Accepted Manuscript

Genetic Algorithm for Energy-Efficient Clustering and Routing in Wireless Sensor Networks

Tianshu Wang , Gongxuan Zhang , Xichen Yang , Ahmadreza Vajdi

Please cite this article as: Tianshu Wang , Gongxuan Zhang , Xichen Yang , Ahmadreza Vajdi , Genetic Algorithm for Energy-Efficient Clustering and Routing in Wireless Sensor Networks, *The Journal of Systems & Software* (2018), doi: https://doi.org/10.1016/j.jss.2018.09.067

Highlights

- The proposed method improves search efficiency of the optimal solution.
- The fitness function is based on energy consumption to improve energy efficiency.
- The proposed method considers the load balancing.
- Simulation shows that our proposed method is better than the existing algorithms.

Genetic Algorithm for Energy-Efficient Clustering and Routing

in Wireless Sensor Networks

Tianshu Wang[1], Gongxuan Zhang[2], Xichen Yang[3], Ahmadreza Vajdi[2]

1. School of Information Technology, Nanjing University of Chinese Medicine, China

2. School of Computer Science and Engineering, Nanjing University of Science and Technology, China

3. School of Computer Science and Technology, Nanjing Normal University, China

Abstract:

Wireless sensor networks have been employed widely in various fields, including military, health care, and manufacturing applications. However, the sensor nodes are limited in terms of their energy supply, storage capability, and computational power. Thus, in order to improve the energy efficiency and prolong the network life cycle, we present a genetic algorithm-based energy-efficient clustering and routing approach GECR. We add the optimal solution obtained in the previous network round to the initial population for the current round, thereby improving the search efficiency. In addition, the clustering and routing scheme are combined into a single chromosome to calculate the total energy consumption. We construct the fitness function directly based on the total energy consumption thereby improving the energy efficiency. Moreover, load balancing is considered when constructing the fitness function. Thus, the energy consumption among the nodes can be balanced. The experimental results demonstrated that the GECR performed better than other five methods. The GECR achieved the best load balancing with the lowest variances in the loads on the cluster heads under different scenarios. In addition, the GECR was the most energy-efficient with the lowest average energy consumed by the cluster heads and the lowest energy consumed by all the nodes.

Keywords: wireless sensor networks; clustering algorithm; genetic algorithm; energy-efficiency; network life cycle; routing

1. Introduction

Wireless sensor networks (WSNs) integrate sensor technology, embedded computing technology, distributed information processing technology, and communication technology. WSNs have been employed widely in various fields, including military, national defence, environmental monitoring, traffic management, health care, manufacturing, and disaster prevention applications [1]. In a WSN, the sensor nodes collaboratively collect and process environmental and physical information from the area covered by the network and send the information to observers [2]. A monitoring area often requires the deployment of a large number of sensor nodes, but the sensor nodes are limited in terms of their computational, storage, and communication capacities in order to reduce costs. For instance, Micaz [3] developed by CrossBow is a representative type of sensor node, which is equipped with an Atmegal28L microprocessor, CC2420 chip, 128 KB Flash, and 4 KB RAM. The sensor node is supplied with limited battery power and it is difficult to provide secondary energy to the nodes. Thus, network failure occurs after more than a certain percentage of the nodes die. Therefore, reducing the energy consumption of sensor nodes and prolonging the network life-cycle is the key challenge for WSNs.

Previous studies have shown that the energy consumption required for transferring 1-bit is much more than that for processing 1-bit data [4]. Thus, reducing the transmitted or received data sizes for sensor nodes and optimizing data transmission routing between the nodes can effectively reduce the energy consumed by the network. Clustering algorithms [5] divide the network into multiple independent clusters, where each cluster comprises a cluster head (CH) node and multiple cluster member (CM) nodes. The CH node is responsible for receiving data from the CM nodes. By using effective data aggregation algorithms, the CH can remove redundant or incorrect data so the large amounts of collected data are merged into a small amount of meaningful information. Therefore, efficient clustering algorithms can reduce the data traffic and optimize the topology, thereby improving the energy efficiency of WSNs.

Cluster-based WSNs usually comprise two types: (i) those with temporary CHs [6] and (ii) those with permanent CHs [7]. In the first type, the sensor nodes have a relatively fair energy supply and equal status, and all the nodes have a chance of being selected as the CH. For instance, at moment t_1, node A is selected to become the CH. However, at moment t_2, node A may only act as the CM. In the second type, the CH nodes and CM nodes are permanent. The CH nodes are also called gateway nodes or relay nodes, which have a higher energy supply. During the operation of the network, the permanent CHs manage all of the sensor nodes in their corresponding clusters. The status of the CHs is stable, but the relationships between each CH and its CMs may change over time. For instance, at moment t_1, the CH of CM a is node A. However, at moment t_2, node a is likely to join CH B. We need to employ clustering algorithms with different mechanisms according to the two different types of cluster-based WSNs. In this study, we consider a clustering algorithm for the second type: WSNs with permanent CHs. Most previous studies of clustering algorithms have considered the first type, and the second type has rarely been investigated. The second type of WSNs are important for the research community because of the following reasons: (i) they are more energy efficient because CH nodes with a higher computing capacity can efficiently operate complex data aggregation algorithms; (ii) they are more secure because the CHs possess more storage and they have a higher capacity, and hence, complex encryption algorithms can be executed and some trusted hardware modules can be equipped [8]; and (iii) the ordinary nodes only need to send the collected data to their corresponding head nodes without conducting the work of CHs, and thus the network life cycle can be prolonged.

It is important to note that devising an effective clustering algorithm with high energy efficiency and load balancing for the second type of large-scale WSNs is an NP-hard problem. If the network has a CH nodes and b ordinary nodes, then there will be a^b clustering schemes. If routing is considered and the average number of neighbouring nodes of each CH is c, then there will be c^a routing schemes. Therefore, for large-scale WSNs, calculating the optimal clustering and routing scheme has high time complexity. Metaheuristic algorithms [9] such as genetic algorithms (GAs) [10] can solve this problem quickly and efficiently.

In this study, we propose a GA-based energy-efficient clustering and routing algorithm (GECR), which employs a GA to obtain the optimal solution. The main contributions of this study are as follows:

- The network time is divided into multiple rounds. The sink node needs to run the GA separately in each round. The optimization objective for the clustering and routing schedule is related to the distances between the nodes. The locations of the nodes are fixed, so the optimal solution for a certain network round is related to the optimal solution for the previous round. In contrast to some traditional GA-based clustering and routing algorithms [11-13], we add the optimal solution from the previous network round to the initial population in the current network round to improve the search efficiency.

- In order to guarantee the energy efficiency, the fitness functions of some algorithms [14-16] are based on the total transmission distances between the nodes. The energy consumption is related to but not absolutely equal to the distance. Algorithms that construct the fitness function based on the total distance can only obtain the final solution with the shortest total distance. In the proposed method, we combine the clustering and routing scheme into a single chromosome and calculate the total amount of energy consumed for clustering and routing together. The fitness function is constructed directly based on the energy consumption of the whole network. In this manner, we can finally obtain the solution with the lowest energy consumption, and thus the final energy efficiency can be improved.

- Load balancing is an important criterion for evaluating clustering and routing algorithms. Many GA-based algorithms [15-18] do not consider load balancing. In addition, although some algorithms [14; 19; 20] do consider load balancing, they only count the number of CMs when calculating the loads on CHs. However, the CH needs to transmit the data from its previous hop nodes in addition to the data from its CMs. In the proposed method, we add the previous hops to the loads on each CH, thereby improving the accuracy of the load calculations.

- Simulations indicated that the performance of our proposed algorithm was better than that of some existing algorithms [14-17; 20] in terms of load balancing, network life cycle, and energy consumption. Thus, the proposed GECR always had the lowest variances in the loads on the CHs under different scenarios. In terms of the network life cycle, GECR had the most living nodes at most times. In addition, GECR consumed the smallest amount of energy in all of the network rounds.

The remainder of this paper is organized as follows. In Section 2, we discuss related research. In Section 3, we present the network model and some terminology. In Section 4, we give some preliminary details used in this study. In Sections 5 and 6, we present the proposed algorithm and the experimental results. In Section 7, we give our conclusions.

2. Related Work

2.1 WSNs with Temporary CHs

Many studies [21-25] have investigated clustering and routing algorithms for WSNs with temporary CHs. The low-energy adaptive clustering hierarchy (LEACH) algorithm [6] is one of the best known hierarchical routing protocols based on clustering, where it divides the network time into multiple rounds. In each round, all of the sensor nodes start to calculate a probability value that determines whether the sensor node can be a CH or not. The nodes that can become CHs broadcast their own information to notify other nodes to join their corresponding clusters. However, LEACH only considers a single-hop network model and it is only suitable for small scale WSNs. In addition, LEACH does not consider the residual energy of the nodes. Some of the nodes with a low amount of residual energy may become a CH, and thus, the protocol can speed up the failure of the network. In order to improve the LEACH algorithm [6], a large number of LEACH-based clustering and routing algorithms [26-31] have been proposed such as LEACH centralized clustering (LEACH-C) [26], hybrid energy-efficient distributed clustering (HEED) [27], balanced clustering algorithm with distributed self-Organization (DSBCA) [19], LEACH medium access control algorithm (LEACH-MAC) [28], and LEACH affinity propagation clustering (LEACH-AP) [29]. The first type of WSNs with temporary CHs can be divided into homogeneous and heterogeneous variants according to the initial energy of the nodes. All of these aforementioned algorithms assume that the sensor nodes are homogeneous and they are not suitable for heterogeneous WSNs. Therefore, many studies have attempted to develop suitable algorithms for heterogeneous WSNs. Distributed energy-efficient clustering [32] is a suitable clustering algorithm for multi-level heterogeneous networks. This algorithm estimates the average energy by considering all the nodes in the network, before selecting the CHs based on the ratio between the average energy and the residual energy of the sensor nodes. Saini et al. [33] proposed a heterogeneous network model that divides the network into five levels by calculating a single parameter value. This model estimates the life cycle for different levels as the 0-level, 1-level, 2-level, 3-level, and 4-level. Elbhiri et al. [34] proposed an energy prediction-based clustering tree control algorithm that considers the link quality and packet loss rate as two important factors that can affect the energy consumption.

Metaheuristic algorithms such as GAs are good solutions to the clustering and routing problems in WSNs. Many previous studies [11; 16-18; 35] have focused on GA-based clustering and routing methods for WSNs with temporary CHs. Liu et al. [17] proposed a LEACH with GA (LEACH-GA) clustering method, where each sensor node needs to generate a random number in the set-up phase. If the random number for a node is less than a threshold, the node can become a candidate CH. All of the candidate nodes send their information to the sink node. The sink node then runs the GA to determine the clustering result with the lowest energy consumption. However, LEACH-GA only considers the total energy consumption, and not the residual energy of the nodes. Elhoseny et al. [11] proposed a GA-based method that optimizes the clustering result of heterogeneous WSNs, where the residual energy, total energy consumption, and data transmission distance are all considered in the fitness function. Peiravi et al. [18] proposed an optimal clustering method that uses a multi-objective two-level GA. The top level obtains clustering schemes where the network life cycle is optimized and the low level used in each cluster aims to obtain the most efficient topology for data transmission. Shokouhifar et al. [16] introduced a hybrid clustering-based routing protocol called application-specific low power

routing (ASLPR), which can optimally balance the energy consumption among the sensors. In order to select the optimal CHs, ASLPR considers all the factors that can affect the network performance, e.g., distances from the sensors to the sink, residual energy, and distances between CHs. However, ASLPR requires that all the sensor nodes know the situations of all the other nodes, which wastes energy and storage capacity.

2.2 WSNs with Permanent CHs

Several clustering and routing algorithms have been proposed for WSNs with permanent CHs. Younis et al. [36] presented the load-balanced clustering (LBC) algorithm, which employs gateway nodes as the administration centre to process the data from ordinary nodes and forward it to the remote nodes. Based on LBC, Kuila et al. [37] proposed an energy efficient LBC algorithm called EELBCA, which combines the remaining energy of the CH nodes with the number of their neighbouring nodes as the clustering parameter. Low et al. [38] developed a clustering algorithm that uses a breadth-first search tree to find the CH with the minimum load. Other studies [39-43] have shown that metaheuristic algorithms (e.g., GA [10], particle swarm optimization algorithm [44], ant colony algorithm [45]) can solve the clustering and routing problem for WSNs.

Some studies have applied metaheuristic algorithms to WSNs with permanent CHs [14; 15; 20; 46]. In one method [46], the particle swarm optimization algorithm is employed to determine the energy efficient clustering and routing schemes. The fitness function of the clustering scheme is based on the ratio of the estimated life cycle and average transmission distance. The total distance and hops for data transmission are used to construct the fitness function in the routing scheme.

Gupta et al. [15] developed an energy efficient routing algorithm called GA-based routing (GAR) by using a GA to compute a new routing schedule. Gateway nodes are grouped into a chromosome in the network. The initial population is a collection of randomly generated chromosomes, where each gene represents the next hop of the corresponding gateway node. The energy consumed by sensor nodes is related to the data transmission distances, so GAR employs the following distance-related fitness function:

$$Fitness(k) = \frac{1}{\sum_{i=0}^{N-1} Dist(C_i, Crom(k,i))}, \tag{1}$$

where $Fitness(k)$ denotes the fitness value of the k-th chromosome in the population, $Crom(k,i)$ is the value of the i-th gene in the k-th chromosome, and $Dist(C_i, Crom(k,i))$ denotes the distance between node i and its next hop. Thus, the chromosome with the highest fitness value can be selected. The final selected routing schedule has low energy consumption because the transmission distance is optimal. However, GAR only considers the energy consumption between the gateways, whereas it ignores the energy consumption between the ordinary nodes and gateways.

In order to balance the network loads, Kuila et al. proposed a GA-based clustering method called the genetic algorithm-based load balanced clustering algorithm (GALBCA) [20], where each chromosome represents a clustering schedule in the population. In a chromosome, the value of the i-th gene denotes the CH for node i. The algorithm can obtain a load-balanced clustering schedule by minimizing the standard deviation (σ) of the CHs, which is given by:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m}(\mu - W_j)}{m}}, \tag{2}$$

where $\mu = \sum_{i=1}^{n} d_i/m$, d_i is the load on sensor node s_i, m is the number of CHs, and W_j is the overall load of CH g_j. GALBCA considers load balancing but it ignores the energy consumption factor, which is very important in WSNs.

A GA-based clustering and routing algorithm (GACR) [14] was proposed to solve the combined energy and load balancing problem. GACR applies a GA to clustering and routing. The fitness function for clustering is related to the average transmission distance and the standard deviation of the ratio of the residual energy. The fitness function for routing is constructed based on the total transmission distances and hops between the CHs. GACR computes the optimal clustering and routing schedules separately, and thus the load on each CH can only comprise the number of CM nodes. However, the CHs in the network also need to transmit the data from the previous hop CHs in addition to collecting the data from the CMs. Thus, the load on the CHs is not fully balanced.

Load balancing is an important criterion for evaluating clustering and routing algorithms. However, when calculating the loads on a CH, most of the aforementioned algorithms [14; 19; 20] that consider load balancing only take the CMs into account. In addition, some GA-based algorithms [14-16] try to minimize the transmission distance to reduce the energy consumption. However, in some cases, the distance is not equal to the energy consumption in the network. Moreover, these algorithms encode clustering and routing separately. Therefore, the energy consumption cannot be estimated globally. By contrast, our proposed GECR algorithm encodes clustering and routing in the same chromosome, and it directly uses the amount of energy consumed as one of the parameters for the fitness function. In addition, GECR adds the number of previous hops (previous CHs) when balancing the load on the CHs.

3. Network Model and Terminology

In this section, we first introduce the network topology and energy model for the proposed GECR algorithm, before providing the terminology used in this study.

3.1 Network Topology

Based on the characteristics of routing connections, the topologies of WSNs can be classified according to two categories: flat and hierarchical [47]. A flat network includes a sink node and multiple ordinary nodes, as shown in Fig. 1(a). The ordinary nodes have the same status and functions. In flat routing protocols [48-50], these nodes transmit data directly or through multiple hops to the sink node. The flat topology is simple and highly robust. However, all of the sensors in this topology need to keep updating the routing tables, so the network has poor scalability and low efficiency. In addition, the energy of the nodes may be wasted sometimes because there are no managers (e.g., CHs) in the flat topology.

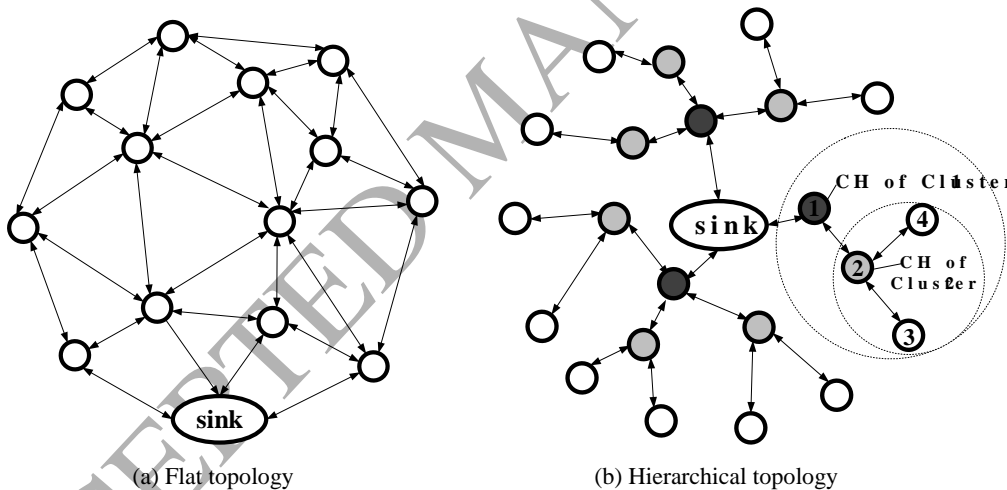(a) Flat topology   (b) Hierarchical topology

Fig. 1 Examples of a flat topology network and hierarchical topology network

In the hierarchical topology network, the sensor nodes are divided into multiple clusters. Each cluster comprises one CH and multiple CMs. The CH is responsible for managing its cluster by collecting and aggregating information from the CMs, and then sending the fused data to its next hop node or the sink node. Fig. 1(b) shows an example of a hierarchical topology network, where the large clusters are divided into multiple small clusters and the small clusters can continue to be divided into smaller subsets. The division process is repeated until the clusters satisfy certain conditions. In Fig. 1(b), sensor nodes 3 and 4 send the collected data to CH 2, which aggregates the data from cluster 2 (nodes 2, 3, and 4) and forwards the fused data to CH 1. CH 1 then aggregates the data from cluster 1 (nodes 1, 2, 3, and 4) and forwards the fused data to the sink node. The hierarchical topology has the following advantages: 1) by running an effective data aggregation algorithm, the CHs can reduce the amount of redundant and faulty data, thereby reducing the energy consumption; and 2) CMs only need to communicate with their own CHs. In addition, the CHs only need to communicate with their CMs and the next hops. Therefore, the hierarchical routing scheme can eliminate unnecessary interactions between nodes to improve the stability of the network. In general, hierarchical topologies can achieve better

performance compared with flat topologies [51]. Our proposed GECR algorithm employs a hierarchical topology.

In the proposed method, the gateways are equipped with more computing power, larger amounts of storage, and better energy supplies, and thus they play the role of CHs. However, the energy supply is still limited for the CHs because the nodes are battery-driven. After a CH dies, no node can replace it. Therefore, the CHs in the network are more important than the ordinary nodes. In our method, similar to the LEACH protocol, the network's time is divided into multiple rounds. Each round comprises a set-up phase and a steady-state stage. In the set-up phase, the optimal clustering and routing scheme for the current round is calculated using the proposed GECR algorithm in the sink node. In the steady-state stage, according to the calculated optimal clustering and routing scheme, the CHs collect and aggregate data from the CMs and send the aggregated data to the sink node through multi-hops.

3.2 Energy Model

Energy plays an important role in the construction and operation of networks because the network life cycle depends on the residual energy of each sensor node. The sensor nodes have limited energy and they need to consume some of this energy when sending or receiving data. We need to adopt suitable energy consumption model to calculate the energy consumption of nodes. In the research of clustering and routing in WSNs, the first-order radio model [6] is the most common energy consumption model and it has been widely used in many methods, e.g. GACR [14], GALBCA [20], HEED [27] and LEACH-MAC [28]. Thus, in the proposed method, we use the first-order radio model as the energy consumption model, as shown in Fig. 2.
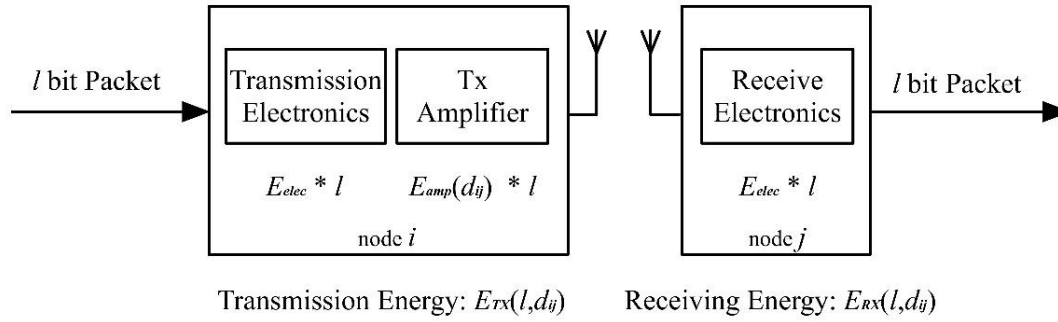

Fig. 2 Energy consumption model

The amount of energy consumed when transmitting an l-bit packet from node i to node j can be represented by:

$$E_{TX}(l, d_{ij}) = E_{elec} * l + E_{amp}(d_{ij}) * l = \begin{cases} E_{elec} * l + \varepsilon_{fs} * l * d_{ij}^2 & if \ d_{ij} < d_0 \\ E_{elec} * l + \varepsilon_{mp} * l * d_{ij}^4 & if \ d_{ij} \geq d_0 \end{cases}, \tag{3}$$

where E_{elec} is the energy required for driving and controlling electronic components, $E_{amp}(d_{ij})$ represents the energy consumed by signal amplification when transmitting 1-bit data, ε_{fs} and ε_{mp} are factors for the free space model and multipath model, respectively, and d_0 is the distance threshold, which is calculated as:

$$d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}. \tag{4}$$

According to Eq. 3, $E_{Tx}(l, d_{ij})$ depends on d_{ij}, which is the Euclidean distance between node i and node j. If the distance d_{ij} is less than the threshold d_0, the propagation of the wireless signal follows the free space model and the transmission energy is proportional to the square of d_{ij}; otherwise, the propagation of the wireless signal follows the multipath model and the transmission energy is proportional to the fourth power of d_{ij}. If we assume that the coordinates of nodes i and node j are (x_i, y_i) and (x_j, y_j), respectively, then the distance between them can be expressed as:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{5}$$

The node consumes the following amount of energy when receiving an l-bit packet:

$$E_{Rx}(l) = E_{elec} \times l. \tag{6}$$

3.3 Terminology

Some of the terminology used in the proposed algorithm is listed below.

1) The set of CHs is denoted by $H = \{h_1, h_2, ..., h_{nH}\}$ and nH indicates the number of CHs. The sink node is the final destination of the data, so it acts as a special CH and it is denoted by h_{nH+1}.

2) The set of ordinary nodes is denoted by $M = \{m_1, m_2, ..., m_{nM}\}$ and nM indicates the number of ordinary nodes.

3) n denotes the number of all nodes in the network:

$$n = nH + nM + 1. \tag{7}$$

4) $sumE$ denotes the total energy consumed in the r-th round where it comprises the energy used for clustering and routing:

$$sumE = cluteringE + routingE. \tag{8}$$

5) mhE_{ij} denotes the energy consumed during communication between CM m_i and CH h_j. If the energy required for sending data from m_i to h_j is denoted by $sendmhE_{ij}$ and the energy required for receiving data from h_j to m_i is denoted by $recmhE_{ji}$, then mhE_{ij} is formulated as:

$$mhE_{ij} = sendmhE_{ij} + recmhE_{ij}, 1 \leq i \leq nM, 1 \leq j \leq nH. \tag{9}$$

6) hhE_{ij} denotes the energy consumed during communication between CH h_i and CH h_j. If data are transmitted from h_i to h_j, the energy required for sending data is denoted by $sendhhE_{ij}$ and the energy required for receiving data is denoted by $rechhE_{ji}$, then hhE_{ij} is formulated as:

$$hhE_{ij} = sendhhE_{ij} + rechhE_{ji}, 1 \leq i \leq nH, 1 \leq j \leq nH + 1. \tag{10}$$

7) d_{max} denotes the maximum communication range of the sensor nodes.

8) $d(h_i, h_j)$ denotes the distance between CH nodes h_i and h_j. $d(m_i, h_j)$ denotes the distance between CM m_i and CH h_j.

9) $pCH(m_i)$ is the set of potential CHs for CM m_i. The CHs are within the communication range of m_i. Hence,

$$pCH(m_i) = \{h_j | \forall h_j \in H \wedge 0 < d(m_i, h_j) < d_{max}\}. \tag{11}$$

10) CH_m_i denotes the CH selected for CM m_i. Thus, we can obtain $CH\_m_i \in pCH(m_i)$

11) $rCH(h_i)$ is the set of all the CHs within the communication range of CH h_i. The sink node may also be a member of $rCH(h_i)$. Therefore,

$$rCH(h_i) = \{h_j | \forall h_j \in H \cup h_{nH+1} \wedge 0 < d(h_i, h_j) < d_{max}\}. \tag{12}$$

12) $nextHops(h_i)$ is the set of the potential next hop nodes of CH h_i. To avoid wasting energy, the next hop must be closer to the sink node than h_i, i.e.,

$$nextHops(h_i) = \{h_j | \forall h_j \in rCH(h_i) \wedge d(h_j, h_{nH+1}) < d(h_i, h_{nH+1})\}. \tag{13}$$

13) $nextHop\_h_i$ denotes the next hop selected for node CH h_i. Then, we can obtain $extHop\_h_i \in nextHops(h_i)$. If the sink node is within the communication range of h_i, then it may become the next hop of h_i.

14) $nL(h_i)$ indicates the number of loads on CH h_i. If the number of the member nodes of h_i is denoted by $nCM(h_i)$ and $nHops(h_i)$ denotes the number of CHs that select h_i as their next hop, then $nL(h_i)$ is

formulated as:

$$nL(h_i) = nCM(h_i) + nHops(h_i). \tag{14}$$

15) $\grave{E}_{h_i h_{nH+1}}$ denotes the total energy consumed when data are transmitted from CH h_i to the sink node through multiple hops.

16) $E_{residual}(h_i)$ denotes the residual energy of CH h_i and $E_{residual}(m_i)$ denotes the residual energy of CM m_i.

4. Preliminaries

4.1 Overview of GA

A GA is a type of metaheuristic that searches for an optimal solution by simulating the natural process of evolution. Fig. 3 illustrates the general process employed by a GA. First, some solutions are initialized randomly to form an initial population. These solutions are called individuals and each solution comprises one or more chromosomes made of a set of characters or strings. In an individual chromosome, each unit (a character or a string) is called a gene. After generating the initial population, a fitness function is used to select the individuals with higher performance. Two individuals are then selected randomly as parents to generate two new individuals as children based on a crossover operation. A mutation operation is executed to produce new children. Next, the fitness function is used to evaluate the fitness of the new children and the parents. It should be noted that the new children are generated after mutation and the parents are generated after selection. Finally, the algorithm selects half the individuals with better performance and adds them to the new population. If the new population meets the termination condition, the algorithm ends; otherwise, the algorithm enters the next generation.
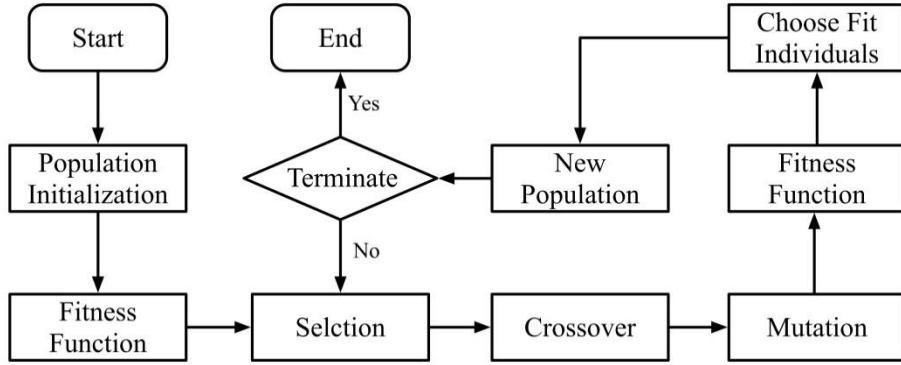
Fig. 3 General process employed by a GA

4.2 Problem Formulation

The sensor nodes are provided with limited energy, so improving the energy efficiency and maximizing the network life cycle is the main target of clustering and routing algorithms. In our proposed method, the network's time is divided into multiple rounds. If the energy consumption during each round is minimized, then the energy efficiency within the life cycle can be guaranteed. A Boolean variable c_{ij} is first defined to assess whether CM node m_i is assigned to CH node h_j. Then, c_{ij} can be calculated as follows.

$$c_{ij} = \begin{cases} 1, & If\ m_i\ is\ assigned\ to\ h_j dfsddf \\ & \forall i,j: 1 \le i \le nM, 1 \le i \le nH \\ 0, & otherwise ffffffffffffffffff \end{cases} \tag{15}$$

Eq. 8 shows that the total energy consumed in each round comprises $clusteringE$ and $routingE$. $clusteringE$ is the communication energy consumed between all the members and their corresponding CHs:

$$clusteringE = \sum_{i=1}^{nM} \sum_{j=1}^{nH} mhE_{ij} * c_{ij}. \tag{16}$$

$routingE$ is the energy consumed when transmitting the data from all the CHs to the sink node:

$$routingE = \sum_{i=1}^{nH} \grave{E}_{h_i h_{nH+1}}. \tag{17}$$

Then, the optimization objective for the clustering and routing problems is:

$$minimize: sumE = \sum_{i=1}^{nM} \sum_{j=1}^{nH} mhE_{ij} * c_{ij} + \sum_{k=1}^{nH} \grave{E}_{h_k h_{nH+1}}, \tag{18}$$

$$subject\ to: \sum_{j=1}^{nH} c_{ij} = 1, 1 \le i \le nM \tag{19}$$

$$\sum_{j=1}^{nH} d(m_i, h_j) * c_{ij} < d_{max}, 1 \le i \le nM, m_i \in M, h_j \in H. \tag{20}$$

Eq. 19 states that any CM node can only join one CH node h_j. Eq. 20 states that the distance between m_i and h_j must be within the maximum communication range of m_i if m_i is assigned to h_j.

5. Proposed Algorithm

Next, we present the detailed design of the proposed GECR algorithm. The chromosome representation (Section 5.1) is introduced first. We then present the population initialization method (Section 5.2) and the fitness function (Section 5.3). Finally, the crossover operator and mutation operator are explained in Section 5.4.

5.1 Chromosome Representation

Before the design of the chromosome, the identities of the nodes should be checked and updated. $ID(h_i)$ denotes the new identity of the CH nodes:

$$ID(h_i) = \begin{cases} i, fff\ If\ i \le nH \\ n, fffIf\ i = nH + 1 \end{cases}, \tag{21}$$

where i is the original identity of the CH nodes. If node h_i is an ordinary CH ($i \le nH$), then its new identity remains the same; otherwise, if h_i is a sink node ($i = nH + 1$), then its new identity is n. For example, in the WSN in Fig. 4, the new identity of CH node h_3 is still 3, but the new identity of sink node h_5 becomes 17. The new identity of CM node m_i is denoted by:

$$ID(m_i) = i + nH , \tag{22}$$

where i is the original identity of the CM nodes. For example, the new identity of CM node m_5 becomes 9.

In the proposed GECR algorithm, we encode the clustering scheme and the routing scheme in the same chromosome, which is called a *scheme*. The length of the *scheme* chromosome is n-1, which is the number of all the sensor nodes except the sink node. $scheme[k]$ denotes the k-th gene of the *scheme* chromosome and k denotes the new identities of the sensor nodes, which are calculated using Eq. 21 and Eq. 22. The *scheme* chromosome comprises two parts. The first part is the routing scheme and $k \le nH$. As shown in Eq. 23, the next hop of the CH node h_k has two options:

$$nextHop\_h_k = \begin{cases} h_{jaaa}, & if\ scheme[k] = j, k \le nH, j \le nH, \\ h_{nH+1}, & if\ scheme[k] = n, k \le nH. aaaaaaa \end{cases} \tag{23}$$

If the k-th gene is j and both k and j are less than or equal to nH, then the next-hop node of the head node h_k is h_j. However, if the k-th gene is n and k is less than or equal to nH, then the next-hop node of h_k is the sink node h_{nH+1}.

The second part of the *scheme* chromosome is the clustering scheme and $nH < k < n$. The CH node of member node m_{k-nH} is calculated as follows:

$$CH\_m_{k-nH} = h_j, if\ scheme[k] = j, nH < k < n, j \le nH. \tag{24}$$

If the k-th gene is j and j is less than or equal to nH, then the CM node m_{k-nH} with identity k is assigned to the CH node h_j. It is important to note that if there are multiple genes with the same value in this part, then the CM nodes corresponding to the genes are assigned to the same CH node.

An example of a WSN clustering and routing scheme is illustrated in Fig. 4. The internal numbers of the nodes indicate the identities of the sensor nodes. Transmissions from CM nodes to their corresponding CH nodes are indicated by the solid arrows and the routing between CH nodes are indicated by dashed arrows. In total, the WSN has 17 sensor nodes, which comprise one sink node, four CH nodes ($H = \{h_1, h_2, h_3, h_4\}$) and 12 CM nodes ($M = \{m_1, m_2, \ldots, m_{12}\}$). Fig. 5 illustrates the corresponding chromosome *scheme* for the clustering and routing scheme in Fig. 4. In Fig. 5, the numbers in the '*ID*' row indicates the identities of sensor nodes, which are related to the corresponding gene position. The chromosome *scheme* consists of two parts: routing scheme and clustering scheme. The gene positions in the first part indicate the identities of the CHs in the network. The allele of a gene position is the identity of the next hop of the corresponding CH. For example, the allele of the gene position 2 in *scheme* is 17, which indicates that the next hop of the CH (identity: 2, corresponding to the CH h_2 in Fig. 4) is the node whose identity is 17 (corresponding to the sink node h_5 in Fig. 4); the allele of the gene position 3 in *scheme* is 4, which indicates that the next hop of the CH (identity: 3, corresponding to the CH h_3 in Fig. 4) is the node whose identity is 4 (corresponding to the CH h_4 in Fig. 4). The gene positions in the second part indicate the identities of the CMs in the network. The allele of a gene position is the identity of the CH of the corresponding CMs. For example, the allele of gene position 7 in *scheme* is 2, which indicates that the CM (identity: 7, corresponding to the CM m_3 in Fig. 4) is assigned to the CH whose identity is 2 (corresponding to the CH h_2 in Fig. 4); the allele of the gene position 12 in *scheme* is 1, which indicates that the CM (identity: 12, corresponding to the CM m_8 in Fig. 4) is assigned to the CH whose identity is 1 (corresponding to the CH h_1 in Fig. 4).
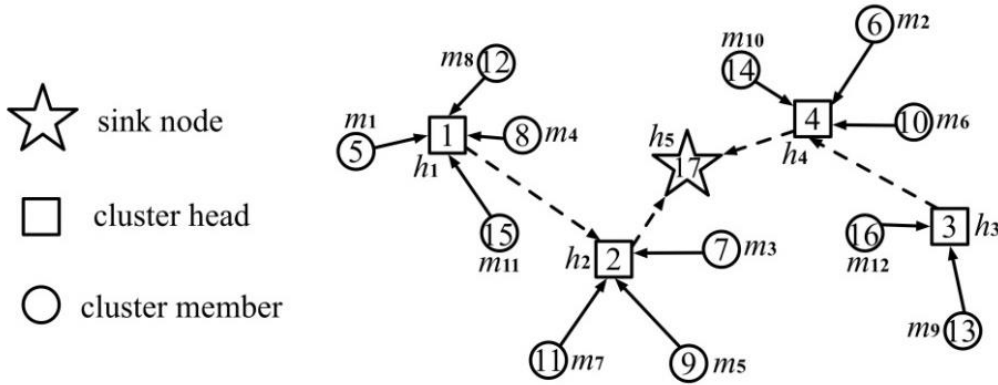

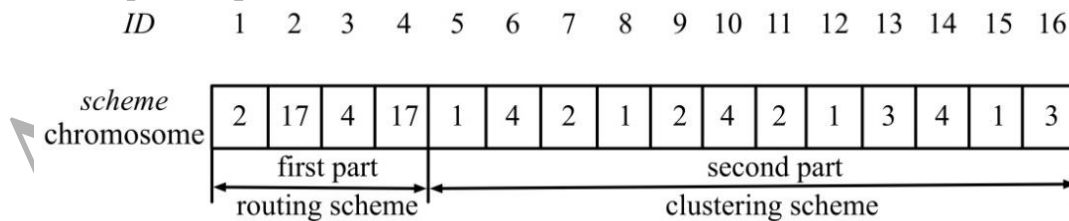Fig. 4 Example of a WSN clustering and routing scheme

Fig. 5 The *scheme* chromosome

5.2 Population Initialization

The initial population comprises multiple randomly generated *scheme* chromosomes. However, due to the limited communication range of the sensor nodes, the validity of each *scheme* chromosome should be guaranteed. For the first part of the chromosome, all the genes must satisfy:

$$scheme[k] \in nextHops(h_k), If\ k \le nH\ , \tag{25}$$

where the value of the k-th gene scheme must belong to the set of potential next hop nodes for the k-th CH h_k. In addition, if the k-th gene is located in the second part of the chromosome, then its value must belong to the set of the potential CHs for CM m_{k-nH}:

$$scheme[k] \in pCH(m_{k-nH}), If \ nH < k < n. \tag{26}$$

Algorithm1 Population initialization

1: **Procedure** *Scheme* = Init_Population(*previous_scheme*, *flag*, *H*, *M*, *nP*)
2:     *Scheme* = [];
3:     n = length(*H*) + length(*M*) + 1;
4:     nH = length(*H*);
5:     **for** i = 1 **to** *nP* **do**
6:         **if** i == 1 && *flag* == 1 **do**
7:             *Scheme*(1,:) = *previous_scheme*;
8:         **else**
9:             **for** j = 1 **to** n-1 **do**
10:                 **if** $j <= nH$ **do**
11:                     $nextHop\_h_j$ = rand($nextHops(h_j)$);
12:                     $Scheme(i, j)$ = $nextHop\_h_j$;
13:                 **else**
14:                     CH_m_{j-nH} = rand($pCH(m_{j-nH})$);
15:                     $Scheme(i, j)$ = CH_m_{j-nH};
16:                 **end if**
17:             **end for**
18:         **end if**
19:     **end for**
20: **end Procedure**

Fig. 6 Population initialization algorithm

In this paper, the network time is divided into multiple network rounds. In each network round, the optimal clustering and routing scheme is calculated through the proposed GECR method. The optimal scheme is different for each network round with the change of the energy of the sensor nodes. Eq. 18 shows that the optimization objective is related to the energy consumption, which is determined by the distance between the nodes. The geographical locations of the sensor nodes in the network are fixed, so the optimal clustering and routing scheme for the r+1-th network round is related to the optimal scheme for the r-th network round. Therefore, in order to reduce the number of iterations in the GA, the optimal scheme for the r-th network round is added to the initial population for the r+1-th network round. Algorithm1 in Fig. 6 illustrates the process followed to initialize the population. In Algorithm1, the optimal scheme obtained in the previous round is denoted as *previous_scheme*. To note there are two cases that *previous_scheme* should not be added into the initial population: (1) the network is in the first round; and (2) one or more failure nodes exist in the previous network round. In Algorithm1, we use the variable *flag* (*flga=0*) to represent these two cases. *H* denotes the set of all the CH nodes except the sink node and *M* denotes the set of all the CM nodes. *H* and *M* are preset before the network operation. The number of the chromosomes in the population is denoted by *nP*. The output of the algorithm is the initial population *Scheme*, which comprises of all the *scheme* chromosomes. In Algorithm1, line 2 initializes the population *Scheme* as an empty set, and lines 3 and 4 calculate the number of all nodes and the number of all CH nodes, respectively. Next, from line 5, the algorithm assigns the *nP* chromosomes. If the assigned object is the first chromosome and the value of *flag* is 1 (line 6), then the first chromosome in the population is set to *previous_scheme*

(line 7); otherwise, from line 8, the algorithm assigns the j-th gene of i-th chromosome in the population. In the first part of the chromosome (the routing scheme part), a node from the set $nextHops(h_j)$ is selected randomly as the next hop for CH h_j, and $nextHop\_h_j$ is assigned to $Scheme(i, j)$ (lines 10–12). In the second part of the chromosome (the clustering scheme part), a node is selected randomly from the set $p\,\square\square(m_{j-nH})$ as the CH for CM m_{j-nH}, and CH_m_{j-nH} is assigned to $Scheme(i, j)$ (lines 13–15).

5.3 Fitness Function

In WSNs, all of the sensor nodes have limited energy supply and each node needs to consume some energy when sending and receiving data from other nodes. If the total amount of energy consumption is reduced, the energy can be saved. Thus, we need to reduce the total amount of energy consumption for all of the nodes. In this paper, the proposed GECR method is applied for the WSNs with Permanent CHs. In this type of WSNs, the CHs are responsible for receiving and transmitting the data from its CMs and previous hops. Load balancing for CHs in the network is another important factor that can affect the network life cycle. If a CH consumes energy too rapidly because of its excessive load, the node will deplete prematurely, and the stability and energy efficiency of the network will both be affected. Thus, we also need to balance the energy consumed by each CH. Different clustering and routing schemes can lead to different amount of energy consumption and different load balancing of CHs. Therefore, in this paper the fitness of a single chromosome in the population is related to the total energy consumption and load balancing of the CHs.

The total energy consumption under the corresponding clustering and routing scheme is denoted by $sumE$. Eq. 18 shows that $sumE$ depends on mhE_{ij} and $\grave{E}_{h_k h_{nH+1}}$. Eq. 9 shows that mhE_{ij} is the sum of $sendmhE_{ij}$ and $recmhE_{ij}$. If the length of the transmitted data is l-bit, then we can calculate $sendmhE_{ij}$ from Eq. 3:

$$sendmhE_{ij}(l) = \begin{cases} E_{elec} * l + \varepsilon_{fs} * l * d^2(m_i, h_j), & If\ d(m_i, h_j) < d_0, \\ E_{elec} * l + \varepsilon_{mp} * l * d^4(m_i, h_j), & If\ d(m_i, h_j) \geq d_0. \end{cases} \tag{27}$$

Using Eq. 6, we can obtain the energy consumed by h_j to receive l-bit data from m_i:

$$recmhE_{ij} = E_{elec} * l. \tag{28}$$

Then, we combine Eq. 27 and Eq. 28 to calculate mhE_{ij}:

$$mhE_{ij}(l) = \begin{cases} 2E_{elec} * l + \varepsilon_{fs} * l * d^2(m_i, h_j), & If\ d(m_i, h_j) < d_0, \\ 2E_{elec} * l + \varepsilon_{mp} * l * d^4(m_i, h_j), & If\ d(m_i, h_j) \geq d_0. \end{cases} \tag{29}$$

The energy consumed during transmitting data from the CH h_k to the sink h_{nH+1} through multi-hops is calculated as follows:

$$\grave{E}_{h_k h_{nH+1}} = \begin{cases} hhE_{k\ nH+1}, yyyyyyIf\ nextHop\_h_k = h_{nH+1}, \\ hhE_{ky} + \grave{E}_{h_y h_{nH+1}}, & If\ nextHop\_h_k = h_y. yyy \end{cases} \tag{30}$$

If the next hop of h_k is the sink node, then $\grave{E}_{h_k h_{nH+1}}$ equals the energy consumed by h_k when directly transmitting data to the sink node; otherwise, $\grave{E}_{h_k h_{nH+1}}$ equals the sum of the communication energy required between all CHs and the energy used by the previous CH to transmit data to the sink node.

The sink node h_{nH+1} has an adequate energy supply, so the energy consumed by h_{nH+1} to receive data is ignored. Therefore, the communication energy between h_k and h_{nH+1} only comprises the energy used by h_k to transmit data. If the length of the transmitted data is l'-bit, then the method used for calculating $hhE_{k\ nH+1}(l')$ is as follows:

$$hhE_{k\ nH+1}(l') = \begin{cases} E_{elec} * l' + \varepsilon_{fs} * l' * d^2(h_k, h_{nH+1}), & If\ d(h_k, h_{nH+1}) < d_0, \\ E_{elec} * l' + \varepsilon_{mp} * l' * d^4(h_k, h_{nH+1}), & If\ d(h_k, h_{nH+1}) \geq d_0. \end{cases} \tag{31}$$

Eq. 10 shows that the communication energy between h_k and its next hop h_y comprises the energy used by h_k to

transmit data ($sendhhE_{ky}$) and the energy used by h_y to receive data ($rechhE_{yk}$). Similarly, if the length of transmitted data is l'-bit, then $hhE_{ky}(l')$ is calculated as:

$$hhE_{ky}(l') = \begin{cases} 2E_{elec} * l' + \varepsilon_{fs} * l' * d^2(h_k, h_y), & If\ d(h_k, h_y) < d_0, \\ 2E_{elec} * l' + \varepsilon_{mp} * l' * d^4(h_k, h_y), & If\ d(h_k, h_y) \geq d_0. \end{cases} \tag{32}$$

The *scheme* chromosome for Fig. 5 is used as an example to calculate the total energy consumption under the corresponding clustering and routing scheme. We assume that the distances between all sensor nodes are less than d_0 and the length of the transmitted data from CM nodes to their corresponding CH node is l-bit. In addition, the length of the transmitted data from CHs to their next hops is l'-bit and the transmitted data can be calculated by aggregating the collected data from the CM nodes. According to Fig. 4, the number of CM nodes *nM* and the number of CH nodes *nH* are 12 and four, respectively. Using Eq. 18, we can obtain the total energy consumption $sumE(l, l')$:

$$sumE(l, l') = \sum_{i=1}^{12} \sum_{j=1}^{4} mhE_{ij}(l) * c_{ij} + \sum_{k=1}^{4} \grave{E}_{h_k h_{nH+1}}(l'). \tag{33}$$

The previous part $\sum_{i=1}^{12} \sum_{j=1}^{4} mhE_{ij}(l) * c_{ij}$ is the clustering energy (denoted by E_1), which comprises the energy consumed by all of the CMs when transmitting the collected data and the energy used by all of the CHs to receive the data. E_1 is related to the second part of the *scheme* chromosome in Fig. 5, and it can be calculated as

$$\begin{aligned} E_1 = {}& mhE_{11}(l) + mhE_{24}(l) + mhE_{32}(l) + mhE_{41}(l) + mhE_{52}(l) + mhE_{64}(l) \\ & + mhE_{72}(l) + mhE_{81}(l) + mhE_{93}(l) + mhE_{10\,4}(l) + mhE_{11\,1}(l) + mhE_{12\,3}(l) \\ = {}& 24lE_{elec} + lE_{fs}(d^2(m_1, h_1) + d^2(m_2, h_4) + d^2(m_3, h_2) + d^2(m_4, h_1) \\ & + d^2(m_5, h_2) + d^2(m_6, h_4) + d^2(m_7, h_2) + d^2(m_8, h_1) + d^2(m_9, h_3) \\ & + d^2(m_{10}, h_4) + d^2(m_{11}, h_1) + d^2(m_{12}, h_3)). \end{aligned} \tag{34}$$

The remainder of Eq. 33, $\sum_{k=1}^{4} \grave{E}_{h_k h_{nH+1}}(l')$, is the routing energy (denoted by E_2), which comprises the total communication energy used when transmitting data from all of the CHs to the sink node. E_2 is related to the first part of the *scheme* chromosome in Fig. 5, and it can be calculated as:

$$\begin{aligned} E_2 = {}& \grave{E}_{h_1 h_5}(l') + \grave{E}_{h_2 h_5}(l') + \grave{E}_{h_3 h_5}(l') + \grave{E}_{h_4 h_5}(l') \\ = {}& hhE_{12}(l') + hhE_{25}(l') + hhE_{25}(l') + hhE_{34}(l') + hhE_{45}(l') + hhE_{45}(l') \\ = {}& 8lE_{elec} + l'\varepsilon_{fs}(d^2(h_1, h_2) + 2d^2(h_2, h_5) + d^2(h_3, h_4) + 2d^2(h_4, h_5)). \end{aligned} \tag{35}$$

Due to the limited energy of the sensor nodes in the network, we need to minimize the energy consumed in each round to prolong the network life cycle. Thus, the chromosomes should be selected that consume lower amounts of energy. In fact, a chromosome with lower energy consumption usually has a smaller fitness value. Therefore, the fitness function is proportional to the total energy consumption $sumE$:

$$Fit \propto sumE. \tag{36}$$

Then we calculate the parameter that is related to the load balancing of the CHs. We define $Avrg_i$ as the average remaining energy assigned to each load on the CH h_i, which can be calculated as:

$$Avrg_i = \frac{E_{residual}(h_i)}{nL(h_i)}. \tag{37}$$

The average value μ of $Avrg_i$ for all CHs can then be calculated:

$$\mu = \frac{\sum_{i=1}^{nH} Avrg_i}{nH}. \tag{38}$$

Based on the value of $Avrg_i$ and μ, we can obtain the standard deviation of the average residual energy $Avrg_i$:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{nH}(\mu - Avrg_i)^2}{nH}}. \tag{39}$$

A smaller value for the standard deviation σ will balance the energy consumption and increase the network life cycle. Therefore, σ is proportional to the fitness value:

$$Fit \propto \sigma. \tag{40}$$

By combining Eq. 36 and Eq. 40, we can obtain the following fitness function:

$$Fit \propto sumE + \sigma. \tag{41}$$

$sumE$ and σ are not in the same range, so these two values should be normalized as follows:

$$sumE = \frac{sumE - sumE_{min}}{sumE_{max} - sumE_{min}}, \tag{42}$$

$$\sigma = \frac{\sigma - \sigma_{min}}{\sigma_{max} - \sigma_{min}}. \tag{43}$$

$sumE_{max}$ and $sumE_{min}$ are the maximum and minimum of $sumE$, respectively, and σ_{max} and σ_{min} are the maximum and minimum of σ. The two parameters $sumE$ and σ affect the fitness to different degrees, so we add a weight λ to the fitness function:

$$Fit \propto \lambda * sumE + (1 - \lambda) * \sigma. \tag{44}$$

5.4 Crossover and Mutation

The design of the genetic operators is a critical step in GA. The genetic operators include the crossover and mutation operators. After entering a new generation, the chromosomes with lower fitness are selected using the roulette wheel algorithm. These selected chromosomes are the parent chromosomes for the crossover operator and they are used to produce new children chromosomes.
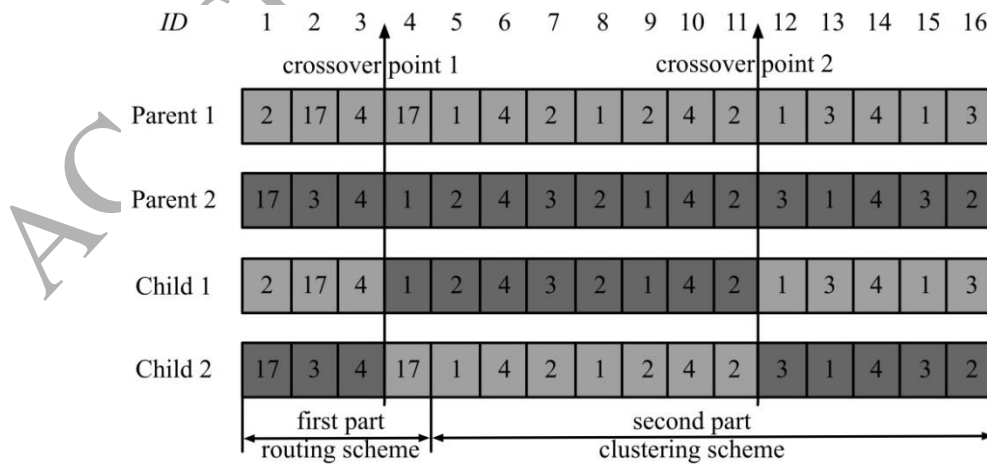

Fig. 7 Crossover between two parent chromosomes

In the proposed GECR algorithm, there is only one chromosome *scheme* for the individuals in the population. Fig. 5 shows that the chromosome contains two parts, which are used to generate the optimal routing scheme and the

optimal clustering scheme. In order to produce better genes in the two parts at the same time, we employ two-point crossover and randomly select a crossover point in the first part and the second part of *scheme*. Fig. 7 illustrates the crossover between two parent chromosomes. In this study, the crossover operator always takes place. As introduced in Section 4.1, the algorithm selects the higher performance chromosomes from the parent chromosomes and the new children chromosomes. Thus, the best solutions can be preserved for the next generation in this manner.

It should be noted that the child chromosomes produced by the crossover operation are still valid. In the first part of *scheme*, Eq. 25 shows that the *k*-th gene is randomly selected from the set $nextHops(h_k)$. After crossover, the *k*-th gene still belongs to $nextHops(h_k)$. Similarly, if the *k*-th gene belongs to the second part of *scheme*, Eq. 26 shows that the *k*-th gene belongs to the set $pCH(m_{k-nH})$.

Mutation is used to produce better chromosomes in the GA. In the proposed GECR method, we use the basic bit mutation operator. Each gene in the chromosome might mutate. We assume that the *k*-th gene needs to mutate. If the *k*-th gene is in the first part of the chromosome, then we need to randomly select a new allele from the set $nextHops(h_k)$ to replace the previous *k*-th allele. If the *k*-th gene is located in the second part of the chromosome, then a new allele needs to be randomly selected from the set $pCH(m_{k-nH})$ to replace the previous allele.

Fig. 8 illustrates the mutation of the chromosome *scheme*. In this figure, the chromosome after mutation is *scheme'*. There are two genes (the 3rd gene and the 15th gene) that need to mutate. The 3rd gene is located in the first part of the chromosome, the set $nextHops(h_3) = \{2,4,17\}$. Then the number 17 is chosen to replace the previous allele 4. Meanwhile, the 15th gene is located in the second part, the set $pCH(m_{11}) = \{1,2\}$. Then the number 2 is chosen to replace the previous allele 1.
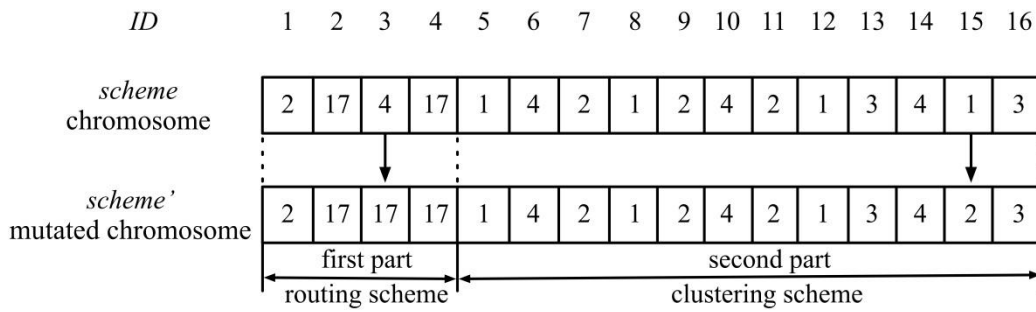

Fig. 8 Mutation of the chromosome *scheme*

6. Experimental Results

In experiments, we compare the proposed GECR algorithm with five clustering and routing algorithms: GACR [14], GAR [15], ASLPR [16], LEACH-GA [17] and GALBCA [20]. These algorithms are all based on GA.

6.1 Parameter Settings

MATLAB R2014a and the C programming language were used to simulate the clustering algorithms. MATLAB R2014a was operated on a system with an Intel CPU Core i7-3770, 16 GB RAM, and Windows 7. We used the same energy model and the corresponding typical parameters that are employed in most of the current routing algorithms for WSNs (e.g., [6; 14; 16; 28]). Table 1 illustrates the simulation parameters where E_{elec}, ε_{fs}, ε_{mp}, and E_{DA} are energy consumption-related parameters. Control packet size is the length of some notification messages. Message packet size is the length of the data messages collected by the sensor nodes. There are two sensing fields: Area1 and Area2. The smaller Area1 is deployed with 100 nodes and the larger Area2 has 200 sensor nodes. The sink node is in the centre of the network area and its coordinates in Area1 and Area2 are (50, 50) and (100, 100), respectively. d_{max} denotes the maximum communication range of the sensor nodes. The nodes are divided into two categories: CMs and CHs. The initial energy of the ordinary CMs is 0.1 J. The CHs are used as gateways, so they require more initial energy. And the setting of proportion and initial energy for CHs could be different [14; 15; 20]. To evaluate the adaptability to different

areas and different setting for CHs by the algorithms, we tested four simulation scenarios: WSN#1, WSN#2, WSN#3, and WSN#4, which were set as: WSN#1 = (Area1, 20 CHs, 0.6 J/CH), WSN#2 = (Area1, 40 CHs, 0.3 J/CH), WSN#3 = (Area3, 40 CHs, 0.6 J/CH), and WSN#4 = (Area1, 80 CHs, 0.3J/CH). To ensure that the comparisons were fair, all of the algorithms (GAR, GACR, GALBCA, LEACH-GA, ASLPR, and the proposed GECR) used the same parameter settings. According to the reference [52], in the proposed GECR method the mutation probability is set as $1/(n\text{-}1)$, where $(n\text{-}1)$ is the length of the chromosome.

Table 1 Experimental parameter settings

Parameter	Value
E_{elec}	50 (nJ/bit)
ε_{fs}	10 (pJ/bit/m^2)
ε_{mp}	0.0013(pJ/bit/m^4)
E_{DA} (Energy for data aggregation)	5 (nJ/bit)
Control packet size	200 (bits)
Message packet size	4000 (bits)
Area1 ($L*L$)	100*100 (m^2) and 100 nodes
Area2 ($L*L$)	200*200 (m^2) and 200 nodes
Location of sink node	Centre of the network area
d_{max}	L/2
Initial energy of CMs	0.1 J
Proportion of CHs	20%/40%
Initial energy of CHs	0.6 J/0.3 J

6.2 Comparison of the Weight λ

The GECR algorithm used a GA to determine the clustering and routing scheme for WSNs. The fitness of an individual in the population depends on the energy consumption and the variance in the loads on the CHs under the corresponding clustering and routing scheme. Eq. 44 shows that the weight λ determines the ratio of the energy consumption and the variance in the load. Thus, the value of λ influences the calculation of the optimal clustering and routing scheme. We selected an appropriate value of λ that obtained the best results in comparative experiments. The main objective of the GECR algorithm is to extend the network life cycle, so we selected a value of λ with a longer network life cycle. To define the life cycle of WSNs, we use three metrics [53-55]: first node death (FND), half nodes alive (HNA), and last node death (LND). The three metrics are fit for different scenarios. In some scenarios (intrusion or fire detection) the network quality decreases considerably as soon as one node dies, thus, all nodes should stay alive as long as possible. In these cases, it is important to know the value of FND. Meanwhile, in some scenarios, the sensors can be placed in proximity to each other and adjacent sensors could record related or identical data, thus, the loss of a single or few nodes does not automatically diminish the quality of service of the network. In these cases, HNA and LND are effective in the analysis the network life cycle.

In order to reduce the running time of the algorithm, we decreased the initial energy levels of the sensor nodes in the experiments when determining the value of λ: WSN#1 and WSN#3 (0.12 J/CH; 0.02 J/CM), and WSN#2 and WSN#4 (0.06 J/CHs; 0.02 J/CM). Table 2 illustrates the network life cycles for different values of λ in various scenarios. Fig. 9 shows the number of living nodes with different values of λ in various scenarios. The top three values are highlighted in bold in each row of Table 2. A larger number of bold values indicates the better performance for a specific value of λ. Table 2 shows that λ = 0.8 and λ = 0.9 obtained the most bold values with 10. In Fig. 9, the curve for λ = 0.8 is shown in black and the curve for λ = 0.9 is shown in brown. For WSN#4, the black curve and brown curve are adjacent to each other. For WSN#2, the black curve is above the brown curve at most time points, but

the brown curve is slightly better within a short time period close to 300 rounds. For WSN#1 and WSN#3, the black curve is significantly above the brown curve. Therefore, $\lambda = 0.8$ with black curve is optimal. In subsequent experiments, the value of λ was set as 0.8.

Table 2 Comparison of the network life cycle using different values of λ in various scenarios

Scenario	Life cycle	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	$\lambda = 0.4$	$\lambda = 0.5$	$\lambda = 0.6$	$\lambda = 0.7$	$\lambda = 0.8$	$\lambda = 0.9$	$\lambda = 1$
WSN#1	FND	36	**72**	**72**	**73**	**73**	**73**	**73**	**74**	**74**	62	35
	HNA	45	80	80	80	81	81	81	83	**84**	**86**	**86**
	LND	47	96	196	234	239	252	**264**	261	254	**267**	**463**
WSN#2	FND	31	70	73	**74**	73	**75**	**75**	**75**	**76**	**76**	56
	HNA	59	**83**	**83**	**83**	82	82	81	**83**	**83**	**84**	**87**
	LND	62	108	161	190	218	248	261	**299**	**299**	**313**	**437**
WSN#3	FND	20	43	45	46	46	**47**	**47**	**51**	**51**	50	46
	HNA	33	57	56	57	58	58	61	**63**	**64**	**65**	63
	LND	36	69	133	161	**188**	**192**	**214**	164	161	170	203
WSN#4	FND	17	31	33	45	43	45	46	**49**	**51**	51	**49**
	HNA	56	59	59	60	60	60	60	**61**	**61**	64	67
	LND	118	229	262	286	312	328	349	365	**385**	**396**	**445**

(a) WSN#1

(b) WSN#2

(c) WSN#3
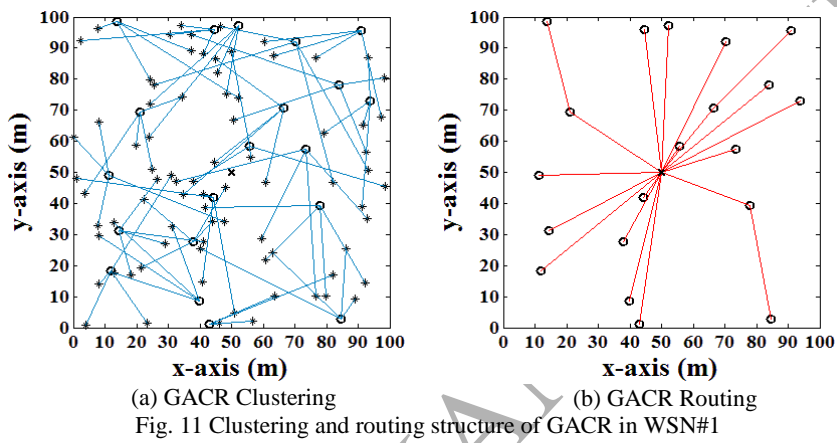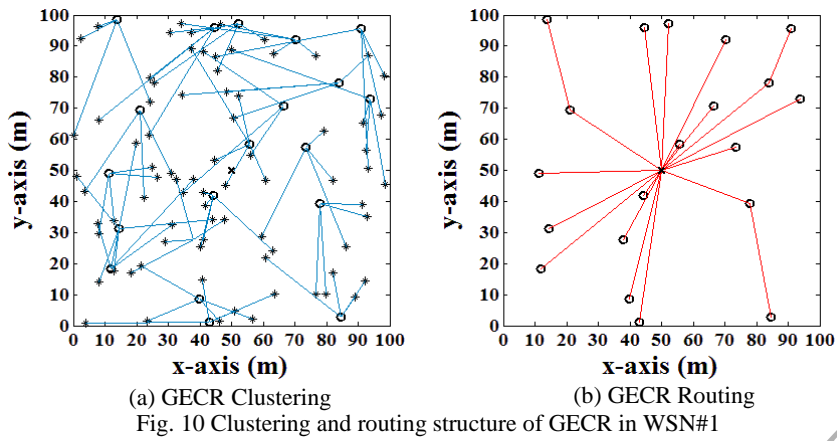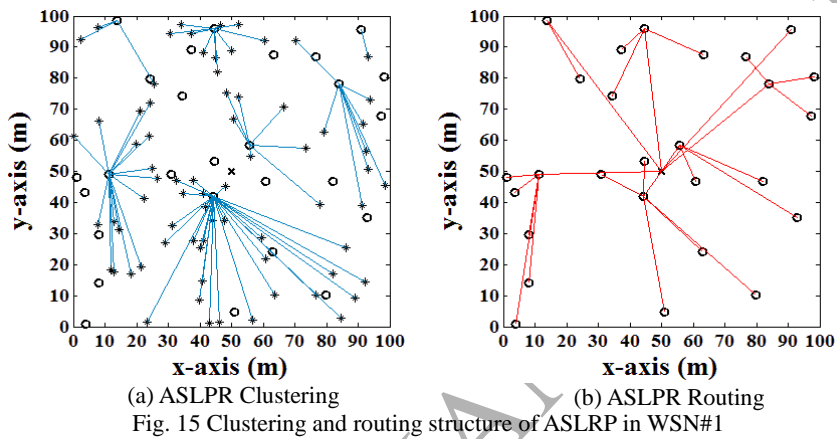
(d) WSN#4
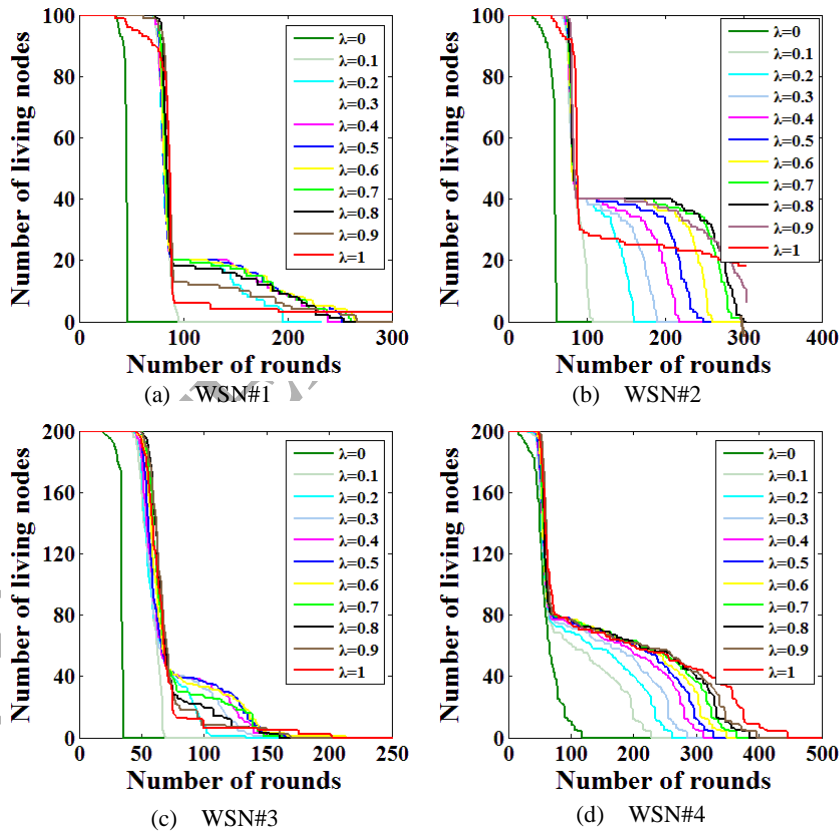
Fig. 9 Comparisons in terms of the number of living nodes using different values of λ in various scenarios

6.3 Clustering and Routing Structure

Fig. 10-15 compares the proposed GECR method with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the clustering and routing structure under the scenario WSN#1. In these figures, the symbol 'o' denotes the CH node, the symbol '*' denotes the CM node, and the symbol 'x' denotes the sink node. The blue line denotes the communication between the CMs and their corresponding CHs. The red line denotes the routing between the CHs.

(a) GECR Clustering　　　　　　　(b) GECR Routing
Fig. 10 Clustering and routing structure of GECR in WSN#1

(a) GACR Clustering　　　　　　　(b) GACR Routing
Fig. 11 Clustering and routing structure of GACR in WSN#1

(a) GALBCA Clustering　　　　　　(b) GALBCA Routing
Fig. 12 Clustering and routing structure of GALBCA in WSN#1

(a) GAR Clustering　　　　　　　　(b) GAR Routing
Fig. 13 Clustering and routing structure of GAR in WSN#1

(a) LEACH_GA Clustering (b) LEACH_GA Routing
Fig. 14 Clustering and routing structure of LEACH_GA in WSN#1

(a) ASLPR Clustering (b) ASLPR Routing
Fig. 15 Clustering and routing structure of ASLRP in WSN#1

From Fig. 10-15, we can draw the following conclusions. Firstly, for GECR in Fig. 10, the CH nodes have almost the same number of loads and the CMs are close to their corresponding CHs. Thus, the CH nodes can consume energy evenly and the amount of consumed energy for CHs and CMs are small. In addition, the CH nodes transmit data to their close CHs without greatly increasing the burden of the next hop nodes. Secondly, for GACR and GALBCA in Fig. 11 and Fig. 12, the clustering distributions of the two methods are more rambling than GECR. For GALBCA, the distances between some CMs and CHs are even longer than half of the edge length. Meanwhile, the two methods calculate the optimal routing schedule separately and do not consider the load balancing of routing. Thus, some CHs need to consume a large amount of extra energy to receive and transmit data from their previous hop CHs. In this way, the CHs cannot consume energy evenly. Thirdly, for GAR, LEACH-GA and ASLPR methods in Fig. 13, Fig. 14 and Fig. 15, there exist many single CHs, which do no connect with any CMs. Thus, the data aggregation function of the single CHs has been underused. In addition, these three methods have the load imbalance problem, which leads to the uneven energy consumption. For the routing result of GAR in Fig. 13(b), the data from the most CHs are transmitted multiple hops before reaching the sink node. Too many times of data transmission leads to the increase of drop rate and energy consumption. For the routing results of LEACH-GA and ASLPR in Fig. 14(b) and Fig. 15(b), some CHs send their data to their next hops, which are farther away from the sink node. In this circumstance, the energy of these CHs will be wasted. In conclusion, the proposed method has more appropriate clustering and routing structure.

6.4 Load Balancing

Load balancing determines the uniformity of energy consumption and communication load on CH nodes. Fig. 16 compares the proposed GECR method with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the variance in the number of loads on CHs under different scenarios (WSN#1, WSN#2, WSN#3, and WSN#4). In order to make the comparison clearer, we only show the variance in the loads in the algorithms over 20–80 rounds. It should be noted that the energy consumption by CHs comprises the energy used for receiving and forwarding the messages in previous hops as well as the energy required for receiving message from CMs. In this study, the load on a CH included

all of the sensor nodes within the cluster as well as the previous hops for the CH. Fig. 16 shows clearly that the variances in the loads were far greater using GAR, LEACH-GA, and ASLPR compared with the other clustering algorithms in all the scenarios. Thus, load balancing is not sufficient in GAR, LEACH-GA, and ASLPR. In WSN#1, the variances in the loads were similar for GECR and GACR, and they were both less than that using GALBCA. In WSN#2, WSN#3, and WSN#4, the variances in the loads were smallest using the proposed GECR method, and the fluctuations were relatively stable. The average variances using GECR in WSN#2, WSN#3, and WSN#4 were 0.43, 0.29, and 1.45 smaller, respectively, than those with the second most optimal algorithm. In all the scenarios, the variances with GECR were 20–30 times lower than those when using LEACH-GA and ASLPR. Therefore, load balancing is optimal with the proposed GECR algorithm.
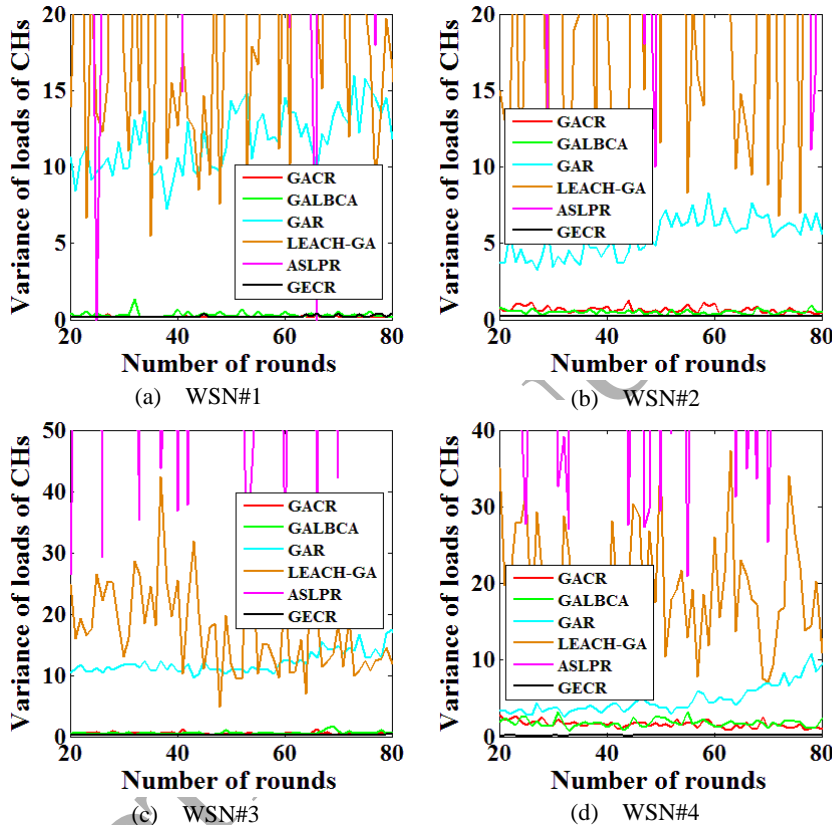

Fig. 16 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of load balancing

6.5 Network Life Cycle

Fig. 17 compares the proposed GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the number of living nodes under different scenarios (WSN#1, WSN#2, WSN#3, and WSN#4). Fig. 17 clearly shows that the failure time and rate of sensor nodes were earlier and faster, respectively, when using GAR, LEACH-GA, and ASLPR. In particular, when using LEACH-GA and ASLPR, some nodes were still alive but there were no living nodes with the other four algorithms. This difference is mainly because the nodes with more initial energy do not always act as CHs in LEACH-GA and ASLPR. Moreover, these two algorithms do not consider the residual energy of sensor nodes. Thus, the last living nodes were the nodes with higher initial nodes. Fig. 17(a) shows that the failure rate for sensor nodes was slowest with GACR within the period of time when the nodes began to die. However, when more than half of the nodes were dead, the failure rate with GACR was significantly faster than that using the proposed GECR. Fig. 17(c) shows that the number of living nodes was largest when using GALBCA within the short period of time before all the nodes dead. However, before this stage, the failure rate with GALBCA was much faster than that when using GACR and GECR. In WSN#1 the proposed GECR had the largest number of living nodes after about the 310th round, excluding LEACH-GA and ASLPR. In WSN#2 and WSN#3, the failure rate of sensor nodes was slowest most of the

time when using GECR. In addition, in WSN#4, the failure time and rate of sensor nodes were the latest and slowest, respectively, when using GECR.
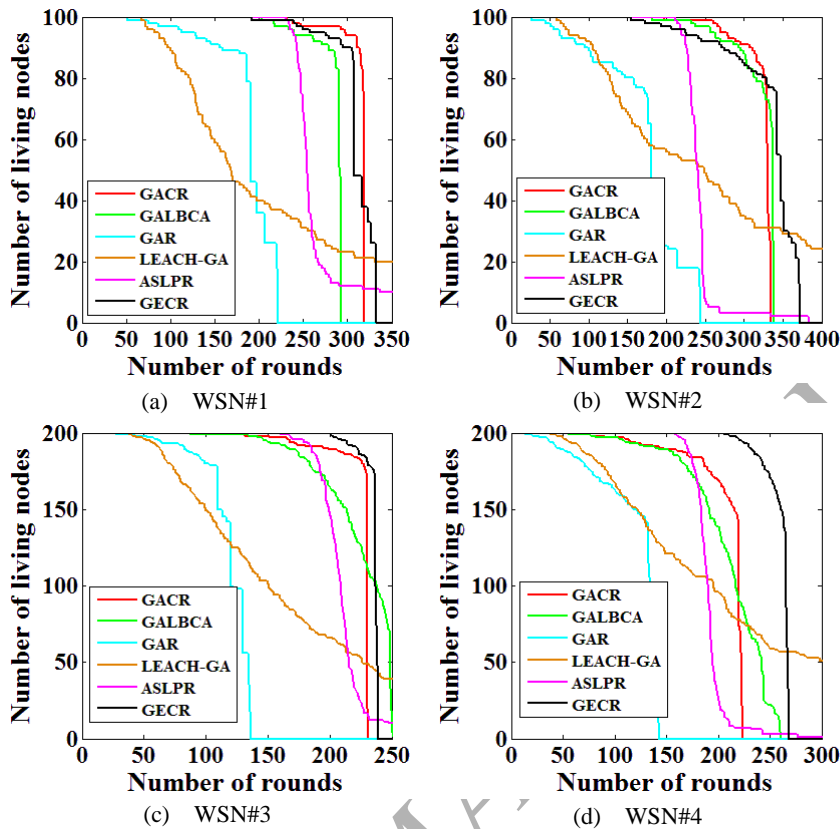

(a) WSN#1

(b) WSN#2

(c) WSN#3

(d) WSN#4

Fig. 17 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the number of living nodes

Table 3 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the network life cycle

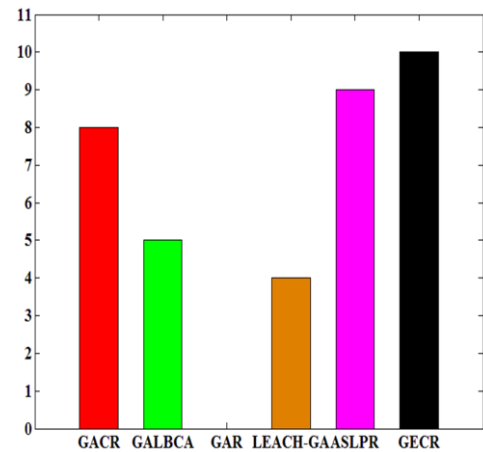| Scenario	Life cycle	GACR	GALBCA	GAR	LEACH-GA	ASLPR	GECR
WSN#1	FND	**232**	**203**	51	67	**231**	192
	HNA	**318**	291	190	168	**253**	**306**
	LND	319	293	221	**>350**	**>350**	333
WSN#2	FND	**197**	**182**	26	59	**212**	155
	HNA	**330**	**336**	180	248	240	**345**
	LND	335	339	244	**>380**	**>380**	372
WSN#3	FND	**94**	88	28	39	**161**	**201**
	HNA	**229**	**237**	121	150	208	**237**
	LND	231	252	136	**>260**	**>260**	239
WSN#4	FND	**57**	56	14	38	**159**	**206**
	HNA	**219**	**216**	133	195	190	**265**
	LND	224	261	143	**>270**	**>270**	**268**

Fig. 18 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the network life cycle

Table 3 illustrates the network life cycle (FND, HNA, and LND) when using the proposed GECR method, GACR, GALBCA, GAR, LEACH-GA, and ASLPR in different scenarios. To compare this metric among the different algorithms, the top three values are highlighted in bold in each row of Table 3. The optimal HNA values with GECR were always the best results, where they were 2.7% and 21% longer than those using the second most optimal algorithm in WSN#2 and WSN#4, respectively. To further compare the network life cycle, the count for each algorithm is shown in bold in Fig. 18, which demonstrates that the frequency of the network life cycle was the longest among the different

clustering algorithms. The frequencies for GACR, GALBCA, GAR, LEACH-GA, ASLPR, and GECR were 8, 5, 0, 4, 9, and 10, respectively. Therefore, our proposed GECR method has advantages in terms of the network life cycle compared with the other clustering algorithms.

6.6 Energy Consumption

In the proposed method, the CH nodes that act as gateways are supplied with more energy than the ordinary nodes (CM nodes). However, the energy supply is still limited for the CHs. In the network operation process, the CHs need to consume more energy than the ordinary nodes during each round. Therefore, reducing the energy consumed by CH nodes is an important evaluation criterion for clustering algorithms. In LEACH-GA and ASLPR, the numbers of CHs are not fixed in different rounds. Thus, we compared the average energy consumed by the CHs in each round. Fig. 19 compares the proposed GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the energy consumed by CHs under different scenarios (WSN#1, WSN#2, WSN#3, and WSN#4). Clearly, the energy consumed by the CHs was the lowest using GECR in all of the scenarios. The average energy consumption levels in the four scenarios were 0.8%, 9.4%, 4%, and 19% lower when using GECR compared with the second most optimal algorithm. Therefore, the proposed GECR method has advantages in terms of improving the energy efficiency of the CH nodes.
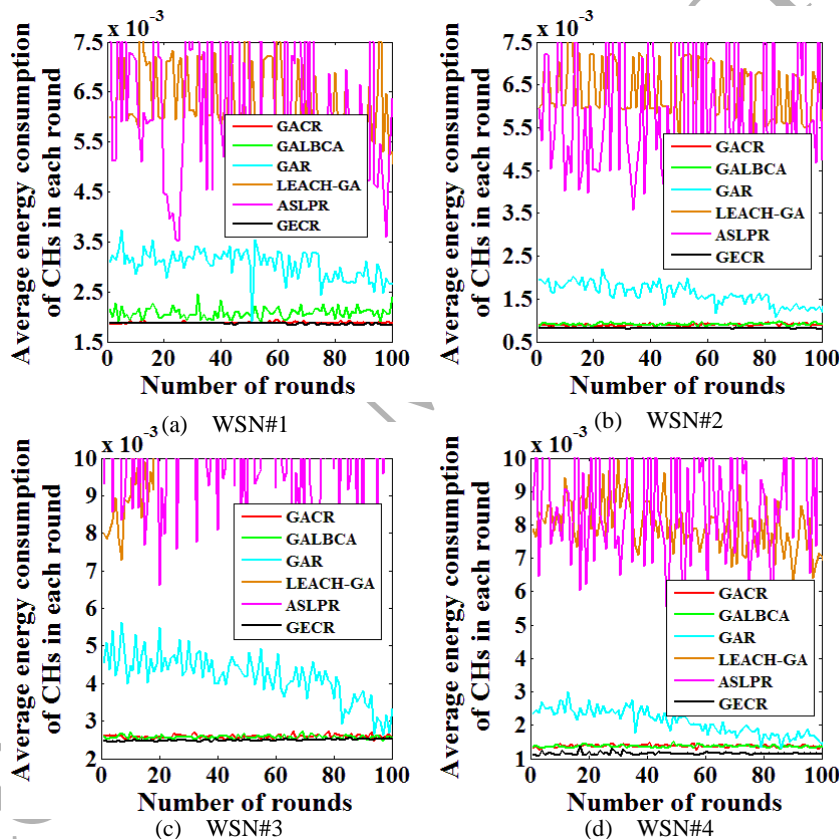

Fig. 19 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of total energy consumed by CHs

Fig. 20 compares GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the energy consumed by all nodes during different rounds in four scenarios (WSN#1, WSN#2, WSN#3, and WSN#4). Clearly, the energy consumed in each round was highest when using GAR and ASLPR, followed by LEACH-GA, GALBCA, and GACR, whereas the proposed GECR method had the lowest energy consumption. Fig. 21 compares GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the total energy consumed by all nodes in all of the previous rounds under different scenarios. In all of the scenarios, the curve obtained for the proposed GECR was always below the curves for the other algorithms. Therefore, compared with the other clustering algorithms, the proposed GECR algorithm had the lowest energy consumption, i.e., 2.6%, 5.6%, 9%, and 14% lower than that by the second best algorithm in WSN#1, WSN#2, WSN#3, and WSN#4, respectively, and the average energy consumed by GECR in the four scenarios was 28.7%, 40.6%, 27.8%, and 35.8% lower than that by the worst algorithm. Thus, the GECR algorithm
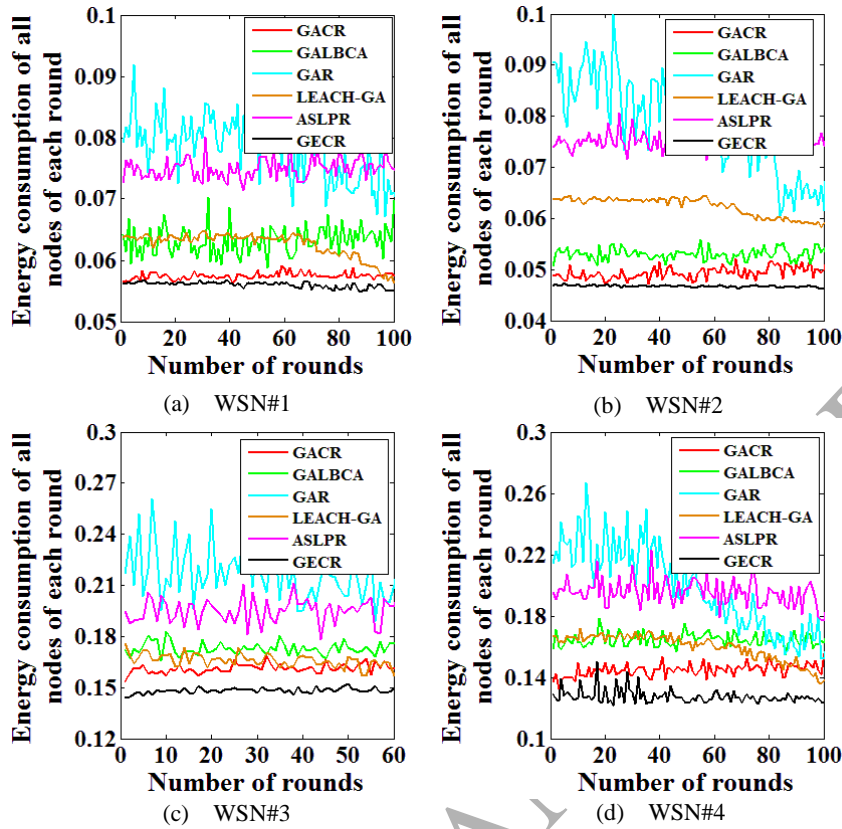
has the highest energy efficiency.

(a)   WSN#1

(b)   WSN#2

(c)   WSN#3

(d)   WSN#4

Fig. 20 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the energy consumed by all nodes in different rounds

(a)   WSN#1
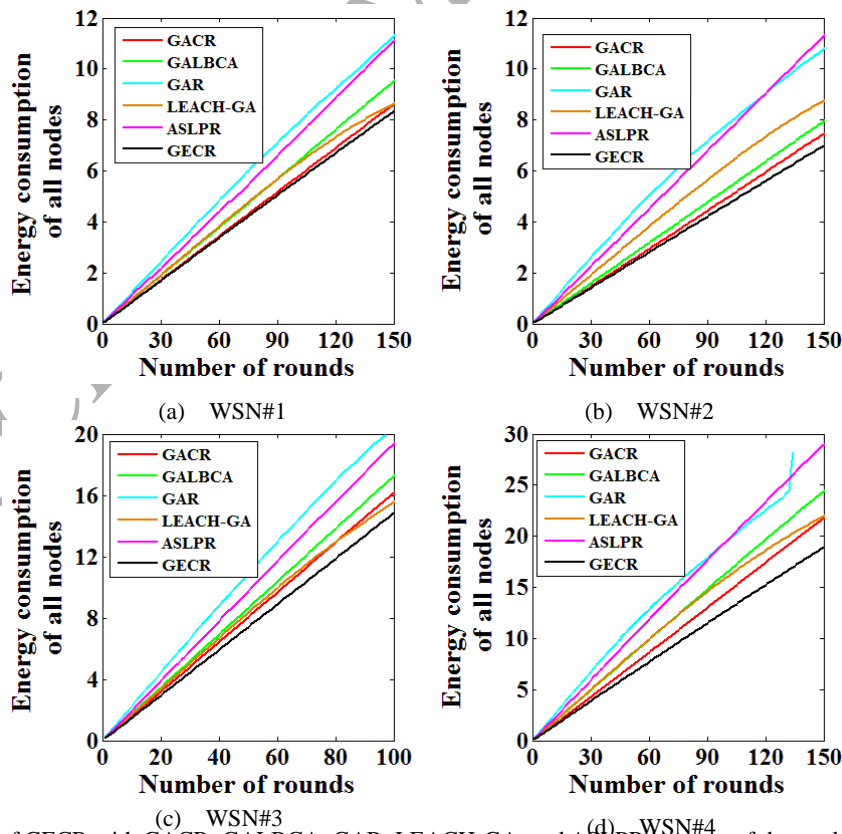
(b)   WSN#2

(c)   WSN#3

(d)   WSN#4

Fig. 21 Comparisons of GECR with GACR, GALBCA, GAR, LEACH-GA, and ASLPR in terms of the total energy consumed by all nodes in all previous rounds

6.7   Experimental Results in OMNeT++

To validate the experimental results in the MATLAB simulator, we use OMNeT++ simulator to compare the proposed method and the GACR method, which is the second optimal method in the MATLAB simulator. OMNeT++ is an open source object-oriented modular discrete event network simulator consisting of hierarchically nested modules. It occupies a very important position in the field of network simulation and it has been widely adopted in the area of WSNs [56-59]. Fig. 22 illustrates the comparisons of GECR with GACR in terms of the total energy consumed by all nodes in different rounds by OMNeT++. From Fig. 22, we can see that the proposed GECR method had the lower energy consumption than the compared GACR method. Therefore, the experimental results in OMNeT++ simulator and the MATLAB simulator have good consistency.

(a)   WSN#1

(b)   WSN#2
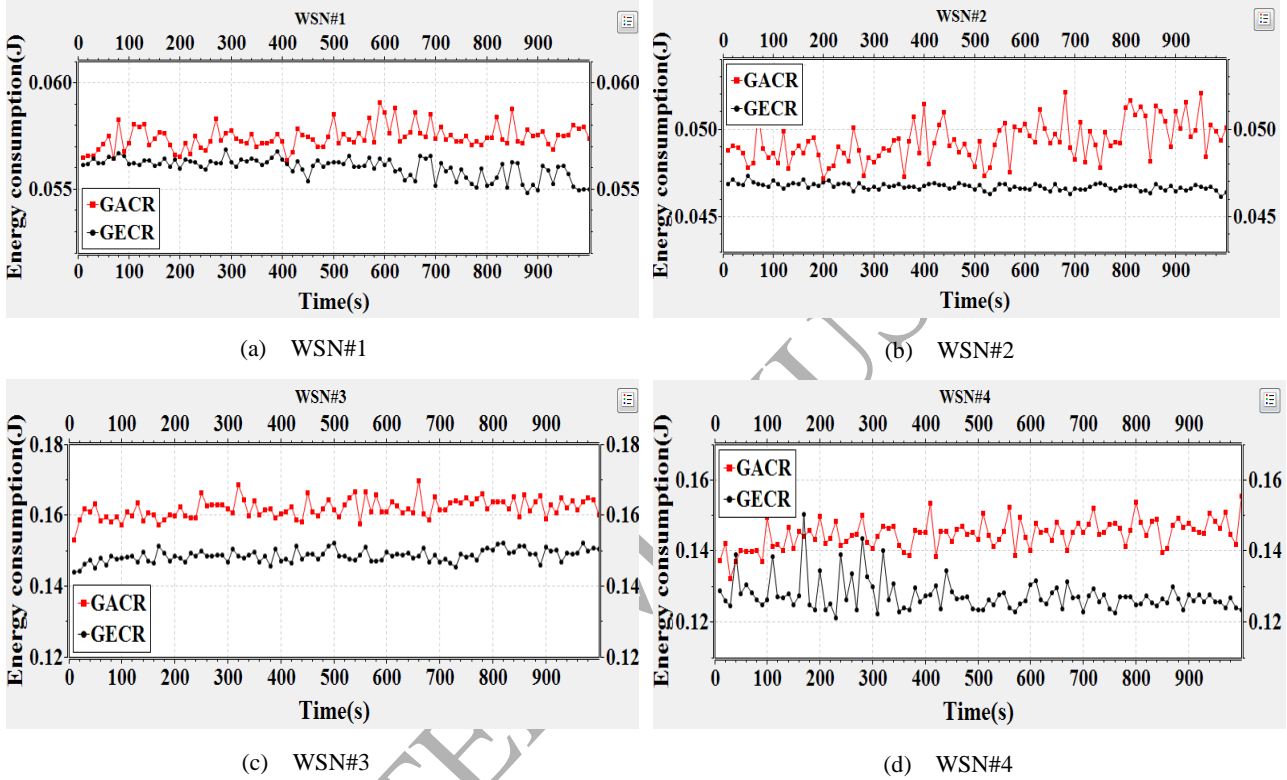
(c)   WSN#3

(d)   WSN#4

Fig. 22 Comparisons of GECR with GACR in terms of the total energy consumed by all nodes in different rounds by OMNeT++

7.   Conclusions

In this study, we proposed a GECR algorithm to calculate globally the total energy consumed by all sensor nodes where the algorithm encodes the clustering scheme and routing scheme together in the same chromosome. GECR treats the total energy consumed by all nodes as a parameter in the fitness function. In cluster-based WSNs, CHs need to transmit the data from the previous hop nodes in addition to the data from the CMs. In the proposed method, we add the previous hops to the loads on each CH, which makes the calculation of the load more accurate. Simulation experiments indicated that the performance of our proposed algorithm was better than that of other GA-based clustering and routing algorithms (GAR, GACR, GALBCA, LEACH-GA, and ASLPR) in terms of load balancing, the network life cycle, and energy consumption. With respect to load balancing, the proposed GECR always had the lowest variances in the loads on the CHs under different scenarios, where the loads were 20–30 times lower than those when using LEACH-GA and ASLPR. In terms of the network life cycle, GECR had the most living nodes at most times. Meanwhile, the HNA values were always among the optimal results with GECR. In addition, GECR consumed the smallest amount of energy in all of the network rounds. The average amounts of energy consumed by the CHs when using GECR in the four scenarios were 0.8%, 9.4%, 4%, and 19% lower, respectively, than those by the second most optimal algorithm, while the average amounts of energy consumed by all nodes using GECR in the four scenarios were 28.7%, 40.6%, 27.8%, and 35.8%

lower than those by the worst algorithm. In summary, GECR performed well in terms of load balancing, network life cycle and energy consumption. In practical applications, sensor nodes are usually energy-constrained, and thus, GECR would be more efficient for WSNs with permanent CHs.

Many metaheuristic algorithms have been developed and applied successfully in WSNs. In future research, we will focus on testing and applying suitable metaheuristic algorithms for clustering and routing in WSNs with permanent CHs.

References

[1] Leu J S, Chiang T H, Yu M C, et al. Energy Efficient Clustering Scheme for Prolonging the Lifetime of Wireless Sensor Network With Isolated Nodes[J]. IEEE Communications Letters, 2015, 19(2): 259-262.

[2] Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless sensor networks: a survey[J]. Computer Networks, 2002, 38(4): 393-422.

[3] Ali N A, Drieberg M, Sebastian P. Deployment of MICAz mote for Wireless Sensor Network applications[C]. IEEE International Conference on Computer Applications and Industrial Electronics, 2012: 303-308.

[4] Abbasi A A, Younis M. A survey on clustering algorithms for wireless sensor networks[J]. Computer communications, 2007, 30(14): 2826-2841.

[5] Singh S P, Bhanot K, Sharma S. Critical Analysis of Clustering Algorithms for Wireless Sensor Networks[M]. Springer Singapore, 2016.

[6] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks[C]//System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on. IEEE, 2000: 10 pp. vol. 2.

[7] Mohapatra N P, Behera S K. Relay Node and Cluster Head Placement in Wireless Sensor Networks[J]. International Proceedings of Computer Science & Information Tech, 2012.

[8] Wang T, Zhang G, Yang X, et al. A Trusted and Energy Efficient Approach for Cluster-Based Wireless Sensor Networks[J]. International Journal of Distributed Sensor Networks,2016,(2016-4-10), 2016, 2016: 1-13.

[9] Yang X S. Nature-Inspired Metaheuristic Algorithms[M]. Luniver Press, 2008.

[10] Goldberg D E. Genetic Algorithm in Search, Optimization, and Machine Learning[J], 1989, xiii(7): 2104–2116.

[11] Elhoseny M, Yuan X, Yu Z, et al. Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks Using Genetic Algorithm[J]. IEEE Communications Letters, 2015, 19(12): 2194-2197.

[12] Lai C C, Ting C K, Ko R S. An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications[C]. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007: 3531-3538.

[13] Rao P C S, Banka H. Energy efficient clustering algorithms for wireless sensor networks: novel chemical reaction optimization approach[J]. Wireless Networks, 2015, 23: 1-20.

[14] Gupta S K, Jana P K. Energy Efficient Clustering and Routing Algorithms for Wireless Sensor Networks: GA Based Approach[J]. Wireless Personal Communications, 2015, 83(3): 1-21.

[15] Gupta S K, Kuila P, Jana P K. GAR: An Energy Efficient GA-Based Routing for Wireless Sensor Networks[C]. International Conference on Distributed Computing and Internet Technologies, 2013: 267-277.

[16] Shokouhifar M, Jalali A. A new evolutionary based application specific routing protocol for clustered wireless sensor networks[J]. AEU - International Journal of Electronics and Communications, 2015, 69(1): 432-441.

[17] Liu J L, Ravishankar C V. LEACH-GA: Genetic Algorithm-Based Energy-Efficient Adaptive Clustering Protocolfor Wireless Sensor Networks[J], 2011, 1(1): 79-85.

[18] Peiravi A, Mashhadi H R, Javadi S H. An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm[J]. International Journal of Communication Systems, 2013, 26(1): 114–126.

[19] Liao Y, Qi H, Li W. Load-Balanced Clustering Algorithm With Distributed Self-Organization for Wireless Sensor

Networks[J]. IEEE Sensors Journal, 2013, 13(5): 1498-1506.

[20] Kuila P, Gupta S K, Jana P K. A novel evolutionary approach for load balanced clustering problem for wireless sensor networks[J]. Swarm & Evolutionary Computation, 2013, 12: 48-56.

[21] Kumar V, Jain S, Tiwari S. Energy Efficient Clustering Algorithms in Wireless Sensor Networks: A Survey[J]. International Journal of Computer Science Issues, 2011, 8(5).

[22] Al-Karaki J N, Kamal A E. Routing techniques in wireless sensor networks: a survey[J]. IEEE Wireless Communications, 2004, 11(6): 6-28.

[23] Raja B, Rajakumar R, Dhavachelvan P, et al. A survey on classification of network structure routing protocols in wireless sensor networks[C]. IEEE International Conference on Computational Intelligence and Computing Research, 2017: 1-5.

[24] Boyinbode O, Le H, Mbogho A, et al. A Survey on Clustering Algorithms for Wireless Sensor Networks[C]. International Conference on Network-Based Information Systems, 2010: 2826-2841.

[25] Rostami A S, Badkoobe M, Mohanna F, et al. Survey on Clustering in Heterogeneous and Homogeneous Wireless Sensor Networks[J]. Journal of Supercomputing, 2017.

[26] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Transactions on Wireless Communications, 2000, 1(4): 660-670.

[27] Younis O, Fahmy S. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks[J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379.

[28] Batra P K, Kant K. LEACH-MAC: a new cluster head selection algorithm for Wireless Sensor Networks[J]. Wireless Networks, 2016, 22(1): 1-12.

[29] Sohn I, Lee J H, Sang H L. Low-Energy Adaptive Clustering Hierarchy Using Affinity Propagation for Wireless Sensor Networks[J]. IEEE Communications Letters, 2016, 20(3): 558-561.

[30] Lindsey S, Raghavendra C S. PEGASIS: Power-efficient gathering in sensor information systems[C]. Aerospace conference proceedings, 2002. IEEE, 2002: 3-3.

[31] Manjeshwar A, Agrawal D P. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks[C]. null, 2001: 30189a.

[32] Li Q, Zhu Q, Wang M. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks[J]. Computer Communications, 2006, 29(12): 2230-2237.

[33] Singh S, Chand S, Kumar B. Energy Efficient Clustering Protocol Using Fuzzy Logic for Heterogeneous WSNs[J]. Wireless Personal Communications, 2016, 86(2): 1-25.

[34] Elbhiri B, Saadane R, Fldhi S E, et al. Developed Distributed Energy-Efficient Clustering (DDEEC) for heterogeneous wireless sensor networks[C]. International Symposium on I/v Communications and Mobile Network, 2010: 1-4.

[35] Rajeswari K, Neduncheliyan S. Genetic algorithm based fault tolerant clustering in wireless sensor network[J]. Iet Communications, 2017, 11(12): 1927-1932.

[36] Gupta G, Younis M. Load-balanced clustering of wireless sensor networks[C]. IEEE International Conference on Communications, 2003: 1848-1852 vol.3.

[37] Kuila P, Jana P K. Energy Efficient Load-Balanced Clustering Algorithm for Wireless Sensor Networks [J]. Procedia Technology, 2012, 6(4): 771-777.

[38] Low C P, Fang C, Ng J M, et al. Efficient Load-Balanced Clustering Algorithms for wireless sensor networks [J]. Computer Communications, 2008, 31(4): 750-759.

[39] Mohajerani A, Gharavian D. An ant colony optimization based routing algorithm for extending network lifetime in wireless sensor networks[J]. Wireless Networks, 2016: 1-11.

[40] Singh A, Behal S. Ant Colony Optimization For Improving Network Lifetime In Wireless Sensor Networks[J]. Research Cell An International Journal of Engineering Sciences, 2013, 8(June 2013): 1-12.

[41] Rajeswari K, Neduncheliyan S. Genetic Algorithm based Fault Tolerant Clustering in Wireless Sensor Network[J]. Iet Communications, 2017.

[42] Rejinaparvin J, Vasanthanayaki C. Particle Swarm Optimization-Based Clustering by Preventing Residual Nodes in Wireless Sensor Networks[J]. IEEE Sensors Journal, 2015, 15(8): 4264-4274.

[43] Elhabyan R S Y, Yagoub M C E. Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network[J]. Journal of Network & Computer Applications, 2015, 52: 116-128.

[44] Kennedy J, Eberhart R. Particle swarm optimization[C]. IEEE International Conference on Neural Networks, 1995. Proceedings, 2002: 1942-1948 vol.4.

[45] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-66.

[46] Kuila P, Jana P K. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach[J]. Engineering Applications of Artificial Intelligence, 2014, 33(1): 127-140.

[47] Jamatia A, Chakma K, Kar N, et al. Performance Analysis of Hierarchical and Flat Network Routing Protocols in Wireless Sensor Network Using Ns-2[J]. International Journal of Modeling and Optimization, 2015, 5(1): 40.

[48] Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing[R]. 2003.

[49] Intanagonwiwat C, Govindan R, Estrin D, et al. Directed diffusion for wireless sensor networking[J]. IEEE/ACM Transactions on Networking (ToN), 2003, 11(1): 2-16.

[50] Haas Z, Halpern J Y, Li L. Gossip-based ad hoc routing[J], 2006, 14(3): 479-491.

[51] Chiti F, Fantacci R, Mastandrea R, et al. A distributed clustering scheme with self nomination: proposal and application to critical monitoring[J]. Wireless Networks, 2015, 21(1): 329-345.

[52] Zhu Z, Zhang G, Li M, et al. Evolutionary Multi-Objective Workflow Scheduling in Cloud[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(5): 1344-1357.

[53] Gherbi C, Aliouat Z, Benmohammed M. An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks[J]. Energy, 2016, 114: 647-662.

[54] Handy M J, Haase M, Timmermann D. Low energy adaptive clustering hierarchy with deterministic cluster-head selection[C]. International Workshop on Mobile and Wireless Communications Network, 2002: 368-372.

[55] Raghuvanshi A S, Tiwari S, Tripathi R, et al. Optimal number of clusters in wireless sensor networks: a FCM approach[J]. International Journal of Sensor Networks, 2011, 12(1).

[56] Faye S, Chaudet C. Characterizing the Topology of an Urban Wireless Sensor Network for Road Traffic Management[J]. IEEE Transactions on Vehicular Technology, 2016, 65(7): 5720-5725.

[57] Zhang J, Chen J, Fan J, et al. OMNeT++ based Simulation for Topology Control in Wireless Sensor Network: A Case Study[C]. Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International, 2008: 1130-1134.

[58] Li W, Zhu C, Zhu C, et al. Scheduling and routing methods for cognitive radio sensor networks in regular topology[J]. Wireless Communications & Mobile Computing, 2016, 16(1): 47-58.

[59] Kucuk K, Kavak A, Yigit H. A Smart Antenna Module Using OMNeT++ for Wireless Sensor Network Simulation[C]. International Symposium on Wireless Communication Systems, 2007: 747-751.

Tianshu Wang, was born in Jiangsu Province, China, in 1989. She received her Ph.D. candidate of School of Computer Science and Engineering in Nanjing University of Science and Technology in China. She is currently a lecturer of School of Information Technology in Nanjing University of Chinese Medicine. Her main research interests include embedded system, wireless sensor networks, clustering and routing, and trusted computing.

Gongxuan Zhang, was born in Jiangxi Province, China, in 1961. He received his Ph.D. in School of Computer Science and Engineering from Nanjing University of Science and Technology, Nanjing, China. He is currently a professor in the School of Computer Science and Engineering at the Nanjing University of Science and Technology. His research interests include cloud computing, trusted computing, wireless sensor networks.

Xichen Yang, was born in Jiangsu Province, China, in 1989. He received his Ph.D. in School of Computer Science and Engineering in Nanjing University of Science and Technology in China. He is currently a lecturer of School of Computer Science and Technology in Nanjing Normal University in China. His main research interests include Image quality assessment, Image processing and computer vision.

Ahmadreza Vajdi, was born in Iran, in 1988. He is a Ph.D. candidate of School of Computer Science and Engineering in Nanjing University of Science and Technology in China. His main research interests include clustering and routing, wireless sensor networks.