J Syst Sci Complex (2018) 31: 234–243

Distributed Clustering Algorithm for Energy Efficiency and Load-Balance in Large-Scale Multi-Agent Systems*

LU Shaolei · FANG Hao · WEI Yue

DOI: 10.1007/s11424-018-7369-4
Received: 2 November 2017 / Revised: 21 November 2017

Abstract To improve the energy efficiency and load-balance in large-scale multi-agent systems, a large-scale distributed cluster algorithm is proposed. At first, a parameter describing the spatial distribution of agents is designed to assess the information spreading capability of an agent. Besides, a competition resolution mechanism is proposed to tackle the competition problem in large-scale multi-agent systems. Thus, the proposed algorithm can balance the load, adjust the system network locally and dynamically, reduce system energy consumption. Finally, simulations are presented to demonstrate the superiority of the proposed algorithm.

Keywords Clustering algorithm, distributed algorithm, energy efficiency, load-balance, multi-agent system.

1 Introduction

In large-scale multi-agent systems with the flat structure, communications among agents are complex and cost lots of extra energy, since adjacent agents may detect the same event and data. Therefore, the hierarchical structure is desired, where agents are grouped into clusters. A cluster head is responsible for collecting data from member agents in the cluster and instructing the members if necessary.

Clustering algorithms were generally applied in wireless sensor networks (WSNs)[1], where the cluster head collected data and transmitted the data to the base station. Low-energy

LU Shaolei · FANG Hao · WEI Yue
Department of Automation, Beijing Institute of Technology, Beijing 100081, China.
Email: lu_shaolei@126.com; fangh@bit.edu.cn; yuewei@bit.edu.cn.

*This work was supported by Projects of Major International (Regional) Joint Research Program NSFC under Grant No. 61720106011, the National Natural Science Foundation of China under Grant Nos. 61573062, 61621063, and 61673058, Program for Changjiang Scholars and Innovative Research Team in University under Grant No. IRT1208, Beijing Education Committee Cooperation Building Foundation Project under Grant No. 2017CX02005, Beijing Advanced Innovation Center for Intelligent Robots and Systems (Beijing Institute of Technology), Key Laboratory of Biomimetic Robots and Systems (Beijing Institute of Technology), Ministry of Education, Beijing, China.
◇*This paper was recommended for publication by Guest Editor XIN Bin.*

adaptive clustering hierarchy (LEACH)[2] is one of the early algorithms in this field. LEACH is a self-organized, adaptive cluster protocol which can randomly distribute energy evenly among the sensors in the network. However, the number of sensors in each cluster is the same and energy consumption is unbalanced[3]. In addition, the sensors in [2] were assumed to have the equal priority and as a result, each of them was chosen as a cluster head randomly, which may result in a fast death of some sensors. Further studies on the LEACH algorithms could be seen in [4, 5].

In order to avoid the randomization in [2], a hybrid, energy-efficient, distributed (HEED) cluster algorithm was proposed in [6], where cluster heads were chosen according to residual energy and another parameter named "communication cost". The cost here could be a function of neighbor proximity or cluster density. When sensors fell within the "ranges" of more than one cluster head, sensors could choose the cluster head according to the cost. Therefore energy efficiency and network lifetime are better than that of LEACH.

Besides the parameter of residual energy considered above, the distance to the base station (BS) was also considered in [7]. According to these two parameters, each node was assigned a rank by BS which maintained a global knowledge of the nodes. Furthermore, the division of the area, the calculation of the rounds, and the dynamic choices of cluster heads were all determined by BS. Thus, the system was central control and might lead to single point failure.

Wireless sensor networks in LEACH and HEED are of uniformly distributed without considering the random distribution of the sensors in practical applications. Thus the uniform clustering algorithm may result in an unbalanced topological structure in this case. Therefore recent researchers focus on the load-balance[8–10]. In [8], a min-heap mechanism was built to balance the load. However, global information such as R_{set} was required and the cluster heads were preset. In [10], a distributed self-organizing load balanced clustering algorithm (DSBCA) was proposed, where clustering radius was determined by density and distance. DSBCA could produce a reasonable size of the cluster by adjusting the radius of the cluster and improve the energy efficiency. However, the system topology reconstructed periodically in a global and absolute time interval, which was a strict condition for real applications.

The algorithms mentioned above in sensor networks are not suitable for multi-agent system networks. Generally, there is no base station in multi-agent systems (MASs) and the topology structures of MASs are distributed where multi-hop should be avoided as much as possible , such as the application in [11]. However, [11] merely applied the clustering algorithm in MAS simply without considering the situations of large-scale and competition conflicts.

The objective of this paper is to design a large-scale distributed cluster algorithm (LSDCA) to study energy efficiency and load-balance in large scale multi-agent systems. The proposed algorithm is expected to achieve better performance for large-scale multi-agent systems and the major contributions in this paper fall into two aspects:

(i) Proposing a concept of radian distribution (RD) motivated by isotropy, which can measure the information spreading capability of an agent in the large-scale multi-agent system, improve the probability of the agents with high radian distributions to be cluster heads and efficiently balance the load;

(ii) Proposing a local topological dynamic adjustment mechanism to adjust the topology in a distributed and local way, which can reduce the energy consumption and enhance the flexibility of the system.

The remainder of this paper is organized as follows. Section 2 describes the system model of MAS network. In Section 3, the proposed algorithm is detailed and simulations are presented in Section 4. Finally, conclusions are given in the last section of the paper.

2 System Model

Let A be the set of agents in MAS, $N = |A|$ be the number of agents in MAS, and $E \subseteq A^2$ be the set of links among the agents. Thus, an MAS network can be modeled as a graph $G = (A, E)$. $a_i \in A$ denotes agent i, v_i the neighbors of a_i, and $n_i = |v_i|$ the number of the neighbors of a_i. The message mode used here is adopted from [10], which includes the ID of the cluster head node (HID), the ID of the sending node (SID), the number of hops from the cluster head (HD) and the other information.

The energy model is required to calculate the energy consumption later. Here the energy model is introduced from [12], which is commonly used. Thus, the power consumption to transmit an l-bit message over a distance d_{ij} from agent i to agent j is given as follows:

$$E_{T_{ij}}(l, d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d_{ij}^2, & d_{ij} < d_0, \\ lE_{elec} + l\varepsilon_{mp}d_{ij}^4, & d_{ij} \geq d_0. \end{cases} \tag{1}$$

And the power consumption to receive the message is given as follows:

$$E_R(l) = lE_{elec}, \tag{2}$$

where E_{elec} denotes the electronic energy, which depends on factors such as the digital coding, modulation, filtering and spreading of the signal. $\varepsilon_{fs}d_{ij}^2$ and $\varepsilon_{mp}d_{ij}^4$ denote the amplifier energy, which depend on the distance from agent i to agent j and the acceptable bit-error rate. d_0 is the threshold for switching to a different model, which is set to be 5m adopted from [13].

3 Proposed Algorithm

LSDCA is a distributed algorithm, where the agents are self-organized without central control. This algorithm can be divided into three stages: Cluster head selecting phase, cluster building phase, and clusters connecting phase.

3.1 Cluster Head Selecting Phase

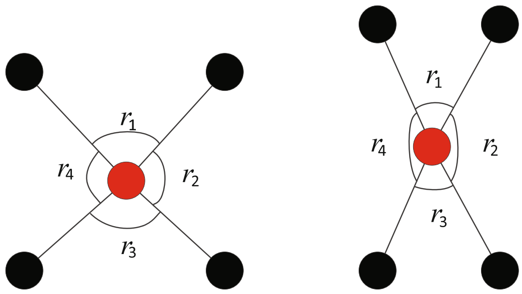
Here, it is assumed that cluster head selecting phase begins after multi-agent coalition formation or multi-agent task allocation. Besides the density in [10], the radian distributions (RDs) of agents is also considered in this paper. RD is an important factor in the large-scale multi-agent system, since more uniformly the neighbors of an agent distribute, the agent performs better information spreading capability. The RD of a_i is defined as follows.

Definition 3.1 RD of a_i implies the degree of the radians distribution of agents around a_i in the polar coordinate system.

The RD of a_i is given as follows.

$$RD\,(a_i) = \begin{cases} \dfrac{r \cdot \overline{r}}{\|r\| \cdot \|\overline{r}\|} * \dfrac{|r|}{bnn}, \\ \dfrac{r \cdot \overline{r}}{\|r\| \cdot \|\overline{r}\|} * \left(1 - \dfrac{|r| - 7}{bnn}\right), \end{cases} \tag{3}$$

where r is a radian vector and an element of r denotes the radian between two adjacent segments formed by a_i and its two neighbors separately. Each element in $\overline{r}$ is equal to the mean value of the elements in r. bnn means the limited number of neighbors to connect with. A simple example is given as Figure 1.



(a) High RD (b) Low RD

Figure 1 An example of high RD (Figure (a)) and low RD (Figure (b))

As shown in Figure 1, the red node in Figure 1(a) has higher RD than the red node in Figure 1(b), since in large-scale multi-agent systems, the former shows the capability of isotropy, which can enhance the information spreading in wide-range.

Thus the weight of a_i is given as follows.

$$w_i = \alpha RD_i + \beta D_i + \gamma \frac{R_i}{E_i}, \quad 0 \le \alpha, \beta, \gamma \le 1,\ \alpha + \beta + \gamma = 1, \tag{4}$$

where α, β, γ are the effect factors defined for specific applications, D_i the density of a_i, R_i the residual energy of a_i, and E_i the initial full energy of a_i.

In this phase, agents calculate their weights using (4) in a distributed way, and the agent with the highest weight in its k-hop or the others' k-hop is marked as a cluster head, otherwise, an ordinary agent.

3.2 Cluster Building Phase

Cluster head accepts agents according to the time order in [10] without considering the competitions among the ordinary agents. However the size of the cluster is limited, and conflicts may exist. Thus, a conflict resolution mechanism is proposed here. When an ordinary agent a_i receives a head-message from a cluster head a_j, a_i sends a join-message including a price pr_{ij}

to a_j. The price pr_{ij} is initialized as follows.

$$pr_{ij} = \frac{1}{1+d_{ij}} \cdot \frac{E_i}{E_i+R_i} + (n-1)\sigma, \quad 0<\sigma<1, R_i > E_{elec}, \tag{5}$$

where σ is a small parameter, and n is the times that a_i has been deported by clusters. The probability of joining a new cluster will be improved when n increases. When R_i decreases, a_i is eager to join in a cluster.

If the cluster size is smaller than the threshold, a_j accepts a_i and sends a payment py to it. The payment py_{ij} is given as follows.

$$py_{ij} = pr_{ij} \cdot \frac{d_{ij}}{\sum_{k\in\{M_j,a_i\}} d_{kj}}, \tag{6}$$

where M_j denotes the cluster members of a_j. According to this, the ordinary agent will quit the current cluster and join a new cluster to get a higher payment. As for a_j, if the current cluster size is smaller than the threshold, it will accept a_i, otherwise, only if a_i offers a higher price than members of a_j, a_j accepts a_i and break the agreement with the agent which offers the lowest price in its cluster.

In [10], a periodic replacement mechanism is proposed to balance the node energy consumption where $T(\omega)$ is the time interval to restart DSBCA. However the node energy consumption for information broadcasting could be heavy when $T(\omega)$ is short, or some agent may be exhausted before DSBCA restarts when $T(\omega)$ is long. Thus, $T(\omega)$ is hard to be designed, since the energies of agents change constantly. In this paper, an event-triggered mechanism is proposed. When the energy of a cluster head is lower than the mean energy of the cluster, as

$$R_i < \frac{\sum_{c_i} R_j}{|c_i|} \cdot \xi, \tag{7}$$

the cluster head sends restart-message to its members, and LSDCA restarts locally. The ξ in (7) is a discount parameter.

3.3 Cluster Connecting Phase

In sensor networks, all the cluster heads connect with a base station, but there is no connection among clusters. However there is no role of the base station in MASs considered here, and MAS networks are distributed connected as a connected graph. To connect different clusters, gate agents are necessary. Agents connected with more than one cluster simultaneously are called gate agents. Gate agents are chosen according to the rule of optimal energy consumption. In order to avoid conflict and save energy, an auction algorithm is used here. Ordinary agents calculate their energy consumption as (1) for connecting with two clusters where l is set to be 1 simply and then send bid-message to cluster heads. The winner will be a gate agent, and once the energy of the agate agent follows (7), it will find an alternative gate agent around it.

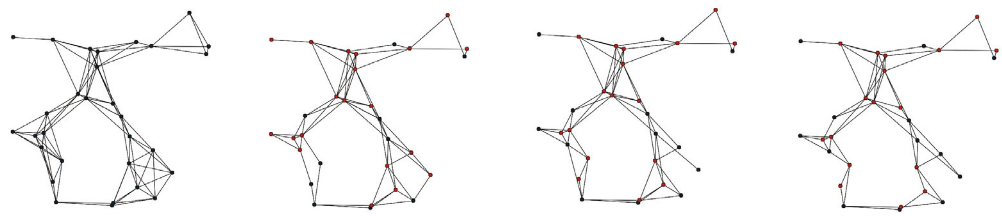
As described above, LSDCA considers the spacial distribution of the agents, and provides a conflict resolution for competitions. The most important difference from DSBCA, LSDCA is total distributed and asynchronous.

4 Simulation

In order to evaluate the performance of the proposed algorithm LSDCA, simulation experiments are presented in comparison with the former algorithm in this section. In comparison, any logic professional software is used as a simulation tool, which is a professional software to simulate distributed systems. The details of the experiment parameters are described as follows.

In multi-agent systems, multi-hop should be limited to a certain extent. Thus, it is assumed that an agent can only get information from the other agents within 2 hops. Agents are randomly distributed in a 50m $\times$ 50m area and the threshold of the cluster size is set as 15. DSBCA in [10] is compared with this proposed algorithm, i.e., LSDCA. Three scales of MASs are given to test the energy efficiencies and the network lifetimes respectively using the two algorithms and the numbers of agents in MASs are set as 30, 50, 80, respectively. The initial full energy of an agent is set to be 100.

At first, cluster connection of a multi-agent system network formed by 30 agents is given as Figure 2 to illustrate the performance using different algorithms. In Figure 2, the cluster heads are marked as red points and ordinary agents black points.

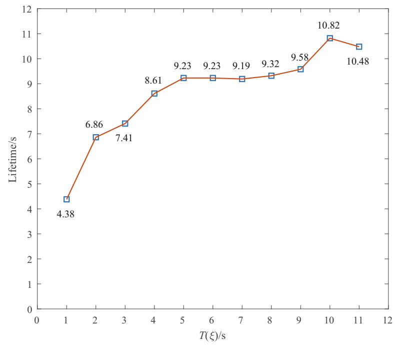


(a) Initial connection (b) DSBCA (c) DSBCA-RD (d) LSDCA

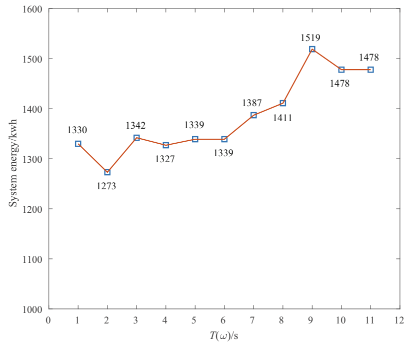
Figure 2 The initial network connection (Figure 2(a)), cluster network connection using DSBCA (Figure 2(b)), cluster network connection using DSBCA combined with RD (Figure 2(c)) and cluster network connection using LSDCA (Figure 2(d))

In Figure 2, cluster heads in Figure 2(c) is more evenly distributed than that in Figure 2(b). Evenly distribution of cluster heads can reduce the information transmission from ordinary agents to cluster heads. Compared with the network in Figure 2(c), the network in Figure 2(d) is with less redundant connections, because the conflict resolution mechanism is used in Figure 2(d).

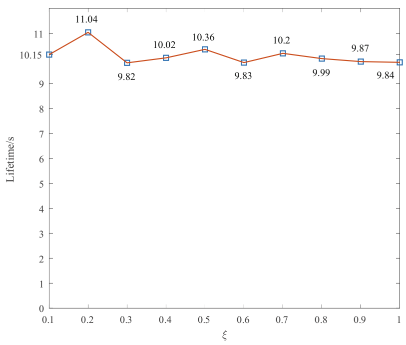
The network lifetime defined here is the period from algorithm start to an agent dies. In the network with 30 agents, $T(\omega)$ of DSBCA ranges from 1s to 10s and results of network lifetimes and system energies are given in Figure 3. Meanwhile, the ξ of LSDCA ranges from 0.1 to 1 and the respective network lifetimes and system energies are also given in Figure 3. And the simulations of MAS with 50 and 80 agents are given in Figures 4 and 5, respectively.



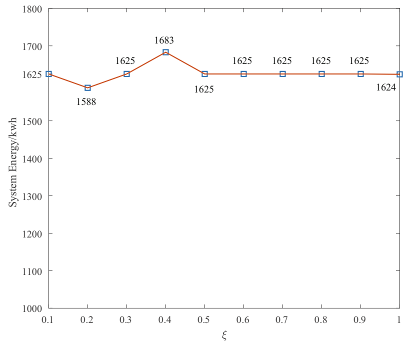
(a) Network lifetime of MAS with 30 agents using DSBCA



(b) System energy of MAS with 30 agents using DSBCA

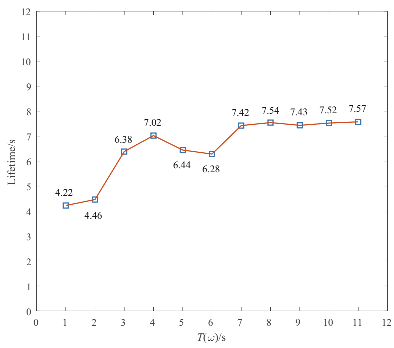


(c) Network lifetime of MAS with 30 agents using LSDCA agents

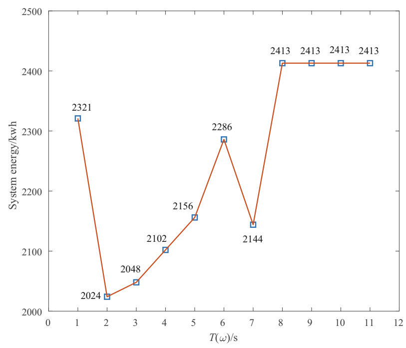


(d) System energy of MAS with 30 agents using LSDCA

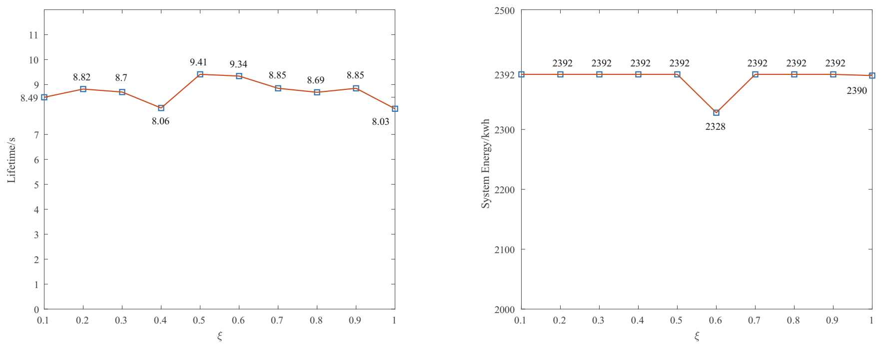
Figure 3 Simulation of MAS formed by 30 agents, including the system energies (Figure(b), Figure(d)) and network lifetime (Figure(a), Figure(c)) using DSBCA and LSDCA



(a) Network lifetime of MAS with 50 agents using DSBCA



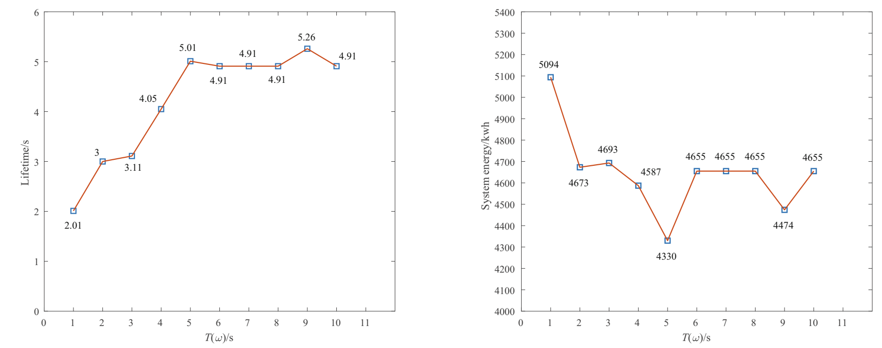
(b) System energy of MAS with 50 agents using DSBCA



(c) Network lifetime of MAS with 50 agents using LSDCA agents

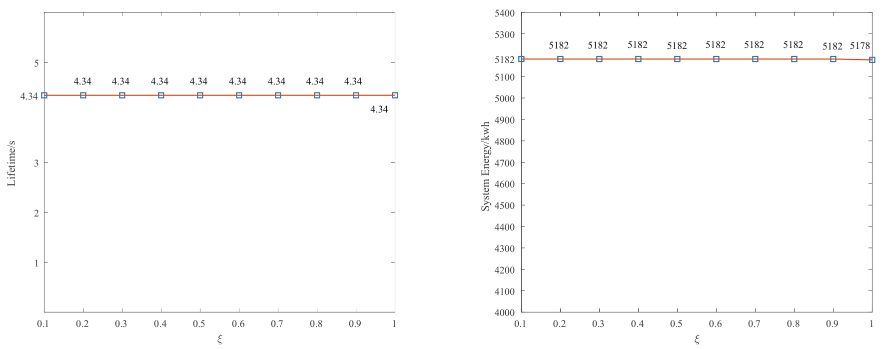
(d) System energy of MAS with 50 agents using LSDCA

Figure 4 Simulation of MAS formed by 50 agents, including the system energies (Figure(b), Figure(d)) and network lifetime (Figure(a), Figure(c)) using DSBCA and LSDCA



(a) Network lifetime of MAS with 80 agents using DSBCA

(b) System energy of MAS with 80 agents using DSBCA



(c) Network lifetime of MAS with 80 agents using LSDCA agents

(d) System energy of MAS with 80 agents using LSDCA

Figure 5 Simulation of MAS formed by 80 agents, including the system energies (Figure(b), Figure(d)) when network lifetime (Figure(a), Figure(c)) comes using DSBCA and LSDCA

As shown in Figure 3 through Figure 5, LSDCA performs more stable and energy efficient than DSBCA. It also can be seen that the parameter $T(\omega)$ is sensitive to network lifetime because shorter $T(\omega)$ makes higher frequency of system network reconstruction and more system energy will be consumed. However, in LSDCA, the reconstruction runs locally so that the system energy is consumed less than DSBCA. Besides, $T(\omega)$ has a close relationship with the times H that an agent has been chosen as a cluster head. A short $T(\omega)$ combined with a high priority of H will make the previous cluster heads lose its superiority fast, and the inferior agents be cluster heads, which will reduce the network lifetime and destroy the connection of the entire system. Obviously, $T(\omega)$ and the priority of H is hard to be designed in randomly distributed multi-agent system networks. However, compared with DSBCA, the effect of the range of ξ in LSDCA could be ignored, which can be seen in Figure 3 through Figure 5. The most important difference from DSBCA is that LSDCA runs totally distributed and asynchronously without global parameter such as in $T(\omega)$ DSBCA. Therefore LSDCA runs locally and save more system energy.

5 Conclusion

In large-scale multi-agent systems, how to save energy, balance the load and solve the conflicts are key problems. In this paper, a large-scale distributed cluster algorithm (LSDCA) is proposed, which is aimed to tackle these problems. Compared with the balanced clustering algorithm with distributed self-organization (DSBCA), LSDCA performs more stable and energy efficient, since it is total distributed and asynchronous. However, much work needs to be done in future, since the practical application environment is always complex, such as existing electromagnetic interference, defection and so on.

References

[1] Tilak S, Abu-Ghazaleh N B, and Heinzelman W, A taxonomy of wireless micro-sensor network models, *ACM Sigmobile Mobile Computing and Communications Review*, 2002, **6**(2): 28–36.

[2] Heinzelman W B, Chandrakasan A P, and Balakrishnan H, An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, 2002, **1**(4): 660–670.

[3] Quang P T A and Kim D S, Clustering algorithm of hierarchical structures in large-scale wireless sensor and actuator networks, *Journal of Communications and Networks*, 2015, **17**(5): 473–481.

[4] Lee J S and Cheng W L, Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication, *IEEE Sensors Journal*, 2012, **12**(9): 2891–2897.

[5] Baraa A A and Khalil E A, A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks, *Applied Soft Computing*, 2012, **12**(7): 1950–1957.

[6] Younis O and Fahmy S, HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, *IEEE Transactions on Mobile Computing*, 2004, **3**(4): 366–379.

[7] Alnuaimi M, Shuaib K, Alnuaimi K, et al., An efficient clustering algorithm for wireless sensor networks, *International Journal of Pervasive Computing* & *Communications*, 2015, **11**(3): 302–322.

[8] Kuila P and Jana P K, Energy efficient load-balanced clustering algorithm for wireless sensor networks, *Procedia Technology*, 2016, **6**(4): 771–777.

[9] Azharuddin M, Kuila P, and Jana P K, A distributed fault-tolerant clustering algorithm for wireless sensor networks, *Advances in Computing, Communications and Informatics* (*ICACCI*), 2013 *International Conference on. IEEE*, 2013, 997–1002.

[10] Liao Y, Qi H, and Li W, Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks, *IEEE Sensors Journal*, 2013, **13**(5): 1498–1506.

[11] Wang Q, Chen J, and Fang H, A construction method of fault-tolerant topology for multi-agent systems, *Pattern Recognition and Artificial Intelligence*, 2014, **27**(4): 356–362.

[12] Heinzelman W B, Chandrakasan A P, and Balakrishnan H, An application-specific protocol architecture for wireless microsensor networks, *IEEE Trans. Wireless Communications*, 2002, **1**(4): 660–670.

[13] Lung C H and Zhou C, Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach, *Ad. Hoc. Networks*, 2010, **8**(3): 328–344.