come up with successively to fix this problem. A representative algorithm is proposed by Lin and Chu [13] with using hops as one of the constraint parameters. In this algorithm, the node is elected as cluster head randomly, and the distance between cluster members and cluster head doesn't exceed k hops. The algorithm is more effective in restricting data forwarding distance, but it still doesn't solve unbalanced clustering (excessive clustering nodes). In addition, in this algorithm, only the root of subtree knows which cluster it belongs to, while other nodes don't have this info. If the cluster head or the root of subtree node fails, it would be inefficient and unfavorable to cluster again.

ESAC [14] algorithm combines the advantages of the above proposed algorithms, and it improves clustering performance by overcoming their shortcomings. This algorithm uses the method of calculating weight in selecting cluster head. The weight of each node is calculated relying on the combination of two parameters: residual energy and mobility.

The cluster size ranges between two thresholds ($\text{Thresh}_{\text{lower}}$ and $\text{Thresh}_{\text{high}}$), and the distance between each cluster node and its cluster head is no more than 2-hop. This differs from LEACH, and the algorithm builds the balanceable and smooth clustering network by considering the k-density, residual energy and mobility so as to avoid fixed cluster head project, which may results in excessive energy consumption of cluster head. The process of electing cluster head is re-launched in a certain period (service period). It calculates the weight of each node in every stage of cluster head building in order to ensure the most appropriate node to become cluster head and restrict the size of cluster very well. When a cluster head dies or is moved to other cluster, the maintenance process is triggered. The process is similar to that of building cluster launched by a random member of the former cluster, and is limited only to the members losing their cluster head. In this way it can avoid the previous 'chain loop' problem existed in clustering algorithm, and has little effect on the network topology. However, the structure of 2-hop clusters is not suitable for all circumstances. In some cases, we need to constitute clusters more than 2-hop.

In the practical application of WSNs, such as node localization, goal tracing, network robustness promotion, etc., overlapping networks work better. Youssef *et al.* propose KOCA algorithm [15], an overlapping clustering algorithm suitable for uniform distribution of sensor nodes. The authors expatiate the overlapping k-hop clustering problem for wireless sensor networks and present a randomized distributed heuristic algorithm for solving the problem. The characteristics of KOCA are also studied through analysis and simulation. But in KOCA, each node becomes cluster head with the equal probability without considering the residual energy of the node. Therefore, it can't be applied in actual situation commendably.

DEBUC [16] adopts an unequal clustering mechanism in combination with an inter-cluster multihop routing. Through a time based competitive clustering method, DEBUC partitions all nodes into unequal clusters in which the cluster heads can preserve more energy for the inter-cluster relay communications to avoid the 'hot-spots' problem. For inter-cluster traffic, DEBUC adopts an energy-aware multihop routing system to reduce and balance the energy overhead of the cluster heads.

Mittin *et al.* put forward an actually self-stabilizing clustering algorithm with good robustness [17]. The authors propose several enhancements to the scheme with directed acyclic graph construction as the key technique to reduce the stabilization time and thus improve stability in a dynamic environment.

Moreover, since selecting the cluster head is difficult in environments of different characteristics, Anno *et al.* [18] employ different fuzzy descriptors and evaluates their performance. The sensor nodes closer to the base station consume much more energy due to the relaying network traffic near the base station. Hence, the sensor nodes closer to the base station may quickly exhaust battery. Besides the residual energy, during the cluster head election, Bagci *et al.* [19] further discusses a fuzzy descriptor, distance to the base station. The unbalance clustering technique comes from the idea of decreasing the cluster sizes in the clusters close to the base station.

In our proposed scheme, we are intended to generate stable and balanced clusters since a stable clustering scheme considerably alleviates overhead among the sensors and prolongs the life cycle of nodes. For that, we introduce determinants to nodes' weight to form a reduced number of clusters on one hand, and to guarantee the stability of the clusters formed on the other hand. We calculate the clustering radius relying on the connectivity density and the distance from the base station. Meanwhile, in order to reduce clusters structure change, we also involve in weight computation of the nodes such parameters as residual energy, connection density and times of being elected as cluster head nodes.

III. System Model

We first recall some notations and expressions to be used in the main algorithm. A wireless sensor network can be modeled as a graph $G = (V,\ E)$. G is a unit disk graph. V is the set of sensor nodes, and $E \subseteq V^2$ is the set of links among the nodes. There is a link between two nodes if they are within the transmission range of each other. What we discuss are multi-hops WSNs with heterogeneous distribution, and each node is assigned to a unique identifier.

The sensor node is not equipped with GPS equipment and couldn't position solely. This paper makes the following assumption. All nodes constitute network topology by self-organizing, and the nodes transmit data with the same signal intensity and with the same maximum distance L in the same frequency (shared channels are freely competitive and faultless). The refreshing time of algorithm is restrained by the switching of nodes state (such as node death). The position of each sensor node is fixed in a certain period of time, which is commonly adopted in WSNs modeling. All the sensors are distributed according to a Poisson process with intensity λ.

We use the same radio model as stated in [20] where to transmit an l-bit message over a distance d, the power consumption is

$$\begin{aligned} E_{Tx}(l,d) &= E_{Tx-\text{elec}}(l) + E_{Tx-amp}(l,d) \\ &= \begin{cases} lE_{\text{elec}} + l\varepsilon_{\text{fs}}d^2, & d < d_o \\ lE_{\text{elec}} + l\varepsilon_{\text{mp}}d^4, & d > d_o \end{cases} \end{aligned} \tag{3}$$

and to receive the message, the power consumption is

$$E_{Rx}(l) = E_{Rx-\text{elec}}(l) = lE_{\text{elec}}. \tag{4}$$

The electronics energy E_{elec} depends on the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\varepsilon_{\text{fs}}d^2$ or $\varepsilon_{\text{mp}}d^4$, depend on the distance to the receiver and the acceptable bit-error rate.

IV. DSBCV Algorithm

The possible applications of clustering algorithms (such as LEACH, HEED, KOCA, etc.), proposed previously, are in uniformly distributed WSNs without considering the distance from the base station. However, in the practical application of WSNs, the nodes are usually randomly arranged. In this case, if the clustering algorithm doesn't take the distribution of nodes into account, using uniform clustering strategy may lead to unbalanced topological structure, and some nodes die rapidly because of excessive energy decline. The purpose of DSBCA is to generate clusters with more balanced energy and avoid creating excessive clusters with many nodes. The clusters near the base station also forward the data from further clusters (all clusters need to communicate with the base station, but long-distance wireless communication consumes more energy), and as we all know, too many members in a cluster may bring about excessive energy consumption in management and communication. Hence, based on the above concerns, DSBCA algorithm considers the connectivity density and the location of the node, trying to build a more balanced clustering structure.

The basic idea of DSBCA is based on the connectivity density and the distance from the base station to calculate k (clustering radius). The clustering radius is determined by density and distance: if two clusters have the same connectivity density, the cluster much farther from the base station has larger cluster radius; if two clusters have the same distance from the base station, the cluster with the higher density has smaller cluster radius.

Fig. 1 shows DSBCA clustering in uniform distribution. DSBCA forms different clustering layers in which the radiuses of farther clustering layers are larger, and in the same layer the clustering radius is identical.

Fig. 2 shows DSBCA clustering in non-uniform distribution. In the case of non-uniform distribution, the cluster radiuses are determined by the distance from the base station and connectivity density of nodes. With farther distance from the base station and lower connectivity density, the cluster radius is larger; on the contrary, with closer distance from the base station and lower connectivity density, the cluster radius is smaller.

DSBCA can be divided into three stages: cluster head selecting phase, clusters building phase and cycle phase.

A. Cluster Head Selecting Phase

DSBCA follows a distributed approach to establish hierarchical structure in self-organizing mode without central control.

DSBCA selects the random nodes to trigger clustering process first. Then the trigger node U_t calculates its connected density and distance from the base station to determine cluster radius k by (5), and becomes the temporary cluster head.

$$k = floor[\beta D(U_t)/D_k(U_t)] \tag{5}$$

where $D(u)$ is the distance from the base station of u, $D_k(u)$ is the connectivity density of node u, β is the sensor parameters determined by specific applications of WSNs, and *floor* is the calculation of rounding.

$D(u)$ can be calculated as follows.

$$D(u) = 10^{\frac{|RSSI-A|}{10\cdot n}} \tag{6}$$

where *RSSI* is received signal strength indicator, and A is the signal strength with 1 meter distance from the base station [21].

$N_k(u)$ is k-hop neighbors of node u.

$$N_k(u) = \{v \in V | v \neq u \wedge d(u,v) \leq k\} \tag{7}$$

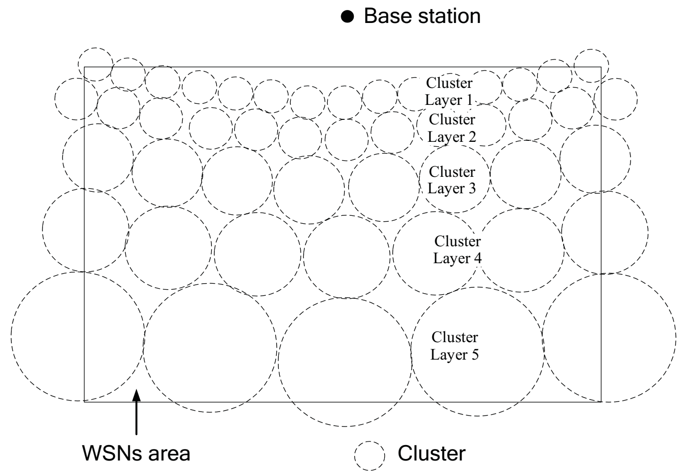

Fig. 1. DSBCA clustering in uniform distribution.

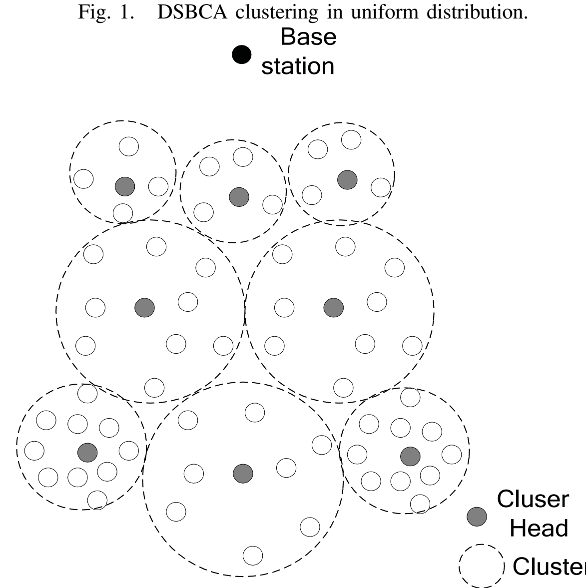

Fig. 2. DSBCA clustering in nonuniform distribution.

where $d(u,v)$ is the hops between node u and node v. We use hops to indicate distance approximately.

Node connection density is calculated by (8), and $|N_k(u)|$ is the number of k-hop neighbors of node u.

$$D_k(u) = \frac{|(t,v) \in E/t, v \in N_k(u) \cup \{u\}|}{|N_k(u)|}. \tag{8}$$

DSBCA follows a distributed approach to build hierarchical structure in self-organizing mode without central control. In this phase, the node with the highest weight in k-hop neighbors of U_t is elected as cluster head. The weight of the node is calculated by (8), which takes the residual energy, connection density, and times of being elected as cluster head of nodes into account. Thus, we can generate clusters more balanced in energy and position.

$$W(u) = \phi \times P[D_k(u)] + \varphi \times P\left[\frac{R_e(u)}{E(u)}\right] - \gamma \times P[H(u)],$$
$$0 \leq \phi, \varphi, \gamma \leq 1, \quad \gamma < \phi + \varphi < 1 \tag{9}$$

where ϕ, φ, γ as the effect factors are defined by specific application, $R_s(u)$ is the residual energy of node u, $E(u)$ is the initial energy of node$u$, $H(u)$ is the times of the node u being elected as cluster head. In this way we decrease the prospects of u being elected as cluster head to balance the overall energy consumption.

In the initial stage, the node U_t triggers the clustering process and sends Hello messages to its k-hop neighbors. The neighbors in k-hop utilize (9) to calculate the respective weight, and then the node with the highest weight will become the cluster head. From then on, cluster head node broadcasts (Head_message) in its k-hop neighbors to declare itself as cluster head and asks them to join the cluster. Head_message includes the ID of cluster head node (HID), the ID of the sending node (SID) and the number of hops from the cluster head (HD). When a node receives Head_message, SID can be used to maintain a path to reach the cluster head. The algorithm discards broadcast package when HD is over k to ensure that the cluster is no more than k-hop. When a neighbor node receives Head_message, even if it is already in a cluster, it sends Join_message to the cluster head to request joining the new cluster as long as its weight is lower. Head_message is limited to transmission within k-hop, so it may happen that some nodes couldn't receive any Head_message. In DSBCA algorithm, if the node does not receive Head_message in $T(w)(T(w) < T(k))$, it declares itself the cluster head, where $T(w)$ is waiting time, and $T(k)$ is the refresh time related to distribution of nodes and specific applications. The settings of $T(w)$ and $T(k)$ should ensure that each node in the network can find its own cluster head, and the algorithm restarts the clustering process after $T(k)$ circularly.

B. Clusters Building Phase

DSBCA sets the threshold of cluster size. The number of cluster nodes can not exceed the threshold to avoid forming large clusters, which will cause extra overhead and thus reduce network lifetime. When the cluster head node receives Join_message sent by the ordinary node, it will compare the size of cluster with threshold to accept new member and update the count of cluster nodes if the size is smaller than threshold, or reject the request. If the rejected node has cluster head already, the clustering process ceases. Otherwise, it finds another appropriate cluster to join.

Each member node of cluster maintains a cluster information table, which saves the HID, HD, SID and other information. If a node receives transmitting packet in work, it will update its cluster information table correspondingly. For example, the node checks HD in a newly received packet, if HD is smaller, then it updates the value of HD in table, with SID updated. That is to say, it has found a shorter path to cluster head and sets the new SID as its forwarding node. There is only a single HID entry in the ordinary node because it belongs to one cluster head, but the overlapping cluster node has multiple HID information entries for different clusters.

DSBCA algorithm avoids the fixed cluster head scheme (cluster head manages cluster and forwards data, so it consumes energy faster than other nodes), with periodic replacement to balance the node energy consumption.

C. Cycle Phase

The cluster is stable for a while until the process of reelecting cluster head is triggered in $T(k)$. The cluster head gathers the weight of all member nodes, and then selects the node with highest weight as the next head node. In this way, the communication costs are decreased. The reelecting of cluster head occurs in the 'old' cluster, so the broadcast of temporary head and the corresponding responses of all the k-hops neighbors are unnecessary.

According to [15], the average overall communication overhead in per cluster can be calculated approximately as follows.

$$N_p = \frac{\ell k(4k-1)(k+1)}{6} \sim O(k^3) \tag{10}$$

where ℓ is average degree of node.

So, the reduced communication can be expressed approximately by

$$N_b + N_r \sim 2N_b \sim 2O(k^3) \tag{11}$$

where N_b is overhead of the broadcast, and N_r is overhead of the corresponding responses.

V. Simulation

In this section, lots of simulation experiments are presented to demonstrate the effectiveness and superiority of the proposed new algorithm in comparison with the previous algorithms. Intuitively, more obvious advantages of DSBCV algorithm could be seen from fig. 3 to fig. 14. More details are described as follows.

In the simulation experiments, WSNs nodes are randomly distributed in the 50 m × 50 m area. We compare DSBCA algorithm with the classical LEACH, HEED and WCA, observing the network lifetime especially. In comparison, we make use of the model in [12], and MATLAB 2009b as simulation tool.

In the course of simulation, the target area is set as [0, 50] × [0, 50], and the base station is in the interval [25, 100]. At the same time, the value of k is fixed as 2, 3 and 4 (when $k = 1$, the limited communication distance of nodes may lead to low network coverage because of insufficient neighbor nodes; when $k > 4$, excessive big clusters or excessive nodes without cluster may reduce network life cycle), and the threshold is set as 15.

To facilitate unified comparison, the survival time is represented by the number of rounds where each round begins with a set-up phase when the clusters are organized, followed by a steady-state phase when the data transfers to the base station. When contrasting algorithms in the same round, the algorithm with lower ratio of dead nodes is deemed better. Similarly, we stipulate that the life cycle ends when 80% nodes are dead in our experiments.

We compare DSBCA algorithm with the classical LEACH, HEED and WCA, observing the network lifetime especially. DSBCA generates harmonious clusters to decrease the energy cost of communication in the cluster, so it prolongs the network lifetime. Fig. 3 and fig. 4 show respectively the average rounds of communication in 20 experiments for various algorithms, when 10%, 20%, 30%, 40%, 50%, 60%, 70%

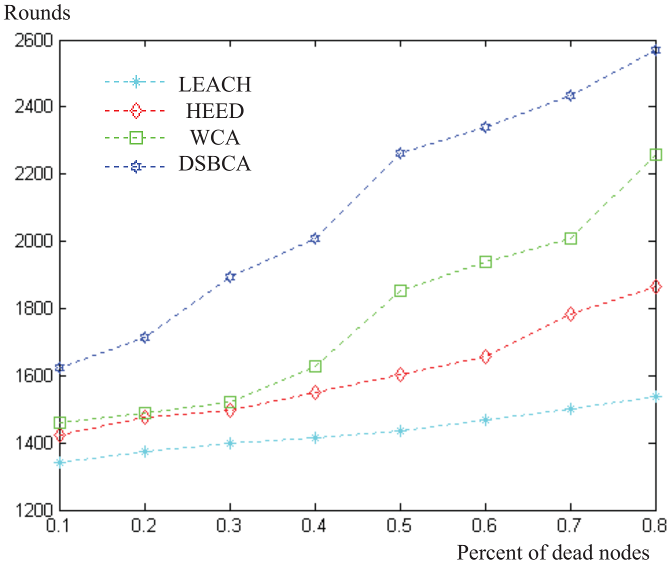

Fig. 3. Comparison of rounds with 144 nodes distributed evenly.

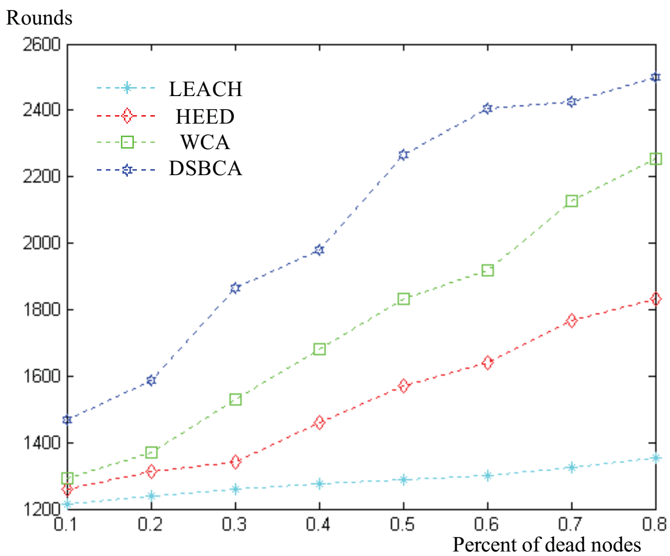

Fig. 4. Comparison of rounds with 196 nodes distributed evenly.

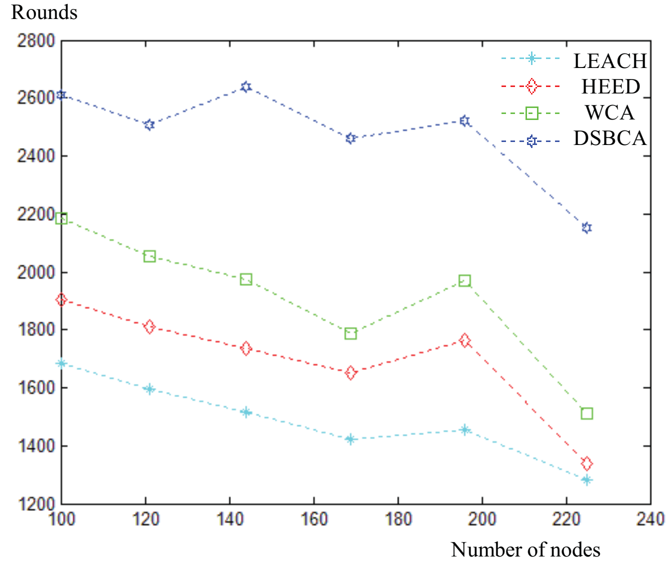

Fig. 5. Comparison of life cycle changing with the number of nodes in uniform distribution.

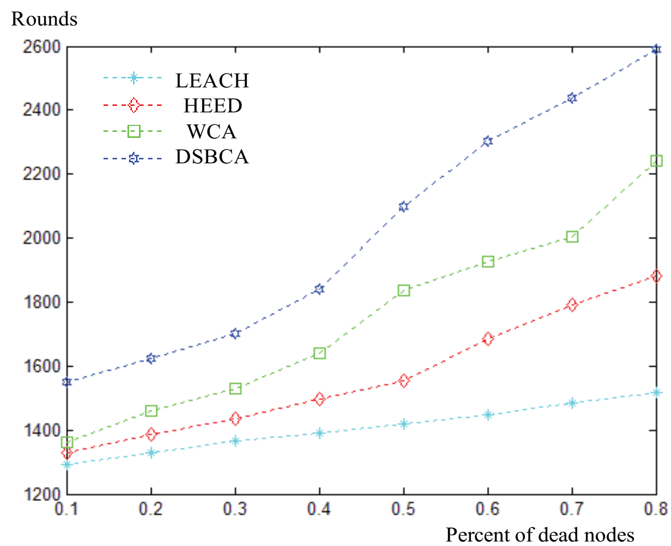

Fig. 6. Comparison of rounds with 150 nodes distributed randomly.

and 80% of nodes are dead. The simulation conditions are 144 nodes (Fig. 3) distributed evenly and 196 nodes (Fig. 4) distributed evenly. We can see that, for the same round, DSBCA has lower ratio of dead nodes compared to the other algorithms. We also carry out the simulation further to find how life cycle of various algorithms changes with the number of nodes in uniform distribution. The result is shown in fig. 5.

By Fig. 5, we deduce that life cycle of various algorithms will change with the number of nodes. Simulation conditions are 100, 121, 144, 169 and 196 nodes in uniform distribution, respectively. The results are based on 20 tests. In simulation, the number of nodes is set as $n^2(n = 10, 11, 12, 13, 14)$ to satisfy uniform distribution of nodes in rectangle.

It can be seen from Fig. 5, DSBCA algorithm achieves better performance on various conditions. DSBCA realizes 2591 and 2642 rounds for 100 and 144 nodes respectively. In the worst case, when the number of nodes is 225, DSBCA realizes 2165 rounds, still better than the other three algorithms. In the case of uniform distribution, the distance from the base station introduced can enhance the life cycle of clustering. In Fig. 5, with the number of nodes increasing, DSBCA algorithm achieves better performance than other algorithms, so it adapts to a variety of node density.

Since the nodes of WSNs are placed randomly in numerous applications, we put emphasis on random distribution in the simulation. Fig. 6 shows the average rounds of communication in 20 experiments for various algorithms, when 10%, 20%, 30%, 40%, 50%, 60%, 70% and 80% of nodes are dead. The simulation conditions are 150 nodes of random distribution. From fig. 6, we find that DSBCA obtains better performance than other algorithms at the same rates of death, and the rounds change with death rate smoothly.

In another set of experiments shown in fig. 7, we come to the same conclusion that DSBCA performs better than other

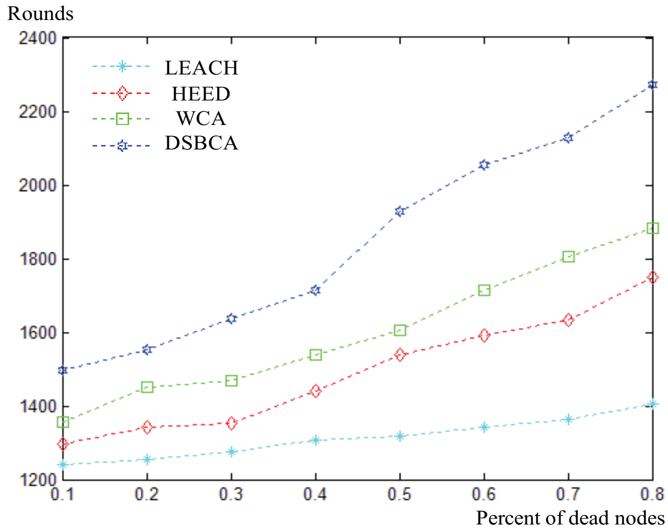

Fig. 7. Comparison of rounds with 250 nodes distributed randomly.

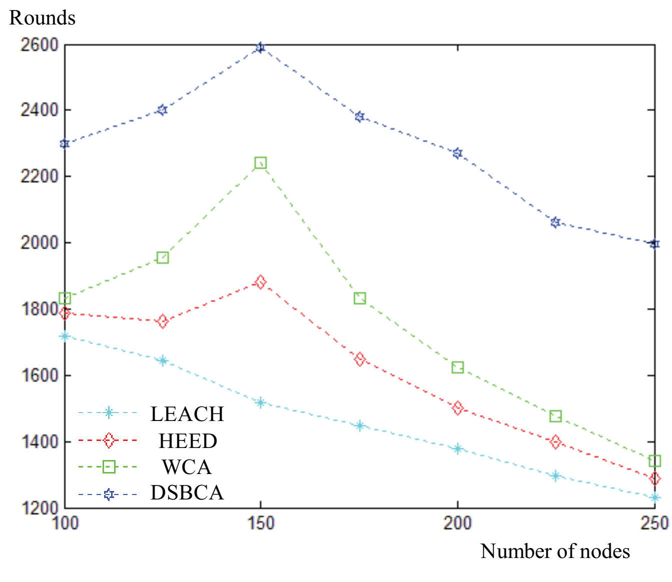

Fig. 8. Comparison of life cycle changing with the number of nodes in random distribution.

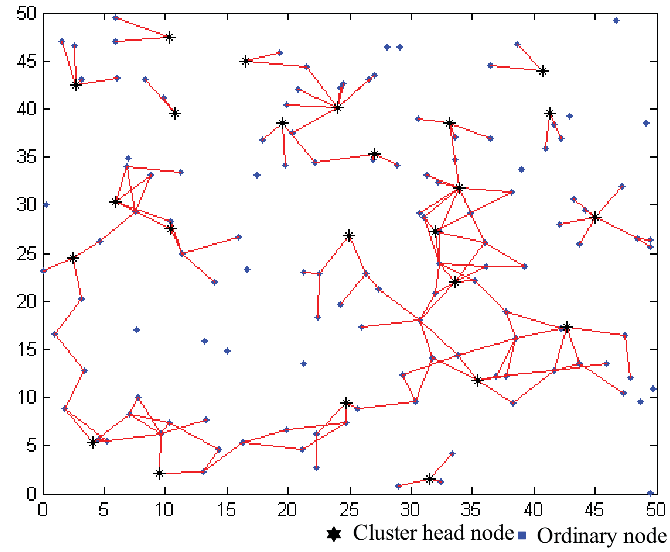

Fig. 9. Simulation result of clustering structure with 150 nodes distributed randomly.

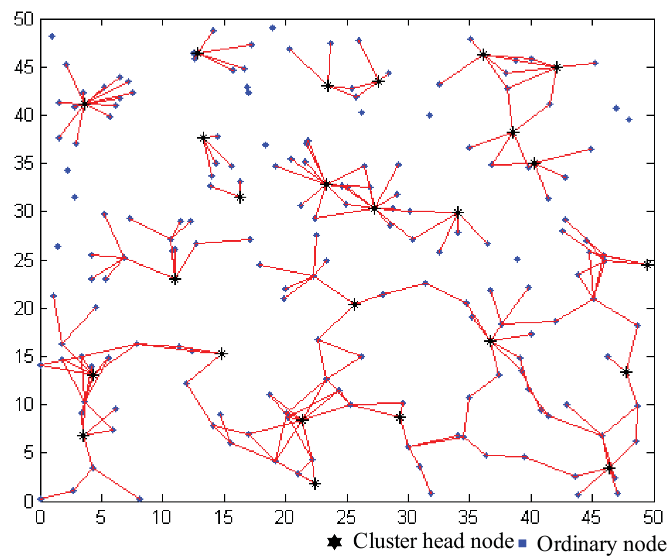

Fig. 10. Simulation result of clustering structure with 150 nodes distributed randomly.

algorithms. Fig. 7 shows the average rounds of communication in 20 experiments for various algorithms, when 10%, 20%, 30%, 40%, 50%, 60%, 70% and 80% of nodes are dead. The simulation conditions are 200 nodes of random distribution.

Meanwhile, we also conclude that the performance of all algorithms is low in fig. 6, since the dense deployment causes higher overload of clusters management. In fig. 5, it shows the same result that the life cycle doesn't increase linearly with the number of nodes increasing.

We further study how the number of nodes affects the performance of algorithms with nodes distributed randomly. Fig. 8 shows the average life cycle of various algorithms changing with the number of nodes in 20 experiments, and the simulation conditions are 100, 150, 200, 250 nodes in random distribution. It can be seen from Fig. 8 that, when the number of nodes is above 150, the properties of all algorithms decline more or less, but DSBCA algorithm still achieves the better performance in various number of nodes.

The curve of DSBCA is smoother than others, which indicates that the clustering structure is more stable. So DSBCA could adapt to to diverse distribution. However, the life cycle of other traditional algorithms drops sharply when the number of nodes exceeds 150. The decline in the fig. 8 is relative to the clustering structure and the selection of cluster head.

DSBCA can form more reasonable cluster structure to avoid frequent exchange of the nodes weight information and temporary cluster head broadcasting after the first clustering. As a result, the energy consumption decreases effectively. The cluster structure changes in each round in LEACH, HEED and WCA; nevertheless, DSBCA maintains relatively stable clustering structure in which switching of cluster head often occurs in the same cluster.

LEACH elects cluster head randomly without considering the residual energy of node; Heed considers the residual

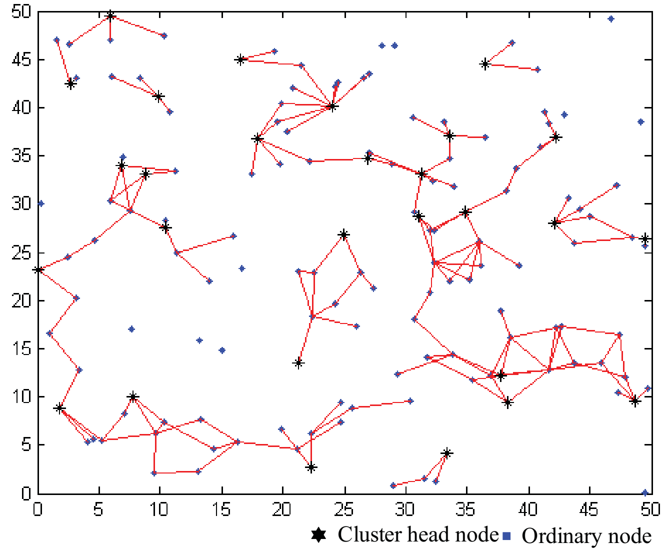

Fig. 11. Simulation result of clustering structure with 150 nodes distributed randomly.

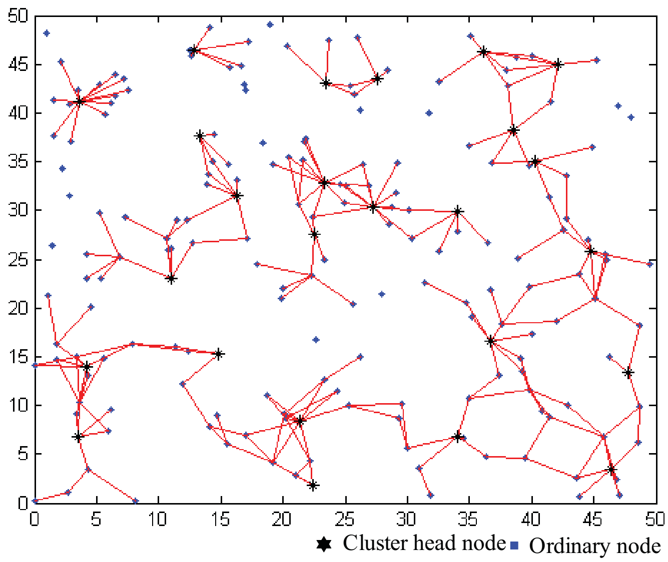

Fig. 12. Simulation result of clustering structure with 200 nodes distributed randomly.

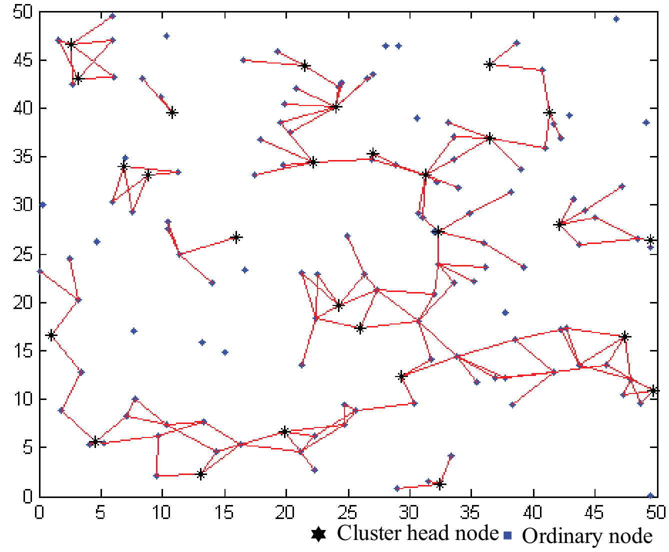

Fig. 13. Simulation result of clustering structure with 200 nodes distributed randomly.

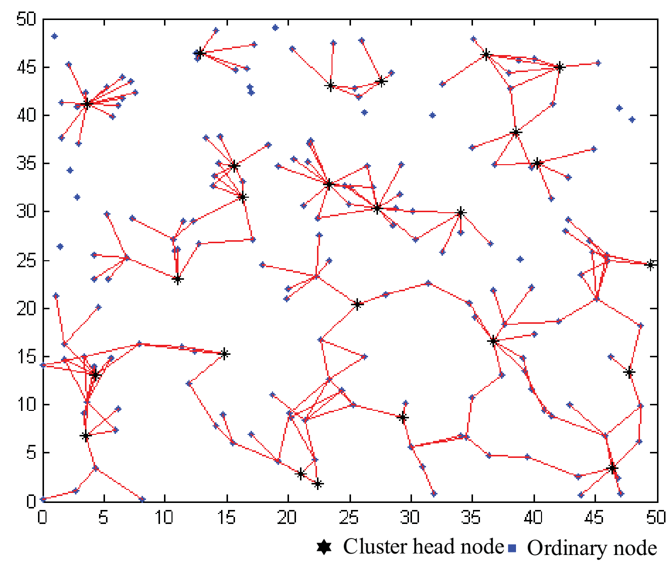

Fig. 14. Simulation result of clustering structure with 200 nodes distributed randomly.

energy, but the cluster structure formed by heed does not suit well the actual distribution; WCA needs each node to store the information of all nodes, resulting in excessive calculation and communication costs. Generally speaking, the clusters formed by DSBCA based on the distance from base station, distribution of nodes and residual energy accord with actual network. Hence, it achieves a better performance when the number of nodes changes.

The simulations of clustering structure in 150 nodes of random distribution in the 50 m $\times$ 50 m region are shown in fig. 9–11. The clustering structure is more balanced compared to other algorithms. DSBCA optimizes the energy consumption and ensures the life cycle in non-uniform distribution of adverse circumstances. In fig. 12–14, we can see the simulation results of clustering structure in 200 nodes of random distribution in the 50 m $\times$ 50 m region, and the same conclusion follows.

VI. Conclusion

In this paper, we propose a balanced clustering algorithm with distributed self-organization for WSNs of non-uniform distribution, taking into account optimal configuration of clusters. Compared with traditional clustering algorithms, the proposed algorithm can form more stable and reasonable cluster structure, and also improve the network life cycle significantly. The simulation result shows that the algorithm is feasible and has superior performance. In addition, the scenario we propose is scalable and works for different network sizes.

REFERENCES

[1] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *Mobile Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, 2002.

[2] I. Bekmezci and F. Alagöz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *Int. J. Distrib. Sensors Netw.*, vol. 5, no. 6, pp. 729–747, 2009.

[3] B. Koo and T. Shon, "Implementation of a WSN-based structural health monitoring architecture using 3D and AR mode," *IEICE Trans. Commun.*, vol. 93, no. 11, pp. 2963–2966, 2010.

[4] H. Hatime, K. Namuduri, and J. M. Watkins, "OCTOPUS: An on-demand communication topology updating strategy for mobile sensor networks," *IEEE Sensors J.*, vol. 11, no. 4, pp. 1004–1012, Apr. 2011.

[5] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro sensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, pp. 1–10.

[6] J. S. Lee and W. L. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors J.*, vol. 11, no. 9, pp. 2891–2897, Sep. 2012.

[7] B. A. Attea and E. A. Khalil, "A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks," *Appl. Soft Comput.*, vol. 12, no. 7, pp. 1950–1957, 2011.

[8] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Dec. 2004.

[9] C. H. Lin and M. J. Tsai, "A comment on HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 10, pp. 1471–1472, Oct. 2006.

[10] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw.*, Jul. 2001, pp. 70–84.

[11] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithms for mobile ad hoc networks," *Cluster Comput.*, vol. 5, no. 2, pp. 193–204, 2002.

[12] Y. Fernandess and D. Malkhi, "K-clustering in wireless ad-hoc networks," in *Proc. 2nd ACM Workshop Principles Mobile Comput. Conf.*, Oct. 2002, pp. 31–37.

[13] H. C. Lin and Y. H. Chu, "A clustering technique for large multihop mobile wireless networks," in *Proc. 51st Int. Conf. Veh. Technol.*, May 2000, pp. 1545–1549.

[14] M. Lehsaini, H. Guyennet, and M. Feham, "A novel cluster-based self-organization algorithm for wireless sensor networks," in *Proc. Int. Symp. Collabor. Technol. Syst. Conf.*, May 2008, pp. 19–26.

[15] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 12, pp. 1844–1856, Dec. 2009.

[16] C. J. Jiang, W. R. Shi, X. L. Tang, P. Wang, and M. Xiang, "Energy-balanced unequal clustering routing protocol for wireless sensor networks," *J. Softw.*, vol. 23, no. 5, pp. 1222–1232, 2012.

[17] N. Mitton, B. Sericola, S. Tixeuil, E. Fleury, and I. G. Lassous, "Self-stabilization in self-organized wireless multihop networks, *Ad Hoc Sensors Wireless Netw.*, vol. 11, no. 2, pp. 1–34, 2011.

[18] J. Anno, L. Barolli, A. Durresi, F. Xhafa, and A. Koyama, "Performance evaluation of two fuzzy-based cluster head selection systems for wireless sensor networks," *Mobile Inf. Syst.*, vol. 4, no. 4, pp. 297–312, 2008.

[19] H. Bagci and A. Yazici, "An energy aware fuzzy unequal clusteringalgorithm for wireless sensor networks," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jul. 2010, pp. 1–8.

[20] W. R. Heinzelman, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[21] G. Makihara, M. Yokomichi, and M. Kono, "Design of nonlinear controllers for active vehicle suspension with state constraints," *Artif. Life Robot.*, vol. 13, no. 1, pp. 41–44, 2008.

Ying Liao received the B.E. degree in communication engineering from the Institute of Information Science and Technology, Zhengzhou, China, the M.E. degree in communication and information system from the National Digital Switching System Engineering and Technological Research Center, Zhengzhou, and the Ph.D. degree from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2001, 2004, and 2012, respectively.

His current research interests include wireless sensor networks, system analysis and simulation, complex systems modeling, and complex networks.

Huan Qi received the B.S. and M.S. degrees in mathematics from the School of Mathematics and Statistics and the Ph.D. degree in system engineering from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 1982, 1987, and 1996, respectively.

He is currently a Full Professor with the Department of Control Science and Engineering, Director of System Engineer Committee and Mathematics Society for Hubei province, China. His current research interests include system analysis and simulation, multi-agent systems, complex systems modeling and complex networks, and decision and scheduling under uncertain information. Prof. Qi serves as a Committee Member of the Journal of Systems Engineering and Electronics.

Weiqun Li received the B.S. degree in mathematics from the School of Mathematics and Statistics and the M.E. degree in system engineering from the Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2011, respectively.

He is an Engineer with Huawei Company, Shenzhen, China. His current research interests include system analysis and simulation and wireless sensor networks.