

Received August 16, 2020, accepted August 25, 2020, date of publication August 28, 2020, date of current version September 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020158

Energy-Efficient and Load-Balanced Clustering Routing Protocol for Wireless Sensor Networks Using a Chaotic Genetic Algorithm

CHUHANG WANG¹, XIAOLI LIU¹, HUANGSHUI HU², YOUJIA HAN^{1,3}, AND MEIQIN YAO³

¹College of Computer Science and Technology, Changchun Normal University, Changchun 130032, China

²College of Computer Science and Engineering, Jilin University of Architecture and Technology, Changchun 130114, China

³College of Computer Science and Engineering, Jilin University of Technology, Changchun 130012, China

Corresponding author: Huangshui Hu (huhuangshui@jluat.edu.cn)

This work was supported in part by the Science and Technology Project of the Education Department of Jilin Province during the 13th Five-year Plan under Grant JJKH20181166KJ, in part by the Science and Technology Development Project of the Department of Science and Technology of Jilin Province under Grant 20190802005ZG, and in part by the Industrial Technology Research and Development Project of Jilin Province Development and Reform Commission under Grant 2019C054-4.

ABSTRACT In wireless sensor networks, organizing nodes into clusters, finding routing paths and maintaining the clusters are three critical factors that significantly impact the network lifetime. In this paper, using a chaotic genetic algorithm, a clustering routing protocol combined with these three features called CRCGA is proposed to improve the network energy efficiency and load balancing. In CRCGA, the chaotic genetic algorithm is used to select the best cluster heads (CHs) and to find the optimal routing paths by coding them into a single chromosome simultaneously. Chaotic genetic operators based on a novel fitness function considering minimum energy consumption and load balancing along with new determination conditions make the algorithm converge quickly. Besides, an adaptive round time considering energy and load balancing is presented to maintain the clusters so as to further reduce energy consumption. Simulation results indicate that CRCGA is better than LEACH, GECR, OMPFM and GADA-LEACH in terms of convergence speed, energy efficiency, load balancing, network throughput and lifetime.


INDEX TERMS WSNs, multi-hop routing, chaotic genetic algorithm, clustering, energy and load balancing.

I. INTRODUCTION

With the rapid development of information technology, wireless sensor networks (WSNs) are widely used not only in military, disaster prevention, space exploration but also in environmental monitoring, intelligent transportation and smart home by means of various sensors built in nodes [1], [2]. Energy saving that prolongs the network lifespan has always been the most important topic in WSNs because the nodes are equipped with constrained energy and limited capabilities. Clustering routing has been proved to be an efficient, robust and scalable energy saving method [2]–[5], in which clusters are constructed by organizing nodes into groups, and cluster heads (CHs) are selected to manage the clusters. Low-energy adaptive clustering hierarchy (LEACH) is the pioneering clustering routing protocol for WSNs [3] with some advantages such as less overhead by clustering and aggregation, equal chance to be CH by randomly selecting CHs, less collisions by applying TDMA

mechanism, and long network lifetime by rotating CHs in rounds. Meanwhile, some shortcomings are also present in LEACH such as selecting nodes with low residual energy as CHs, distributing clusters unevenly, unbalancing energy consumption and increasing overhead of control messages due to its random CH selection and fixed round time. So great efforts have been made to improve the performance of LEACH from different aspects for clustering routing methods and expected results have been achieved since then [6]–[10]. Clustering routing method usually contains three main phases: clustering, routing, maintaining clusters.

Clustering usually consists of CHs selection and clusters formation. Many approaches have been provided for CHs selection, which can be categorized into probability based [4]–[6], [11]–[14], weight based [7], [15]–[17] and heuristics based [9], [10], [18]–[20] approaches. The nodes are selected as CHs when their threshold values are less than a random assigned real number between 0 and 1 in probability based approaches. However, some nodes with small threshold values may still be selected as CHs, thus resulting in their premature death. Therefore, in order to

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir .

solve the problem, only nodes with large weight are directly selected as CHs in weight based approaches. But local decision as well as neglecting uncertainties and dynamics of the network, makes it hard to find the optimal solutions for CHs selection. Moreover, non-deterministic polynomial (NP) hard problem for clustering has not been solved efficiently in probability and weight based approaches [1]. Therefore, heuristics based approaches are used to obtain approximate solutions to NP hard problems for CHs selection [21]. As a kind of important optimization methods, heuristic algorithms can find the proper answers for clustering by utilizing local or global search. Consequently, fuzzy logic inference [9], [18], [22], particle swarm optimization algorithm [23], bat algorithm [19], genetic algorithm [10], [20], differential evolution and harmony search [21] have been used to find the optimal CHs. Once the CHs are selected, each of them broadcasts an advertisement message announcing its CH identity, and the normal nodes decide to join the clusters based on the received signal strength, residual energy of the CHs and so on in order to form uniform and energy efficient clusters [5], [11], [12], [15], [21]–[23]. Moreover, a TDMA schedule like in LEACH is used for intra-cluster communication to further reduce energy consumption [3]–[23].

Routing is used to find the optimal routing path for each CH because the CMs only can communicate with their corresponding CHs. When a CH receives all the data from its CMs, it fuses and forwards the data to the BS directly in single-hop mode [3]–[6], [19], [22], [23] or indirectly in multi-hop mode [7]–[18], [24]. Usually, communication between CHs and BS in single hop mode may expend excessive energy when the CH appears far away from the BS and undoubtedly reduce the scalability of the network. Thus, data forwarding in multi-hop mode gradually becomes a common way for routing by selecting proper relay nodes to the BS so as to reduce energy consumption [25]. Similar to cluster selection, weight-based [7], [11], [18] and heuristics based [8]–[10], [13], [16], [17], [20], [21], [26] approaches are used for finding routing paths. Weight based on residual energy of the next-hop CHs [7], distance to the next-hop CHs [18], and so on [11] is used to select the nodes with higher weight values as the next-hop CHs. Similarly, local decision as well as neglecting uncertainties and dynamics of the network makes it hard to find the optimal routing paths. Hence, fuzzy logic [9], particle swarm optimization [8], ant colony optimization [20], [26], genetic algorithm [10], [13], [17], [20] are utilized to find the optimal routing paths for each CH so as to achieve balanced energy consumption of CHs and increased network lifetime. Moreover, they can also alleviate hot spot problem caused by hop-by-hop routing to some extent as other unequal clustering routing methods by adjusting cluster size, selecting vice CHs, and controlling hop counts and so on [12], [15], [24], [27].

Maintaining clusters is used to rotate the CHs for even distribution of the energy consumption among the nodes. Usually, round is adopted to reconstruct clusters in the

network like in LEACH, but it is difficult to determine the most appropriate round time. Moreover, fixed round time inevitably leads to premature death of CHs with low residual energy, resulting in interruption of communication [5], [9], [11], [12], [15], [16], [24]. Especially in the later period of network operation, this situation is even worse. So variable round time is proposed to prevent depleting the CHs' remaining energy and keep the normal communication in the entire network [5], [9], [11], [24], or CH substitution is used to replace the CHs whose residual energy is lower than a preset threshold [12], [16] or hybrid [15]. In this way, the number of rotations is significantly decreased, and the same time energy consumption is saved and the network lifetime is extended accordingly.

In this paper, an improved multi-hop clustering routing protocol using a chaotic genetic algorithm (CRCGA) is proposed to minimize the network energy consumption and to balance the network load. To this end, a chaotic genetic algorithm is used to find the optimal CHs and routing paths simultaneously, in which chaotic selection, crossover and mutation operations are adopted to avoid local optimum and to improve the convergence speed. Real number coding is used to combine CHs selection and finding routing paths into a single chromosome so as to significantly decrease its size. Minimizing the network energy consumption and balancing the network load are considered as factors in a novel fitness function. Moreover, a new adaptive round time is presented to further reduce the energy consumption of the network. Simulation experiments are carried out to verify the effectiveness of CRCGA, compared with several existing relevant algorithms in terms of convergence speed, energy efficiency, load balancing, network throughput and lifetime.

The remainder of this paper is organized as follows. The related works are described in Section 2, and the system model is presented in Section 3. In Section 4, the proposed CRCGA is introduced in detail, and simulation results are given in Section 5. Finally, Section 6 concludes the paper.

II. RELATED WORKS

As mentioned above [1], [2], clustering routing protocols have been widely used to improve energy efficiency and to extend the network lifetime. Moreover, heuristics based methods are more likely to get the optimal solutions than traditional methods [8]–[10], [13], [17], [20]–[23] due to their capabilities of adaptivity on network dynamics and uncertainties, and excellent search ability. By simulating the evolution process of biological populations in nature, genetic algorithms with low computational complexity can not only directly operate on the objects, automatically obtain and guide the optimized search space, but also adaptively adjust the search direction, quickly converge, and find the best global solution in the end. So they have been widely used in the clustering routing methods from CHs selection [20], [28]–[30] to routing search [13], [17], [31], [32] or hybrid [10], [33]. Generally, population initialization, fitness function definition, genetic operators and determination

conditions are the main aspects which need to be considered in designing genetic algorithms. Consequently, the next introduction only focuses on the clustering routing protocols based on genetic algorithm that are related to CRCGA from these mentioned aspects.

Population initialization is used to randomly produce a set of individuals whose size depends on the specific applications [10], [13], [20] or the length of chromosomes [29]. Individuals are represented by chromosomes, and a chromosome represents a solution for CH selection [20], [28]–[30], [33] or routing finding [13], [17], [31]–[33] or hybrid [10]. Moreover, a unit in a chromosome is called a gene. Generally, binary coding and real number coding are used to produce chromosomes. Examples for chromosomes in different coding schemes are illustrated in Fig. 1.

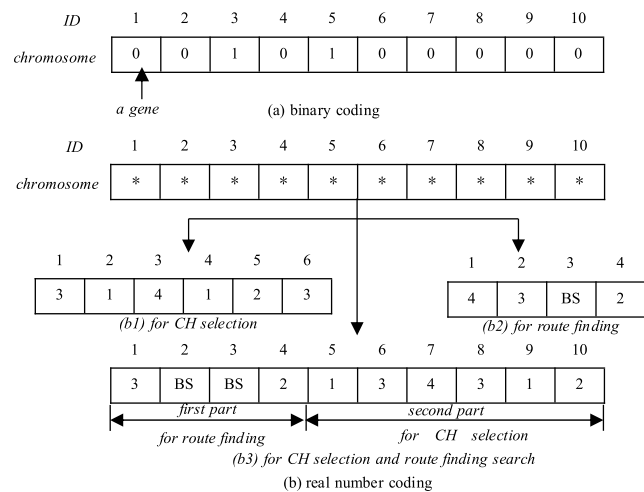


FIGURE 1. Chromosome representation.

As shown in Fig. 1(a), the authors in [28]–[30] used binary value ‘1’ located in a certain unit in the chromosome to indicate that the corresponding node denoted by its ID is a CH, while Alshalabi *et al.* [13] used it to indicate a next-hop CH. In contrast, ‘0’ means the corresponding node is a normal node or not included in the routing path. Accordingly, the CHs {3, 5} or routing path 3→5→BS can be obtained. Different from binary coding, real number coding uses a node’s ID in positive integer to indicate this node is the CH or the next-hop CH of the node located at the corresponding ID position, as shown in Fig. 1(b). Similarly, Gupta and Jana [33] represented a chromosome in real number coding, which is used for CH selection. At the same time, the authors in [6], [17], [32] used chromosomes in real number coding for routing search. Moreover, Wang *et al.* [10] represented a chromosome in real number coding for CH selection and routing search. For CH selection, normal nodes and CHs are represented by their ID sets respectively, a gene with CH’s ID indicates the CH of the normal node located at the corresponding ID position. As shown in Fig. 1(b1), the sets of CHs and normal nodes are denoted as {1, 2, 3, 4} and {1, 2, 3, 4, 5, 6} respectively, then the CH of each normal

node can be obtained, such as the CH of nodes 1 and 6 is CH 3, and so on. For routing search, normal nodes and CHs are represented by the same ID set, and a gene indicates the next-hop CH of the corresponding CH, as shown in Fig. 1(b2), all routing paths for each source CH can be found according to the chromosome, 1→4→2→3→BS, 2→3→BS, 3→BS, 4→2→3→BS. For representing CH selection and routing search simultaneously, a chromosome is divided into two parts, as shown in Fig. 1(b3), one is for CH selection, and the other is for routing search. Apparently, nodes 5, 6, 7, 8, 9, 10 are normal nodes, and their CHs are 1, 3, 4, 3, 1, 2 respectively. Besides, the routing paths for each CH are 1→3→BS, 2→BS, 3→BS, 4→2→BS respectively. It can be seen from Fig. 1 that the length of chromosomes may be different for the same size network, which usually equals to the total number of nodes in the network. However, Dehghani *et al.* [29] pointed out that the longer the chromosome length is, the slower the convergence speed may be. Therefore, the authors in [13], [29] used residual energy and other parameters to exclude the low performance nodes from participating in CH selection or routing search so as to reduce the length of chromosomes and improve the convergence speed. Furthermore, invalid chromosomes may be produced due to random gene generation. So the authors in [10], [17], [28], [32], [33] used communication range of nodes, ID sequence and other parameters to guarantee the validity of produced chromosomes. In this way, a certain number of valid chromosomes form the initial population for the subsequent genetic operations.

Fitness function is defined to evaluate the fitness values of individuals in the current population, in order to make the nodes with excellent performance pass on to the next generation. Accordingly, meeting the design objectives of the proposed protocols is the main criteria for defining fitness functions. First, in order to find the minimum number of CHs, the authors in [20], [28]–[30], [33] used residual energy of nodes, distances to other nodes and BS as main parameters to define different fitness functions. For instance, Kong *et al.* [28] used number of CHs, distance to the BS of CHs and distance to the BS of all nodes to define the fitness function which is expressed as follows:

$$f_m = \frac{\alpha * |S - C| - \beta * (\sum_{1 \leq i \leq C} d(C_i, S_{sink}))}{\sum_{1 \leq i \leq n} d(x_i, S_{sink})} \quad (1)$$

where m denotes chromosome m , S is the number of nodes, and C is the number of CHs. $d(C_i, S_{sink})$ and $d(x_i, S_{sink})$ denote the distance between a CH or a node to the BS respectively, and α, β are distance independent coefficients. However, nodes with low residual energy may be chosen as CHs, resulting in unbalanced energy consumption. So Dehghani *et al.* [29] added the residual energy of all nodes to define the fitness function, which is given in Eq. (2):

$$F(x) = \alpha * \frac{E_{Total} - E_{Dissipated}}{E_{Total}} + \beta * \frac{D_{NB} - D_N}{D_{NB}} + \gamma * \frac{N_N - N_C}{N_N} \quad (2)$$

where X denotes chromosome X , E_{Total} is the sum of initial energy of all nodes, $E_{Dissipated}$ is total energy consumption of CMs and CHs, DNB is the sum of distance between CMs to the BS, DN is the total distance between CMs to CHs, NN and NC are the number of sensor nodes and CHs respectively, α , β , γ are application-related coefficients indicating the importance of the variables. Then, nodes with more residual, shorter distance to the BS and smaller cluster centrality will be selected as CHs. Furthermore, the optimal number of CHs is obtained. Second, in order to find the best routing paths, the authors in [13], [17], [32], [33] used the hop count, residual energy of nodes, distances to other nodes and to the BS as main parameters to define the fitness functions. For instance, different from [13], [28]–[31], [33], Genta et al. [17] used the fitness function to evaluate the fitness values of genes instead of individuals, which is shown as follows:

$$f(v_i) = \frac{1}{Dist(v_{source}, v_i)} + \frac{1}{Dist(v_i, BS)} + E_{res}(v_i) \quad (3)$$

where v_i denotes gene i which means a possible next-hop CH in a certain chromosome, $Dist(v_{source}, v_i)$ is the distance between the source CH and v_i , and $Dist(v_i, BS)$ is the distance between v_i and the BS, $E_{res}(v_i)$ is the residual energy of v_i . The higher the fitness is, the more likely it is to be selected as the next-hop CH of the source CH. However, the optimal routing paths may not be found only by selecting the next-hop CHs. Thus, Alshalabi et al. [13] presented a fitness function to find the optimal paths, which is expressed as follows:

$$F(i) = D(CH_s, BS) + \frac{N_{par}}{N} + No_{ofpart} + members \quad (4)$$

where i represents the chromosome i , $D(CH_s, BS)$ is the average distance from the source CH and the BS through the intermediate CHs, N_{par} is the number of CHs through the path, N is the number of CHs in chromosome i , No_{ofpart} is the total number of participation in the transmission process of all CHs in the path, and $members$ is the total number of member nodes in all clusters that are related to the CHs in the path. The smaller the fitness, the more likely the path is to be chosen as the optimal path. However, local decision according to a certain path or a node or some nodes is impossible to obtain the global optimal solution, moreover, load balance is rarely considered in the above protocols. Last, in order to find the optimal CHs and routing paths simultaneously, Wang et al. [10] used residual energy of nodes and loads of nodes as parameters to define the fitness function which is expressed as follows:

$$Fit \propto \alpha * sumE + (1 - \alpha) * \sigma \quad (5)$$

where $sumE$ is the total energy consumption of all CMs and CHs, σ is the standard deviation of the average remaining energy assigned to each load on a CH, which is expressed as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{nH} \left(\frac{E_{residual}(h_i)}{nL(h_i)} - \frac{\sum_{i=1}^{nH} \frac{E_{residual}(h_i)}{nL(h_i)}}{nH} \right)^2}{nH}} \quad (6)$$

where nH is the number of CHs, $E_{residual}(h_i)$ is the residual energy of CH h_i , and $nL(h_i)$ is load on CH h_i . Moreover, $sumE$ and σ are normalized to make them stay within the same range. It is obviously seen from Eq. (6) that the optimal chromosome can minimize the network energy consumption, and the optimal CHs and routing paths are obtained simultaneously. However, distance between the nodes, if not considered in the fitness function, may lead to uneven cluster distribution.

Genetic operators including selection, crossover and mutation are utilized to produce a better next generation population than the previous one, based on the defined fitness function, so as to make the individuals be better from generation to generation. Selection operator is used to choose better individuals from the current population for crossover and mutation operators. Generally, roulette wheel selection is the most popular strategy used in clustering routing protocols [10], [13], [17], [20], [29], [31], [33], in which the chromosomes with better fitness values are chosen from the population based on probability. Besides, elitist selection is another strategy used to propagate some elite individuals to the next generation directly [32], and the other chromosomes in the population are passed on to crossover operator. Compared with roulette wheel selection, elitist selection can improve convergence speed as well as reduce the computational complexity by its direct reproduction of elite individuals to the next generation. For crossover operator, genes are swapped between two parent chromosomes based on crossover point so as to produce two child chromosomes. Then, two of the four chromosomes with better fitness values are selected for mutation operator. Single-point crossover is one of the most widely used crossover methods in clustering routing protocols by applying a random [13], [17], [31]–[33] or fixed [28] crossover point, in which the genes after the crossover point exchange between two parents, as shown in Fig. 2(a). Moreover, two-point crossover is also used for some clustering routing protocols such as GECC [10], in which swapping genes between two crossover points occurs between two parents, as shown in Fig. 2(b).

For mutation operator, bit mutation is the basic method for clustering routing protocols [10], [13], [28], [32], [33], in which a gene changes at the position of random determined mutation point, as shown in Fig. 3(a). Moreover, the mutation point also can be determined by considering the gene positions of multiple parents [13]. Besides, two genes can be switched according to the mutation point [31], as shown in Fig. 3(b).

Similar to chromosomes production, invalid chromosomes also may be generated by crossover and mutation operators due to random swap and switch [13], [20], [30], [31], then the validity of new produced chromosomes is also judged in [10], [28], [32], [33] so as to produce a valid next generation.

Determination conditions are applied to determine whether to stop genetic operators or not. Just like the authors in [10], [13], [20], [28]–[33], iteration number was used

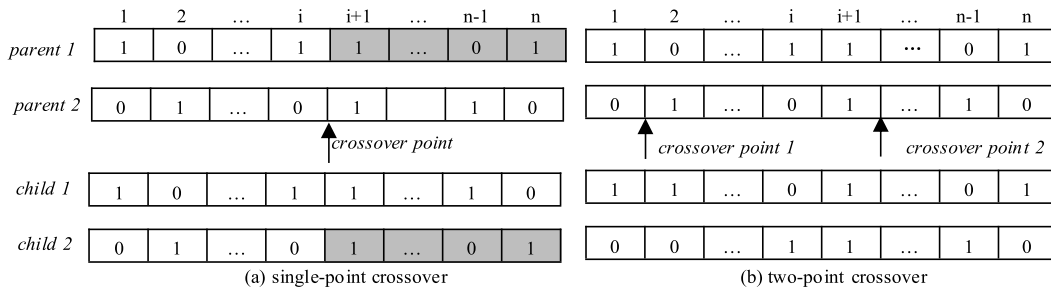


FIGURE 2. Crossover taking binary coding as an example.

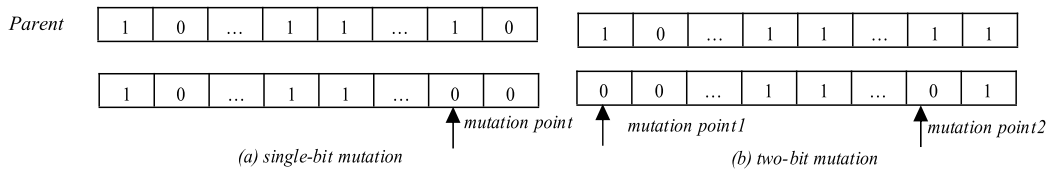


FIGURE 3. Mutation taking binary coding for an example.

as the determination condition for most clustering routing protocols. Once the iteration number reaches the threshold, genetic algorithm stops and the optimal solution with the best fitness value in the new generated population is obtained. Otherwise, the current population is replaced by the new generated population which is used for genetic operators, iteratively.

Simulation results showed that the above mentioned protocols can achieve their own design objectives by using genetic algorithms. However, the authors in [34], [35] indicated that the traditional genetic algorithm has some drawbacks such as premature convergence and local optimum. So inspired by chaos theory that studies dynamical systems, and whose present state depends on the characteristics of the previous state, in contrast to random systems where it is impossible to predict the next state from the previous one, chaotic optimization has been used to avoid local optimization, randomness and ergodicity in search processes [34], [36], [37]. Therefore, a chaotic genetic algorithm integrated chaotic optimization with traditional genetic algorithm is introduced in CRCGA to modify the stochastic parameters of genetic algorithm. And chaotic map is used to generate the initial population in order to avoid the premature convergence to a local solution. In this way, the global optimal solution of CHs selection and routing paths finding is obtained by utilizing chaotic selection, crossover and mutation operators. Consequently, the energy consumption is minimized and the network load is balanced. Moreover, a new adaptive round time is presented to further prolong the network lifetime.

III. SYSTEM MODEL

In CRCGA, n power constrained nodes with unique IDs are randomly deployed in the network whose area is $M \times Mm^2$, and the nodes are grouped into clusters. There is a CH and several CMs in a cluster, and the CH is selected to manage

the cluster including forming the cluster, receiving data from its CMs, aggregating and sending data directly or indirectly to the BS, maintaining the cluster and so on. A CM has direct access to only one CH, and a CH has direct access to only one BS or another CH at the same time. Moreover, for simplicity and comparability, the QoS (Quality of Service) of links such as reliability [38], [39] and security [40], [41] are not considered in this paper just like in [27]–[35]. That is to say a node can communicate with other nodes as long as it is within their communication radius. Besides, all nodes are supposed to be synchronized at the same moment like in [9]. Without loss of generality, the following assumptions are considered while utilizing the proposed network model.

- Once deployed, all nodes including the BS are unmovable.
- Homogeneous nodes are considered with the same sensing, processing, storage, communication and initial energy capabilities.
- The BS has unlimited capability in energy, processing, communication and so on. It can communicate with all the nodes in the network.
- The wireless link between any two nodes is symmetric, so that energy consumption is identical for the same amount of data transmission between the two nodes.
- The distance between any two nodes can be obtained by the received signal strength.

Based on the proposed network model, our main goal is to minimize the network energy consumption and balance the network load. To this end, the terminology used in this paper is listed below.

- $H = \{h_1, h_2, \dots, h_k\}$, $M = \{m_1, m_2, \dots, m_q\}$ denote the sets of CHs and CMs of the network respectively, and k is the number of CHs, q is the number of CMs, $k + q = n$. Moreover, the BS is denoted by h_{k+1} for

the convenience of the following calculation, and the number of CMs of h_i is denoted by $n_{CM_{h_i}}$.

- $N_i = \{n_1, n_2, \dots, n_p\}$ denotes the set of neighbors of node i , and p is the number of neighbors.
- E_{mh} denotes the energy consumption of intra-cluster communication of the network.
- E_{hh} denotes the energy consumption of inter-cluster communication of the network.
- E_{total} denotes the total energy consumption of the network, and $E_{total} = E_{mh} + E_{hh}$.
- d_{ij} denotes the distance between nodes i and j , and d_{max} is the maximum communication range for all the nodes in the network.
- $CH_{m_i} = \{h_j | \forall h_j \in H \wedge 0 < d_{m_i h_j} \leq d_{max}\}$ denotes the candidate CHs for m_i , and H_{m_i} is the selected CH for node m_i from CH_{m_i} .
- $CH_{h_i} = \{h_j | \forall h_j \in (H \cup h_{k+1}) \wedge d_{h_i h_{k+1}} > d_{h_j h_{k+1}} \wedge d_{h_i h_j} \leq d_{max}\}$ denotes the candidate next-hop CHs for h_i during its communication range, and H_{h_i} is the selected next-hop CH for h_i from CH_{h_i} . Moreover, nH_{h_i} is used to present the number that is selected as the next-hop CH for CH h_i .

$$E_{Tij} = \begin{cases} l * E_{elec} + l * \varepsilon_{fs} * d_{ij}^2, & d_{ij} < d_0 \\ l * E_{elec} + l * \varepsilon_{fs} * d_{ij}^4, & d_{ij} \geq d_0 \end{cases} \quad (7)$$

where E_{elec} denotes the energy consumption for transmitting or receiving 1-bit data, ε_{fs} and ε_{mp} denote the amplifier coefficients of free space and multi-path fading respectively, d_{ij} is the distance between nodes i and j , d_0 is the threshold distance given by $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$. In addition, the amount of energy consumption for receiving l - bits data is given by

$$E_{Rij} = l * E_{elec} \quad (8)$$

And the amount of energy consumption for aggregating l - bits data is given by

$$E_{DA} = l * E_{pDb} \quad (9)$$

where E_{pDb} is the energy consumption for 1-bit data fusion. Based on Eq. (7), (8), (9), the energy consumption for m_i and h_i can be obtained, which is expressed as follows:

$$E_{m_i} = E_{T_{m_i H_{m_i}}} \quad (10)$$

$$E_{h_i} = \sum_{j=1}^{n_{CM_{h_i}} + nH_{h_i}} E_{R_{j_i}} + E_{DA} + (1 + nH_{h_i}) * E_{T_{i H_{h_i}}} \quad (11)$$

Accordingly, the total amount of energy consumption in the network can be calculated as follows:

$$E_{total} = \sum_{i=1}^q E_{m_i} + \sum_{i=1}^k E_{h_i} \quad (12)$$

It can be seen from Eq. (12) that the uniform cluster distribution can also be obtained since CRCGA considers the minimum energy consumption of not only intra-cluster but also inter-cluster communication.

Moreover, the load on each CM is almost the same in CRCGA as the other clustering routing protocols. Then only the loads on CHs are used to measure the load balancing of the network, which is expressed as follows:

$$LB_{CHs} = \sqrt{\frac{\sum_{i=1}^k (Ld_{h_i} - avg_{CHs})^2}{k}} \quad (13)$$

where

$$Ld_{h_i} = (1 + nH_{h_i}) * Ld \quad (14)$$

$$avg_{CHs} = \frac{\sum_{i=1}^k Ld_{h_i}}{k} \quad (15)$$

which mean the load of CH h_i and the average load of CHs respectively. Therefore, the objective of CRCGA is to minimize the values of E_{total} and LB_{CHs} .

IV. THE PROPOSED CRCGA

In CRCGA, an improved chaotic genetic algorithm is used to find the optimal CHs and routing paths simultaneously. Real number coding instead of binary coding is adopted to represent the chromosomes, and a novel fitness function is designed to determine the quality of chromosomes. Genetic operators are conducted to produce new chromosomes. In this way, the optimal CHs are selected to form clusters and the optimal routing paths are found to transfer data. Moreover, adaptive round time based re-clustering is used to rotate the CHs, with the routing paths updated, accordingly. The detail realization of CRCGA is described as follows.

A. ENCODING CHROMOSOME

A chromosome represents a possible solution for the optimal CHs and routing paths, which is coded as a sequence of positive integers in CRCGA, whose length equals the number of nodes n in the network, and a positive integer represents the ID number of a node. Moreover, a chromosome consists of two parts, one for routing paths with k genes is called routing genes, and the other is for CHs selection with $n - k$ genes called CH genes accordingly. A routing gene denotes the next-hop CH of the selected CH, and a CH gene denotes the CH of the corresponding CM. Then a chromosome with an example is depicted in Fig. 4.

The ID of all nodes in the network is listed in the first row from 1 to n , the ID of randomly selected CHs is listed in the front part of the second row along with the ID of CMs. The chromosome is represented in the third row which contains n genes. A modified chaotic logistic map is adopted to generate genes in CRCGA because of its sensitivity to initial values, better random sequence generation, traversing all state point in chaotic region and long term unpredictability, which is expressed as follows:

$$g_i = \left\lfloor \frac{b-1}{1-0} * \mu * z_i * (1-z_i) \right\rfloor, \quad 0 < z_i < 1, \quad 0 \leq \mu \leq 4 \quad (16)$$

where μ is a control parameter, and when $\mu > 3.57$ and $z_i \neq 0.25, 0.5, 0.75$, the system goes into chaotic state. In this

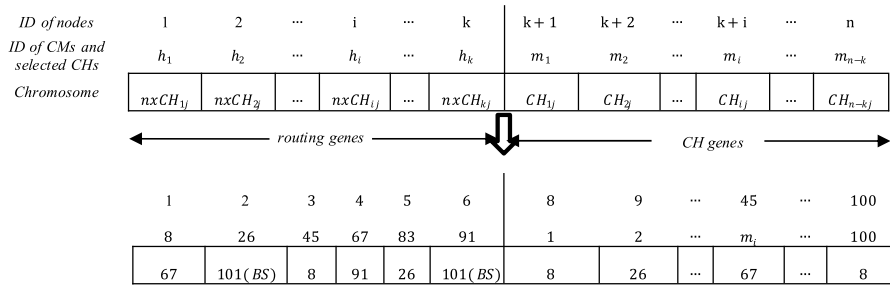


FIGURE 4. Chromosome representation for routing paths and CHs selection.

paper, $\mu = 4$ like in [33], [34], b is the number of genes of a chromosome. In order to produce a chromosome, first, randomly produce k unequal genes according to Eq. (16) as CHs which must be satisfied with the constraint condition given by:

$$\forall m_j | (CH \cap CH_{ij}) \neq \emptyset, \quad i \in [1 - k], j \in [1, n - k] \quad (17)$$

That is to say at least one of the candidate CHs of each CM must exist in the set of CHs, accordingly, one CM has a valid CH. Second, one of the candidate next-hop CHs of each CH h_i is randomly selected as its net-hop CH. Last, a valid chromosome is finished by combining the CHs for each CM with the selected candidate next-hop CHs for each CH. As illustrated in the example of Fig. 4, 100 nodes $S = \{1, 2, \dots, 100\}$ are deployed in a network, and six valid CHs $CH = \{8, 26, 45, 67, 83, 91\}$ are selected, that is, for CMs m_1, m_{12}, m_{94} , their CHs are 8, 26, 8 respectively, and the like. Also, for CH 8, 26, 45, 67, 83, 91, their next-hop CHs are 67, BS, 8, 91, 26, BS. Consequently, the clusters and routing paths can be obtained from the chromosome. There are six clusters, namely $\{8, 1, 100, \dots\}, \{26, 2, \dots\}, \{45, \dots\}, \{67, \dots\}, \{83, \dots\}$ and $\{91, \dots\}$. And the routing paths for each CH are $8 \rightarrow 67 \rightarrow 91 \rightarrow BS, 26 \rightarrow BS, 45 \rightarrow 8 \rightarrow 67 \rightarrow 91 \rightarrow BS, 67 \rightarrow 91 \rightarrow BS, 83 \rightarrow 26 \rightarrow BS, \text{ and } 91 \rightarrow BS$. The detail process of a valid chromosome production is illustrated in Fig. 5.

B. CONSTRUCTING FITNESS FUNCTION

Fitness function plays a vital role in seeking for the optimal solution of genetic algorithm, which is used to assess the quality of the individuals. Minimizing network energy consumption and balancing network load are two main objectives of CRCGA. At first, it is necessary to decrease the amount of energy consumption of CMs and CHs as much as possible. So the residual energy of CMs and CHs is considered as a factor named α for the fitness function, which is expressed as $E_{totalres} = n * E_{init} - E_{total}$. E_{init} is the initial energy of all nodes. Secondly, due to its great influence on energy efficiency, the load balancing among CHs is used as the other factor named β for the fitness function. Therefore, the fitness

function in CRCGA is expressed as follows:

$$\text{Fitness} = \frac{1}{1 + \alpha} + \beta \quad (18)$$

where α and β are the normalized values of network energy consumption and load balancing, which are expressed respectively as follows:

$$\beta = \frac{LB_{CHS} - LB_{CHSmin}}{LB_{CHSmax} - LB_{CHSmin}} \quad (19)$$

$$\alpha = \frac{E_{totalres} - E_{totalresmin}}{E_{totalresmax} - E_{totalresmin}}$$

where LB_{CHSmin} and LB_{CHSmax} are the minimum and maximum of LB_{CHS} for all individuals, and $E_{totalresmin}$ and $E_{totalresmax}$ are the minimum and maximum of $E_{totalres}$. It can be seen from Eq. (18) that the smaller the value of the fitness function is, the better the quality of the individual is, and then the more likely it is to pass on to the next generation, until the optimal individual with the minimum fitness function value is obtained.

C. IMPOSING GENETIC OPERATORS

Before imposing genetic operators, the initial population is produced by continuously calling the procedure *produce_chromosome* according to the population size. And then the fitness value of each chromosome is calculated and arranged in ascending order. Soon afterwards, the selection, crossover and mutation operators are used to produce the next population.

Selection is used to make the better individuals with superior valuation based on their fitness function values have a greater probability to be selected for the next generation, which imitates the natural selection process. The less the value of the fitness function, the closer the individual is to the optimal solution, and the greater probability it is to be selected. In CRCGA, elitist selection strategy is used to make the elitist individuals directly be selected to the next generation population. For the other individuals, each one determines whether its fitness function value is less than that of a random individual generated by the procedure *produce_chromosome*. If it is, the individual is selected for crossover operation, otherwise, the randomly generated one is selected to accelerate convergence as well as ensure the diversity of the population.

```

1: Procedure chrom = produce_chromosome (int n)
2: for i=1 to n do
3:   Init_ID_list[i] = i;
4:   Sel_ID_list[i] = 0;
5:   chrom[i] = 0;
6: end for
7: CH = Chaotic_genes(n, k);
8: Ascending CH;
9: j = 1; m = 1;
10: for i=1 to n do
11:   if Init_ID_list[i] != CH[m] then
12:     CM[j] = Init_ID_list[i];
13:     j++;
14:   else
15:     m++;
16:   end if
17: end for
18: for i = 1 to n do
19:   if i > k then
20:     Sel_ID_list[i] = CM[i-k];
21:   else
22:     Sel_ID_list[i] = CH[i];
23:   end if
24: end for
25: for i = 1 to n do
26:   if i < k then
27:     chrom[i] = random_select(nxCHi);
28:   else
29:     chrom[i] = random_select(CHi);
30:   end if
31: end for
32: end Procedure

```

FIGURE 5. Producing a valid chromosome.

Crossover means a reproduction process permitting the genes of parents to be forwarded to their offsprings so as to obtain improved characteristics. Two-point crossover is used in CRCGA, in which two random crossover points are located in two parts of the chromosome respectively. An example is illustrated in Fig. 6.

The fitness function values of the children are calculated and compared with their parents, and if the value of a child is less than that of its parent, it is selected for mutation operation. Or else, the procedure *produce_chromosome* is used to produce a random individual, and also the fitness function value is computed to decide whether it is less than that of the parent. If it is, the random individual is selected for mutation operation, and vice versa. In this way, the convergence speed is further accelerated.

Mutation offers changes into genes for a more substantial searching capability as well as diversity promotion. Bit mutation operator is carried out in CRCGA, which determines a random mutation point to change the corresponding gene according to Eq. (16), and then the new individual is obtained. Furthermore, its fitness function value is calculated to determine whether it is superior to its parent, and the better one is selected for the next generation. Fig. 7 shows an example of bit mutation.

After the mutation operation, all the individuals are combined with the selected elitist individuals directly. Then the next generation population is produced.

D. FINDING THE OPTIMAL SOLUTION

By iterating the genetic operators, generation by generation, optimized population is produced. In this way, the optimal solution can be obtained as long as one of the termination conditions is satisfied. That is to say, the optimal CHs and routing paths have been found simultaneously. Without loss of generality, the iteration number is considered as one of the termination conditions, which is usually adopted to avoid algorithms falling in an infinite loop [27]–[33]. Besides, the deviation degree of the fitness function values is used as the other termination condition, which is expressed as follows:

$$\left| \frac{\sum_{i=1}^{\omega} Fitness_i}{\omega} - Fitness_{max} \right| \leq \varepsilon \quad (20)$$

where ω is the population size, $Fitness_i$ indicates the value of fitness function for individual i , and $Fitness_{max}$ is the maximum value of fitness function in the population, ε is a preset small positive number which represents the similarity of individuals. In CRCGA, $\varepsilon = 10^{-5}$ like in [34]. Although the number of iterations is less than the threshold, the algorithm will terminate as long as the deviation degree is satisfied with the set condition accordingly. Therefore, as soon as finding the optimal solution stops operating, the optimal individual is selected from the population whose fitness function value is minimum. Right now, the optimal CHs and routing paths can be obtained according to the selected optimal individual. Moreover, the selected optimal individual is directly added to the initial population next time, in order to further improve the convergence speed. The detail flow diagram of CRCGA is illustrated in Fig. 8.

	1	2	3	4	5	6	8	9	...	45	...	100
ID	8	26	45	67	83	91	1	2	...	m_i	...	100
ID of CMs and selected CHs	67	101(BS)	8	91	26	101(BS)	8	26	...	67	...	8
Parent 1	91	101(BS)	26	91	26	101(BS)	8	26	...	8	...	83
Parent 2	67	101(BS)	8	8	45	101(BS)	67	91	...	67	...	8
Child 1	-----											
Child 2	-----											

FIGURE 6. Two-point crossover for CRCGA.

Before Mutation	91	101(BS)	26	8	45	101(BS)	67	91	...	8	...	83
					↑ Mutation Point							
After Mutation	91	101(BS)	26	8	26	101(BS)	67	91	...	8	...	83
					└ Gene produced by Chaos calculation							

FIGURE 7. Bit mutation for CRCGA.

E. MAINTAINING CLUSTERS AND ROUTING PATHS

Generally, fixed round time [4], [17], [21] is used for CHs rotation during the entire network operation so as to reduce the energy consumption. However, variable round time has been proved to be more suitable for reducing the network energy consumption [15]. In CRCGA, a new adaptive round time is proposed to save network energy consumption due to frequent CHs rotation and improve network throughput. Here, load balancing and energy balancing of the network are both considered, and are represented as γ and δ respectively, namely:

$$T_{\text{newround}} = 2 * T_{\text{round}} * (1 - \gamma) * (1 - \delta) \quad (21)$$

where:

- T_{round} is the traditional round time
- γ is the energy factor which is given as follows:

$$\gamma = \sqrt{\frac{\sum_{i=1}^n (E_{\text{residual}_i} - \frac{\sum_{i=1}^n E_{\text{residual}_i}}{n'})^2}{n}} \quad (22)$$

and γ is normalized as:

$$\gamma = \frac{\gamma - E_{\text{residual}_{\min}}}{E_{\text{residual}_{\max}} - E_{\text{residual}_{\min}}} \quad (23)$$

$E_{\text{residual}_{\min}}$ and $E_{\text{residual}_{\max}}$ are the minimum and maximum of E_{residual_i} ($0 < i < n'$) respectively, n' is the number of current alive nodes.

- δ is the load factor which is given as follows:

$$\delta = \frac{LB_{\text{CHS}} - Ld_{h_i_{\min}}}{Ld_{h_i_{\max}} - Ld_{h_i_{\min}}} \quad (24)$$

$Ld_{h_i_{\min}}$, $Ld_{h_i_{\max}}$ are the minimum and maximum of Ld_{h_i} respectively. Obviously, no extra overhead is produced for the residual energy because the state of the nodes is usually attached by the nodes in the data packets delivered to the BS. At the beginning of each round, the BS uses chaotic genetic algorithm to obtain the optimal CHs and routing paths, and then broadcasts them to the network together with the calculated round time. All the nodes communicate with each other based on the received information.

F. COMPLEXITY ANALYSIS

The proposed CRCGA focuses on a performance enhancement of energy efficiency and load balancing using a chaotic genetic algorithm and an adaptive round time calculation scheme. So its time complexity can be given as:

$O(\text{CRCGA}) = O(\text{time complexity of CGA} + \text{time complexity of adaptive round time calculation})$. CGA consists of population initialization, genetic operators and finding the optimal solution, which is an iterative algorithm with the number of iterations I . It can be seen from Fig.5 and Eq. (16) that the time complexity of encoding chromosome is $O(k*n)$, where k is the number of CHs, and n is the number of nodes or the length of chromosomes, so the time complexity of population initiation is $O(P*k*n)$, where P is the population size. After that, the genetic operators and fitness function values calculation are iterated. Obviously, the time complexities of elitist selection, two point crossover, bit mutation and fitness function values calculation are $O(n)$,

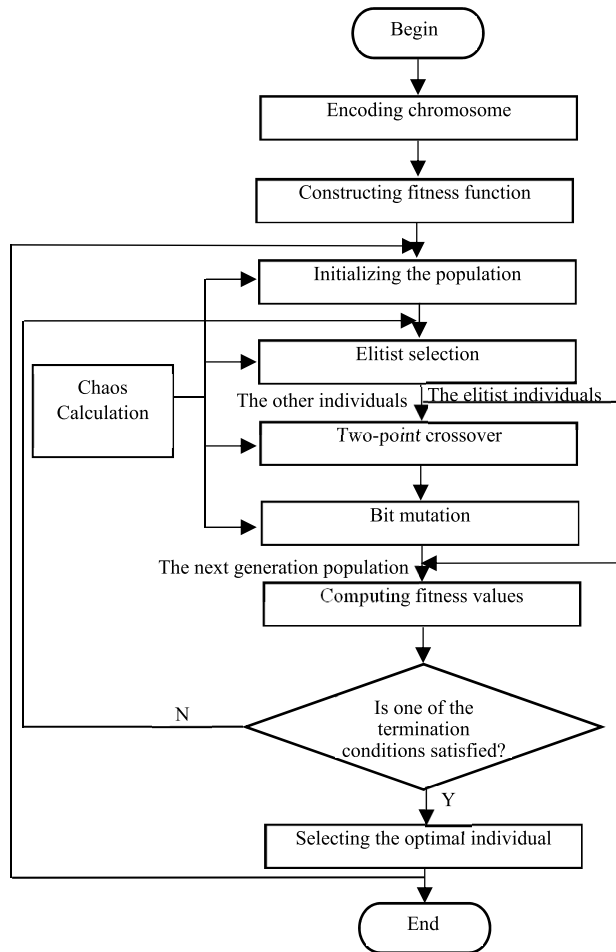


FIGURE 8. The flow diagram of CRCGA.

$O(n/2)$, $O(k*n)$, $O(P*n)$ respectively. Consequently, the time complexity of the genetic operators is $O(I*(3/2 + k + P)*n)$. Once one termination condition is satisfied, the optimal solution is found by bubble sort whose average time complexity is $O(n^2)$ while $O(n)$ and $O(n^2)$ in best and worst cases. Then the time complexity of CGA is $O(P*k*n + I*(3/2 + k + P)*n + n^2)$. Moreover, it can be seen from Eq. (21)-(24) that the time complexity of adaptive round time calculation is $O(n)$. Therefore, the time complexity of $O(\text{CRCGA})$ is $O(P*k*n + I*(5/2 + k + P)*n + n^2)$, without loss of generality, the number of CHs is far less than the number of nodes in WSNs, that is $k \ll n$, and the population size usually equals the length of chromosomes in genetic algorithms, and in CRCGA, the length of chromosomes equals the number of nodes, so $P = n$. Besides, the number of iterations I is a preset constant. As a result the time complexity of $O(\text{CRCGA})$ can be simplified as $O(n^2)$.

Similarly, the space complexity of CRCGA can be obtained by $O(\text{CRCGA}) = O(\text{space complexity of CGA} + \text{space complexity of adaptive round time calculation})$. Seen from Fig.5 and Eq. (16), the space complexity of encoding chromosome is $O(1)$, so the space complexity of population

initialization is also $O(1)$. For genetic operators, although they are run iteratively, all operations and results are based on the initial population matrix and do not require additional storage units. Moreover, during an iteration process, only the two-point crossover needs $2 \sim n$ temporary storage units to hold the genes to be exchanged according to the locations of two crossover points, and in order to sort individuals according to the fitness function values for elitist selection, P temporary storage units are added to the population as a column, so the space complexity of CGA is $O(P+n)$. In addition, the space complexity of finding the optimal solution by bubble sort is $O(1)$. Also, the space complexity of adaptive round time calculation is obviously $O(1)$. Therefore, the space complexity of CRCGA is $O(n + P)$ which can be simplified as $O(n)$.

V. PERFORMANCE EVALUATION

In this section, simulation experiments are performed to evaluate the effectiveness of CRCGA using MATLAB. At first, 100 nodes are randomly deployed in an area of $100m*100m$, with the BS located at the center. The specific parameter settings are illustrated in Table 1, which is similar to those in [10], [13]. The sensor nodes are fixed and the external environment does not affect the radio signals adversely [20]. The results are obtained from the average value of 10 tests. Moreover, we compare the proposed CRCGA with the existing related algorithms LEACH [3], GECR [10], OMPFM [13], and GADA-LEACH [20] using the metrics convergence speed, energy efficiency, load balancing, network throughput and network lifetime.

TABLE 1. Parameter settings.

Parameters	Values
Initial energy of node	1J
E_{elec}	50nJ/bit
ϵ_{fs}	10pJ/bit/m ²
ϵ_{mp}	0.0013pJ/bit/m ⁴
Cluster head proportion	0.1
E_{pDb}	5nj/bit
Packet size	4000 bits
Message size	100 bits
Maximum communication range of nodes/ d_{max}	50m
Network size	100*100m ²
Crossover rate	0.7
Mutation rate	0.1
Population size	100

The convergence speed is the the number of iterations from producing the initial population to the final determination of the optimal chromosome, which is firstly tested to indicate the time complexity of CRCGA compared with the relevant

genetic algorithm based clustering routing protocols. The results are shown in Fig. 9.

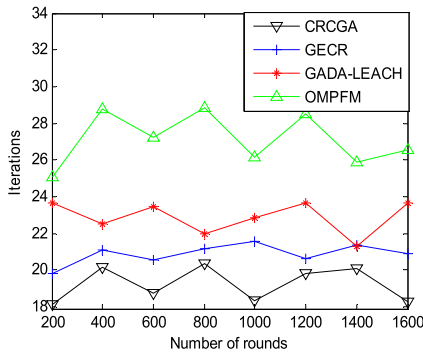


FIGURE 9. Comparison of convergence speed.

It can be seen from Fig.9 that OMPFM has the slowest convergence speed and CRCGA has the fastest convergence speed. Binary coded chromosomes used for routing search in OMPFM make genetic algorithm run once for each source CH, thus slowing down the convergence speed. Although binary coded chromosomes are also used for CHs selection in GADA-LEACH, the genetic algorithm is only run once for all CMs, so its convergence speed is faster than that of OMPFM. Similarly, real number coded chromosomes in GEGR and CRCGA for CHs selection and routing paths search simultaneously make genetic algorithm run once for all nodes. Moreover, pre-process is carried out to exclude some low energy or far distance nodes, and the last optimal individual is directly added to the population next time, which are further improve the convergence speed of GEGR and CRCGA compared to OMPFM and GADA-LEACH. Especially, the deviation degree of the fitness function values, used as the other termination condition of chaotic genetic algorithm in CRCGA, makes it converge faster than GEGR. As a result the convergence speed of CRCGA is increased by 7.95%, 15.96% and 29.09% respectively compared to GEGR, GADA-LEACH and OMPFM.

In order to evaluate the energy efficiency of CRCGA, the energy consumption of all nodes and the standard deviation of residual energy are tested, the results are illustrated in Fig. 10 and 11 respectively.

It can be seen from Fig. 10 and 11 that the energy consumption of CRCGA is not only less but also more balanced than LEACH, GADA-LEACH, OMPFM and GEGR. Due to its single-hop, random CH selection without considering residual energy, LEACH is easy to produce isolated nodes and uneven clusters which result in the worst performance. Minimizing the total energy consumption of the network for fitness function construction during the process of CHs selection and finding routes makes GEGR perform better than GADA-LEACH only for CH selection and OMPFM only for finding routes. Furthermore, the results also indicate that adaptive round time can also affect the network energy efficiency. In summary, the total energy consumption

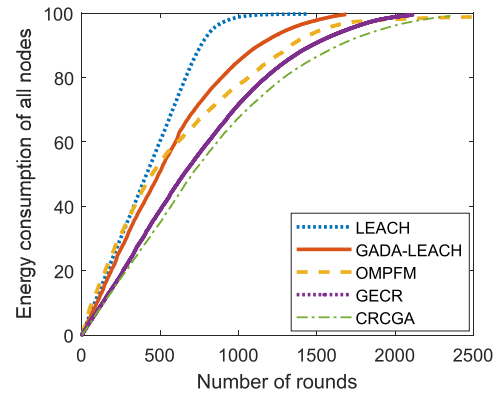


FIGURE 10. Comparison of energy consumption of all nodes.

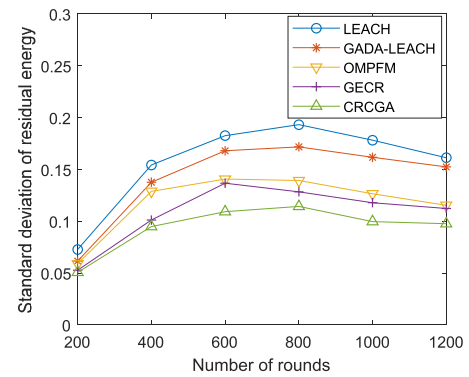


FIGURE 11. Comparison of standard deviation of residual energy of all nodes.

of CRCGA is 35.51%, 20.03%, 11.7%, 5.08% lower than those of LEACH, GADA-LEACH, OMPFM and GEGR. The standard deviation of residual energy is also 39.9%, 33.61%, 20.33%, 12.78% lower than those of the other algorithms.

Load balancing is usually used to measure the uniformity of load and energy consumption among CHs, which is decided by the number of members and the number of selected relay nodes. In order to compare load balancing more clearly, the standard deviation of load on CHs is tested and the results are shown in Fig. 12.

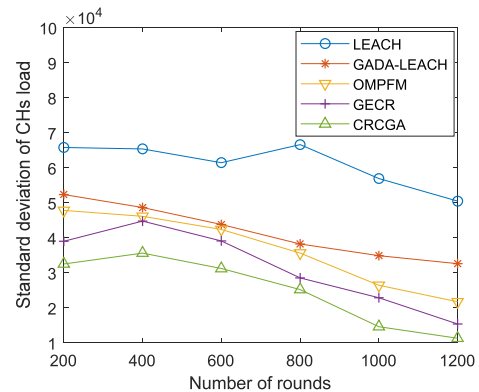


FIGURE 12. Comparison of standard deviation of load balancing.

It is obvious from Fig. 12 that LEACH unevenly distributes its load on CHs due to its random CHs selection. Though GADA-LEACH can select efficient nodes as CHs through a genetic algorithm, the normal nodes join a cluster only according to its distance to CHs like LEACH, which forms unequal sized clusters, thus causing an unequal load on CHs. OMPFM considers not only CHs selection but also routes finding. However, it still does not consider the number of cluster members while forming clusters, but CRCGA and GEGR do. Moreover, CRCGA can find the better CHs and routing paths than GEGR by utilizing a chaotic genetic algorithm. The standard deviation of load balancing of CRCGA is 58.92%, 39.9%, 31.61%, 20.63% lower than those of LEACH, GADA-LEACH, OMPFM and GEGR.

The network throughput denoted by the received valid packets of the BS in this paper is always used to measure the quality of service of the network. Moreover, it is also the direct reflection on CHs distribution and load balancing. The results are illustrated in Fig. 13.

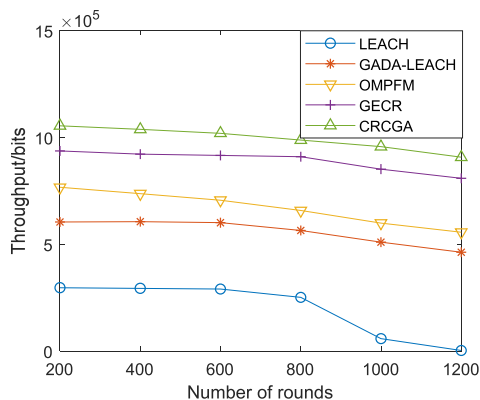


FIGURE 13. Comparison of network throughput.

As can be seen from Fig. 13, the network throughput decreases as the number of rounds increases. So it can be concluded that less packets will be forwarded to the BS when there are more isolated nodes or dead nodes in the network. In practically, CRCGA outperforms LEACH by 79.96%, GADA-LEACH by 43.84%, OMPFM by 32.53% and GEGR by 10.37% respectively in the network throughput.

Next, the number of dead nodes and the number of living nodes and FND (First Node Dies), HND (Half Nodes Die), LND (Last Node Dies) in rounds are tested to evaluate the network lifetime of CRCGA, and the results are shown in Fig. 14, 15 and Table 2.

Seen from Fig. 14, there are 10% of nodes die at round 690, 617, 446, 842, 896 in LEACH, GADA-LEACH, OMPFM, GEGR and CRCGA respectively. 90% of dead nodes occur at round 1005, 1679, 1805, 2100, 2544 in LEACH, GADA-LEACH, OMPFM, GEGR and CRCGA. It can be known from Table 2 that the FNDs of GADA-LEACH, OMPFM, GEGR and CRCGA are 336, 328, 349 and 377 rounds, which are less than that of LEACH, because the CHs near to the BS in GADA-LEACH, OMPFM, GEGR and CRCGA undertake

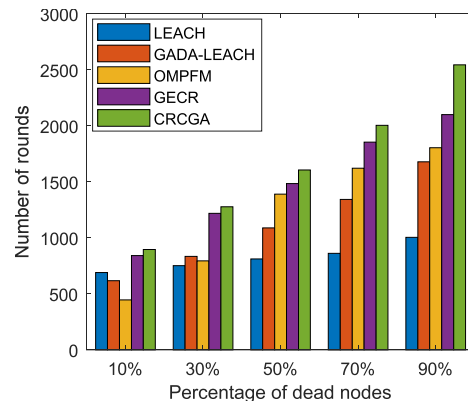


FIGURE 14. Comparison of the number of dead nodes.

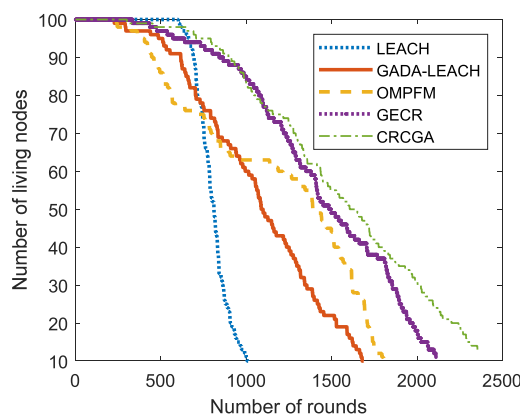


FIGURE 15. Comparison of the number of living nodes.

TABLE 2. Comparison of FND, HND and LND.

Items	Protocols				
	LEACH	GADA-LEACH	OMPFM	GEGR	CRCGA
FND	603	336	328	349	377
HND	811	1088	1392	1485	1606
LND	1446	1842	1953	2127	2633

data forwarding tasks resulting in premature death. However, the CHs in LEACH communicate with the BS directly. With the running of the network, the CHs far away from the BS die quickly in LEACH due to their direct communication with the BS. Therefore, the LND of LEACH is much smaller than those of GADA-LEACH, OMPFM, GEGR and CRCGA. In short, the number of rounds obtained in CRCGA, i.e. lifetime, is increased by 23.79%, 34.82%, 42.94% and 82.09%, respectively. Especially, no matter which round it is, CRCGA has the lowest number of dead nodes. Moreover, CRCGA has the slowest change rate in the number of living nodes according to Fig. 15. Thus one can see that CRCGA has the longest network lifetime.

Finally, the network lifetime is further evaluated in scenarios with 600, 800 and 1000 nodes deployed in a vast area of 1000m*1000m and 1000 nodes deployed in the areas of 600m*600m, 800m*800m, 1000m*1000m. The results are shown in Table 3 and Table 4.

TABLE 3. FND, HND and LND in the area of 1000m*1000m with different nodes.

Nodes scale		Protocols				
		LEACH	GADA-LEACH	OMPFM	GEGR	CRC GA
1000	FND	1	75	84	106	121
	HND	11	547	687	755	803
	LND	288	706	933	992	1143
800	FND	1	58	66	89	103
	HND	14	497	606	698	732
	LND	327	649	848	953	982
600	FND	1	44	56	63	70
	HND	10	412	534	604	626
	LND	338	593	765	903	926

TABLE 4. FND, HND and LND in different areas with 1000 nodes.

Area		Protocols				
		LEA CH	GADA-LEACH	OMP FM	GEGR	CRC GA
1000 m*1000m	FND	1	75	84	106	121
	HND	11	547	687	755	803
	LND	288	706	933	992	1143
800m*800m	FND	1	117	138	168	192
	HND	28	659	812	906	972
	LND	601	760	995	1070	1239
600m*600m	FND	1	132	156	189	216
	HND	72	695	855	958	1022
	LND	661	806	1047	1125	1302

It can be seen from Table 3 and Table 4 that the network lifetime decreases with the increase of the network area. That is because the average hop count increases in vast area as well as the distance between some clusters, leading to an exponential increase in energy consumption. At the same time, the network lifetime increases with the number of nodes in the vast area network because the more uniform distribution of clusters and the smaller average distance between clusters will be achieved if there are more nodes deployed in the network, so less energy will be consumed accordingly. As for

LEACH in the vast area network, CHs that are far from the BS die rapidly, so FND occurs at the first round. Compared to GADA-LEACH and OMPFM, GEGR and CRCGA design the fitness functions based on the network energy consumption minimization, so their network lifetime is larger than that of GADA-LEACH and OMPFM. For CRCGA, adaptive round time and load balancing consideration are used to further extend the network lifetime. Therefore its network lifetime is the largest in all the scenarios. In other words, the scalability of CRCGA is much better than LEACH, GADA-LEACH, OMPFM and GEGR. It is thus clear that LEACH using single hop communication is not suitable for applications in large scale WSNs due to its worst scalability. To be specific, compared with GEGR, GADA-LEACH, OMPFM and LEACH, the network lifetime of CRCGA is increased by 15.22%, 61.9%, 22.51% and 296.9% when the number of nodes in area of 1000m*1000m is 1000, by 3.04%, 51.31%, 15.8%, 200.31% when the number of nodes is 800, by 2.55%, 56.16%, 21.05% and 173.96% when the number of nodes is 600. Moreover, the network lifetime of CRCGA is also increased by 15.8%, 63.03%, 24.52%, 106.16% when the area with 1000 nodes is 800m*800m, by 15.73%, 61.53%, 24.36% and 96.97% when the area with 1000 nodes is 600m*600m.

VI. CONCLUSION

In this paper, an energy efficient and load balanced clustering routing protocol called CRCGA based on a chaotic genetic algorithm is proposed to minimize the network energy dissipation. To this end, the best CHs are selected and the optimal routing paths are found simultaneously. It's done by running the chaotic genetic algorithm with the definition of a novel fitness function considering minimizing the energy consumption of all the nodes and balancing the loads of CHs in each round. Moreover, an adaptive round time considering energy efficiency and load balancing is calculated to further reduce the total energy consumption of the network. Simulations are conducted to prove the effectiveness of CRCGA whose convergence speed is the fastest in all of the network rounds. In terms of energy efficiency, CRCGA consumes the smallest amount of energy while the energy consumption is the most balanced. The average network throughput in CRCGA is increased by 10.37%, 79.96% compared with the second most optimal algorithm GEGR and the worst algorithm LEACH. Meanwhile, the living nodes of CRCGA is the largest at most times, and the network lifetime of CRCGA is 15.22%, 3.04%, 2.55% and 296.9, 200.31%, 173.96% longer than those of GEGR and LEACH in the network area of 1000m*1000m with nodes of 1000, 800 and 600. Also its lifetime in scenarios with 1000 nodes in the network areas of 800m*800m and 600m*600m is increased by 15.8%, 15.73% and 106.16%, 96.97% compared with GEGR and LEACH respectively. In summary, CRCGA is superior to LEACH, GEGR, OMPFM, and GADA-LEACH in terms of convergence speed, energy efficiency, load balancing, network throughput and lifetime.

Although CRCGA has made some contributions as mentioned above, it is presented for the static WSNs. In the future, the artificial neural network along with chaotic genetic algorithm will be used to further improve the performance of multi-hop clustering routing protocol in the mobile WSNs. Moreover, QoS of the links such as reliability and security will also be considered. Especially, a further direction will carry out practical application in real wireless environment for WSNs.

REFERENCES

- [1] D. Wohwe Sambo, B. Yenke, A. Förster, and P. Dayang, "Optimized clustering algorithms for large wireless sensor networks: A review," *Sensors*, vol. 19, no. 2, p. 322, Jan. 2019.
- [2] F. Faniian and M. Kuchaki Rafsanjani, "Cluster-based routing protocols in wireless sensor networks: A survey based on methodology," *J. Netw. Comput. Appl.*, vol. 142, pp. 111–142, Sep. 2019.
- [3] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," in *Proc. Syst. Sci.*, Honolulu, HI, USA, 2000, Art. no. eneti005-3014.
- [4] J. Hong, J. Kook, S. Lee, D. Kwon, and S. Yi, "T-LEACH: The method of threshold-based cluster head replacement for wireless sensor networks," *Inf. Syst. Frontiers*, vol. 11, no. 5, pp. 513–521, Nov. 2009.
- [5] X.-X. Ding, M. Ling, Z.-J. Wang, and F.-L. Song, "DK-LEACH: An optimized cluster structure routing method based on LEACH in wireless sensor networks," *Wireless Pers. Commun.*, vol. 96, no. 4, pp. 6369–6379, Jul. 2017.
- [6] S. A. Fatemi Aghda and M. Mirfakhraei, "An improved cluster routing protocol to increase the lifetime of wireless sensor network (WSN)," *Wireless Pers. Commun.*, vol. 109, no. 3, pp. 2067–2075, Dec. 2019.
- [7] N. Khoulalene, L. Bouallouche-Medjokoune, D. Aissani, A. Mani, and H. Ariouat, "Clustering with load balancing-based routing protocol for wireless sensor networks," *Wireless Pers. Commun.*, vol. 103, no. 3, pp. 2155–2175, Jun. 2018.
- [8] M. Azharuddin and P. K. Jana, "PSO-based approach for energy-efficient and energy-balanced routing and clustering in wireless sensor networks," *Soft Comput.*, vol. 21, no. 22, pp. 6825–6839, Jun. 2016.
- [9] H. El Alami and A. Najid, "ECH: An enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks," *IEEE Access*, vol. 7, pp. 107142–107153, 2019.
- [10] T. Wang, G. Zhang, X. Yang, and A. Vajdi, "Genetic algorithm for energy-efficient clustering and routing in wireless sensor networks," *J. Syst. Softw.*, vol. 146, pp. 196–214, Dec. 2018.
- [11] S. Al-Sodairi and R. Ouni, "Reliable and energy-efficient multi-hop LEACH-based clustering protocol for wireless sensor networks," *Sustain. Comput., Informat. Syst.*, vol. 20, pp. 1–13, Dec. 2018.
- [12] V. Vijayalakshmi and A. Senthilkumar, "USCDRP: Unequal secure cluster-based distributed routing protocol for wireless sensor networks," *J. Supercomput.*, vol. 76, no. 2, pp. 989–1004, Nov. 2019.
- [13] M. Al-Shalabi, M. Anbar, T.-C. Wan, and Z. Alqattan, "Energy efficient multi-hop path in wireless sensor networks using an enhanced genetic algorithm," *Inf. Sci.*, vol. 500, pp. 259–273, Oct. 2019.
- [14] S. A. Sert, H. Bagci, and A. Yazici, "MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks," *Appl. Soft Comput.*, vol. 30, pp. 151–165, May 2015.
- [15] F. Zhu and J. Wei, "An energy-efficient unequal clustering routing protocol for wireless sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 9, pp. 1–15, Sep. 2019.
- [16] Y. H. Robinson, E. G. Julie, R. Kumar, and L. H. Son, "Probability-based cluster head selection and fuzzy multipath routing for prolonging lifetime of wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1061–1075, May 2019.
- [17] A. Genta, D. K. Lobiyal, and J. H. Abawajy, "Energy efficient multipath routing algorithm for wireless multimedia sensor network," *Sensors*, vol. 19, no. 17, pp. 1–21, Aug. 2019.
- [18] S. A. Sert, A. Alchihabi, and A. Yazici, "A two-tier distributed fuzzy logic based protocol for efficient data aggregation in multihop wireless sensor networks," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3615–3629, Dec. 2018.
- [19] X. Cai, Y. Sun, Z. Cui, Z. Cui, W. Zhang, and J. Chen, "Optimal LEACH protocol with improved bat algorithm in wireless sensor networks," *KSI Trans. Internet Inf. Syst.*, vol. 13, no. 5, pp. 2469–2490, May 2019.
- [20] T. Bhatia, S. Kansal, S. Goel, and A. K. Verma, "A genetic algorithm based distance-aware routing protocol for wireless sensor networks," *Comput. Electr. Eng.*, vol. 56, pp. 441–455, Nov. 2016.
- [21] M. K. Sohrabi and S. Alimirzaee, "Improving performance of node clustering in wireless sensor networks using meta-heuristic algorithms and a novel validity index," *J. Supercomput.*, vol. 75, no. 11, pp. 7550–7572, Aug. 2019.
- [22] J.-S. Lee and C.-L. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1095–1103, Aug. 2017.
- [23] B. Pitchaimanickam and G. Murugaboopathi, "A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 7709–7723, Jun. 2020.
- [24] S. Arjunan and P. Sujatha, "Lifetime maximization of wireless sensor network using fuzzy based unequal clustering and ACO based routing hybrid protocol," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2229–2246, Nov. 2017.
- [25] H. El Alami and A. Najid, "EEA: Clustering algorithm for energy-efficient adaptive in wireless sensor networks," *Int. J. Wireless Netw. Broadband Technol.*, vol. 7, no. 2, pp. 19–37, Jul. 2019.
- [26] W. Wang and G. Tong, "Multi-path unequal clustering protocol based on ant colony algorithm in wireless sensor networks," *IET Netw.*, vol. 9, no. 2, pp. 56–63, Mar. 2020.
- [27] S. Arjunan and S. Pothula, "A survey on unequal clustering protocols in wireless sensor networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 31, no. 3, pp. 304–317, Jul. 2019.
- [28] L. Kong, J.-S. Pan, V. Snášel, P.-W. Tsai, and T.-W. Sung, "An energy-aware routing protocol for wireless sensor network based on genetic algorithm," *Telecommun. Syst.*, vol. 67, no. 3, pp. 451–463, Mar. 2018.
- [29] S. Dehghani, B. Barekatain, and M. Pourzaferani, "An enhanced energy-aware cluster-based routing algorithm in wireless sensor networks," *Wireless Pers. Commun.*, vol. 98, no. 1, pp. 1605–1635, Jan. 2018.
- [30] P. S. Ragavan and K. Ramasamy, "Optimized routing in wireless sensor networks by establishing dynamic topologies based on genetic algorithm," *Cluster Comput.*, vol. 22, no. S5, pp. 12119–12125, Sep. 2019.
- [31] Y. Song, C. Gui, X. Lu, H. Chen, and B. Sun, "A genetic algorithm for energy-efficient based multipath routing in wireless sensor networks," *Wireless Pers. Commun.*, vol. 85, no. 4, pp. 2055–2066, Jul. 2015.
- [32] G. Yao, Z. Dong, W. Wen, and Q. Ren, "A routing optimization strategy for wireless sensor networks based on improved genetic algorithm," *Tamkang Inst. Technol.*, vol. 19, no. 2, pp. 221–228, Jan. 2016.
- [33] S. K. Gupta and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: GA based approach," *Wireless Pers. Commun.*, vol. 83, no. 3, pp. 2403–2423, Apr. 2015.
- [34] C. Yanan and W. Muqing, "A novel RPL algorithm based on chaotic genetic algorithm," *Sensors*, vol. 2018, no. 18, pp. 1–20, Sep. 2018.
- [35] M. Javidi and R. Hosseinpoufard, "Chaos genetic algorithm instead genetic algorithm," *Int. Arab. J. Inf. Technol.*, vol. 12, no. 2, pp. 162–167, Mar. 2015.
- [36] F. Guillermino, V. Manuel, and A. Miguel, "Chaotic genetic algorithm and the effects of entropy in performance optimization," *Chaos*, vol. 29, no. 1, pp. 1–7, Jan. 2019.
- [37] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Comput.*, vol. 22, no. 9, pp. 2935–2952, May 2018.
- [38] H. Mostafaei and M. S. Obaidat, "Learning automaton-based self-protection algorithm for wireless sensor networks," *IET Netw.*, vol. 7, no. 5, pp. 353–361, Sep. 2018.
- [39] H. Mostafaei, "Energy-efficient algorithm for reliable routing of wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5567–5575, Jul. 2019.
- [40] A. B. Huda and A. A. Saad, "Efficient and secure data transmission and sinkhole detection in a multi-clustering wireless sensor network based on homomorphic encryption and watermarking," *IEEE Access*, vol. 8, pp. 92098–92109, 2020.
- [41] S. Ramesh and C. Yaashuwanth, "Enhanced approach using trust based decision making for secured wireless streaming video sensor networks," *Multimedia Tools Appl.*, vol. 79, nos. 15–16, pp. 10157–10176, Apr. 2019.



CHUHANG WANG received the master's degree in computer application from Jilin University, China, in 2005. She is currently an Assistant Professor with the College of Computer Science and Technology, Changchun Normal University, China. Her research interests include wireless sensor networks and realtime embedded systems.



YOUJIA HAN is currently pursuing the master's degree with the College of Computer Science and Engineering, Changchun University of Technology, China. His research interests include security of wireless sensor networks and RFID.



XIAOLI LIU is currently pursuing the master's degree with the College of Computer Science and Technology, Changchun Normal University, China. Her research interests include intelligent algorithms in wireless sensor networks and embedded systems.



HUANGSHUI HU received the Ph.D. degree in computer application technology from Jilin University, China, in 2012. He is currently a Professor with the College of Computer Science and Engineering, Jilin University of Architecture and Technology, China. His research interests include topology control in wireless sensor networks and multifunction vehicle bus networks.



MEIQIN YAO is currently pursuing the master's degree with the College of Computer Science and Engineering, Changchun University of Technology, China. Her research interests include clustering routing algorithms of wireless sensor networks and multifunction vehicle bus networks.

...