

# Research on Low-energy Adaptive Clustering Hierarchy Protocol based on Multi-objective Coupling Algorithm

Wuzhao Li<sup>1\*</sup>, Yechuang Wang<sup>2</sup>, Youqiang Sun<sup>2</sup>, Jie Mao<sup>1</sup>

<sup>1</sup>Department of Information Technology, Jiaxing Vocational and Technical College  
Jiaxing, Zhejiang, 314036, China.  
[e-mail: rick\_2006003@139.com]  
[e-mail: jm0321@163.com]

<sup>2</sup>Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology  
Taiyuan, Shanxi, 030024, China  
[e-mail: yechuangwang@sina.com]  
[e-mail: youqiang\_sun@163.com]

\*Corresponding author: Wuzhao Li

*Received August 13, 2019; revised October 31, 2019; revised December 1, 2019;  
accepted December 17, 2019; published April 30, 2020*

---

## Abstract

Wireless Sensor Networks (WSN) is a distributed Sensor network whose terminals are sensors that can sense and check the environment. Sensors are typically battery-powered and deployed in where the batteries are difficult to replace. Therefore, maximize the consumption of node energy and extend the network's life cycle are the problems that must to face. Low-energy adaptive clustering hierarchy (LEACH) protocol is an adaptive clustering topology algorithm, which can make the nodes in the network consume energy in a relatively balanced way and prolong the network lifetime. In this paper, the novel multi-objective LEACH protocol is proposed, in order to solve the proposed protocol, we design a multi-objective coupling algorithm based on bat algorithm (BA), glowworm swarm optimization algorithm (GSO) and bacterial foraging optimization algorithm (BFO). The advantages of BA, GSO and BFO are inherited in the multi-objective coupling algorithm (MBGF), which is tested on ZDT and SCH benchmarks, the results are shown the MBGF is superior. Then the multi-objective coupling algorithm is applied in the multi-objective LEACH protocol, experimental results show that the multi-objective LEACH protocol can greatly reduce the energy consumption of the node and prolong the network life cycle.

---

**Keywords:** LEACH protocol, multi-objective, coupling algorithm, wireless sensor network

## 1. Introduction

Wireless sensor networks (WSN) are composed of wireless sensor nodes which has limited storage and processing capacity by self-organization. The sensors have the ability to collect, extract and transmit the information of the environment cooperatively in the network area [1]. In 2003, technology review magazine listed the wireless sensor networks as the first of ten emerging technologies for the future of human life. Meanwhile, WSN is internationally recognized as the second largest network after the Internet [2]. In recent years, wireless sensor network has developed rapidly and has been widely used in all aspects of society, and the application value of WSN is more and more recognized by people, and it has a wide range of applications in the field of military environment and intelligent home. Therefore, the wireless sensor network is the focus of current academic research [3]. Energy conservation is one of the most important research focuses in WSN. The sensor node is powered by batteries, and once the power is exhausted, the battery cannot be replaced manually or otherwise due to the harsh and remote working conditions [4]. The energy consumption and life span of a single node will directly affect the life cycle of the whole network. Therefore, how to save the sensor's energy and balance the consumption between nodes are the most important issues in current WSN research [5].

The emergence of routing protocol makes the energy consumption of wireless sensor networks decrease greatly. A single-hop or multi-hop are used in the protocol when the wireless sensor network transmits the data that collected by the sensors [6-8]. The routing protocol plays an important role in establishing and maintaining data transmission links between sensors in WSNs [9-10], and a proper routing protocol helps to optimize the overall network energy and prolong the network life cycle [11-12]. Therefore, it is an important and effective method to reduce the energy consumption of sensors by optimizing routing protocol [13]. Generally, the existing routing protocols have two types [14]: flat routing protocols and hierarchical routing protocols. In flat routing, all sensor nodes have the same status and function, and the nodes work together to complete the sensing task. As for hierarchical routing, all nodes are clustered according to different clustering methods, the cluster including cluster-head node and non-cluster-head nodes, because the logical structure of the network is hierarchical, the information is passed by cluster-head node. Compared with other types of routing protocols, LEACH [15] has the advantages of stability, expandability, efficiency and energy saving, and has become the focus of the routing protocols.

At present, optimization problems with multiple objective functions are common in many engineering problems [16-17]. The goal of multi-objective intelligent optimization algorithms [18-19] is to find a set of solutions that represent the trade-offs between these objectives, because there is no single solution available to solve these problems in these areas. In recent years, multi-objective optimization technology has been greatly developed, and many studies show that this technology is more practical and effective than traditional multi-objective optimization method. The representative multi-objective optimization algorithms mainly include NSGA [20], NSGA2 [8], SPEA [21], SPEA2 [22], MOEA/D [23], PAES [24] and PESA [25], etc. At the same time, some novel algorithms have been proposed.

Swarm intelligence optimization algorithms [26-32] are mainly designed according to the behavior of many creatures in nature, and used to solve related problems in different fields [33]. K.N.Krishnanad and D. Ghose have conducted an important multi-objective study on glowworm swarm optimization algorithm (GSO) [34]. Wang proposed multi-objective GSO

Algorithm (MGSO) [35], which has a good effect on continuous optimization. GSO Algorithm [36] is an algorithm inspired by foraging behavior of firefly luminescence, the main purpose of Firefly flash is to serve as a signal system to attract other fireflies. Dasgupta S [37] proposed bacterial foraging optimization (BFO) by explained foraging strategy of the bacteria group. Although many swarm intelligence algorithms [38-39] have been proposed in recent years, each has its own merits. On the basis of the no free lunch theorem, a single algorithm cannot solve all the optimization problems. Therefore, a new coupling algorithm which is coupled by three or more algorithms that can overcome the shortcomings of a single algorithm. In this paper, a multi-object coupling algorithm based on GSO, BFO and BA is proposed. The coupling algorithm adopt a parallel method by the simultaneous evolution and mutual communication of each individual in these algorithms. Simulation results show that the accuracy and the stability of coupled algorithm is improved obviously.

The rest of the paper is organized as follows: in section 2, Some related work is introduced in recent years. Then, the model of LEACH protocol and the details of the proposed coupling algorithm are described in section 3. Next, the experimental results are showed and discussed in Section 4. Lastly, Section 5 concludes the paper.

## 2. Related Work

LEACH protocol is proposed by Heinzelman [40] on the Massachusetts institute of technology, which is a kind of low power adaptive layered algorithm. The selection of Cluster-head node is a general optimization problem, this problem not only considers the influence of position, but also the restriction of node energy [41-42]. In order to reduce the network energy consumption, it is necessary to optimize the selection of LEACH cluster-head nodes.

The biological heuristic algorithms [36-43-44] are effective way for working out this engineering problem, such as CS [45], BA [13], GSO [46], BFO [47], PSO [48] and ABC [49] et al. Most engineering problems are multi-objective optimization problems. In practical engineering, obtaining the optimal solution of these problems has always been the focus of academic and engineering circles. Recently, many researchers have been started to solve multi-objective optimization problems by using different algorithm. For dispatch problem, a new multi-objective hybrid bat algorithm (MHBA) is proposed by Liang H et al [50], and an elitist non-dominated sorting method and an improved crowding distance sorting method are introduced. Simulated results shows that MHBA is better than other algorithms for solving large-scale systems. Latif A et al. [51] proposed multi-objective reactive power dispatch in distribution networks using modified bat algorithm. However, it has a large probability of falling into local optimum. Hence, Chen Wei et al. [52] investigate a fuzzy combination and selection problem with multi-objective optimization. A hybrid BA-DE is proposed by combining advantages of bat algorithm (BA) and differential evolution (DE) for combination and selection problem. Krishnanand KN et al. [34] applied glowworm swarm optimization for calculating multiple optimal solution of multimodal functions simultaneously. Experimental results demonstrate that the proposed glowwormswarm optimization algorithm is effective in obtaining multiple optima.

In order to get better solution, Wu B et al [53] improved GSO by two strategies. One is the greedy acceptance criteria. The glowworms update their position by each dimension. Another is the changed movement formulas. Experimental results show that the proposed and improved algorithm can enhance accuracy and convergence rate for global optimization problems. Meanwhile, Tang Z et al. [54] also presented an improved GSO algorithm, the

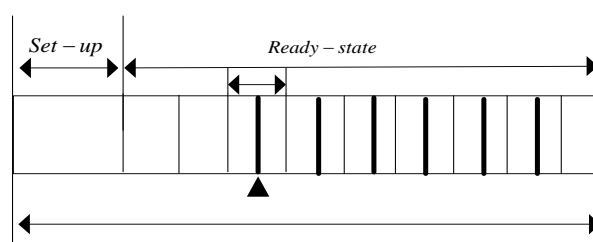
detailed method is altering strategy of mutation. Experimental results show that the improved algorithm has higher calculation accuracy and faster convergence speed compared to standard algorithm. Chen et al. [55] adopted a battery of measures to improve the standard BFO. In the modified method, the scope of searching and step size are changing dynamically. At the same time, to make group move towards the global optimal solution, a variable is introduced, and it represents the overall optimal value. Simulated result shows that it has high efficiency, rapid speed of convergence and strong capability of global search. And Pang B et al. [56] adopted globally optimal solution for improving the swarming performance. And the step size of every bacterium is determined by the random flight length of the improved Lévy flight. Therefore, Okaeme NA et al. [57] proposed a novel heuristic optimization algorithm, which combined the genetic algorithms (GAs) and the bacterial foraging optimization (BFO). And the simulated results reveal that hybrid bacterial foraging optimization strategy is superior than classical algorithm by comparing performance.

To sum up, bat algorithm has strong ability of search optimal solution, but it has a large probability of falling into local optimum. Glowworm swarm optimization is implemented easily. But the convergence rate is slow in the process of complex function optimization. However, Bacterial foraging algorithm is easy to jump out of local optimum. Coupling the three algorithms can make the prediction results more accurate by combine the advantages of the three algorithms. Therefore, the coupling algorithm based on bat algorithm, bacterial foraging optimization and glowworm swarm optimization is designed in this paper.

### 3. Multi-objective LEACH protocol

#### 3.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

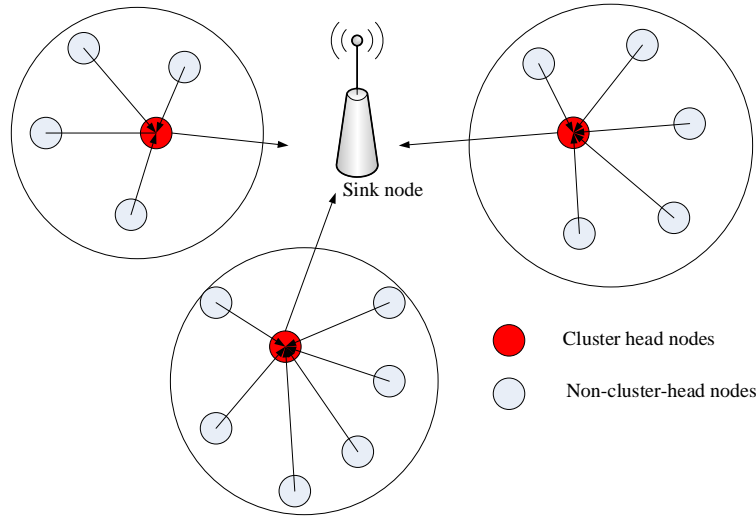
LEACH protocol is the first clustering routing algorithm. For cluster protocols, a cluster consists of a unique cluster-head node and several non-cluster-head nodes. Within the cluster, the non-cluster-head node only communicates with the cluster-head node. Since the sensor energy is given, it is necessary to reduce the overall energy consumption of the network. The LEACH protocol has two phase cycle, establishment phase of the cluster (setup phase) and stable phase of the transmitted data (ready phase). The two phase of LEACH is shown in Fig. 1.



**Fig. 1.** The flow chart of the LEACH

**Fig. 1** shows the composition of the phases of the LEACH protocol. The protocol mainly consists of set-up phase and preparation phase. In the setup phase, two works need to be finished, determining the cluster-head node and establishing the schedule table of the cluster. So, the cluster-head node needs more energy to complete the collection of node information within the cluster and the transfer of information between the base stations. The LEACH protocol tries to make each node consume the same amount of energy, and extending the life of the entire network. At former period each round, the nodes in the wireless sensor network generate a

random number  $R(R \in 0 \sim 1)$ , the node will be a cluster-head node when the  $R < T(n)$ . During the round, the node will not be chosen as the cluster-head node when the  $T(n) = 0$ , not so, the node becomes the cluster-head node with the probability of  $T(n)$ . The  $T(n)$  is represented as follow Equation (1):



**Fig. 2.** The typical topology of LEACH

Although the energy of different sensors are taken into account when selecting cluster-head node, the network performance will be affected due to the random probability in selecting cluster-heads. And there are some problems such as uneven clustering and difficulty in optimizing the number of cluster-head nodes.

$$T(n) = \begin{cases} \frac{P}{1 - P \left[ r \bmod \left( \frac{1}{P} \right) \right]}, & n \in G \\ 0, & \text{other} \end{cases} \quad (1)$$

In Equation (1), the  $r$  represents the number of sheaves, the  $P$  means the probability of cluster-head nodes,  $r \bmod (1/P)$  means the number of times the node becomes a cluster-head node and  $G$  is the set of nodes that not selected as cluster-head node.

During ready phase, the cluster-header divides the working time sequence into multiple time frames and allocates slots for each member node in each frame. The non-cluster-head node state will change from sleep state to work state, when the non-cluster-head node arrives at its own time slot, and it will quickly collect information and send it to the cluster-head node. The cluster-head node will send the data to the sink node after processing. As shown in **Fig. 1**, the set-up phase of each round occupies a relatively short actual time, while the ready phase occupies the main part of each round, which is conducive to reducing the extra energy consumption caused by frequent clustering. Thus, it has greatly improved energy efficiency and extended the life cycle of the network.

As shown in **Fig. 2**, LEACH adopts single-hop clustering network model, and the network structure elements in the model include wireless sensor nodes and sink node. In LEACH, the cluster-head node collects the data by the member sensor nodes in the cluster, and then

compresses, integrates and sends it to the sink node. The process of LEACH protocol as shown in Fig. 3.

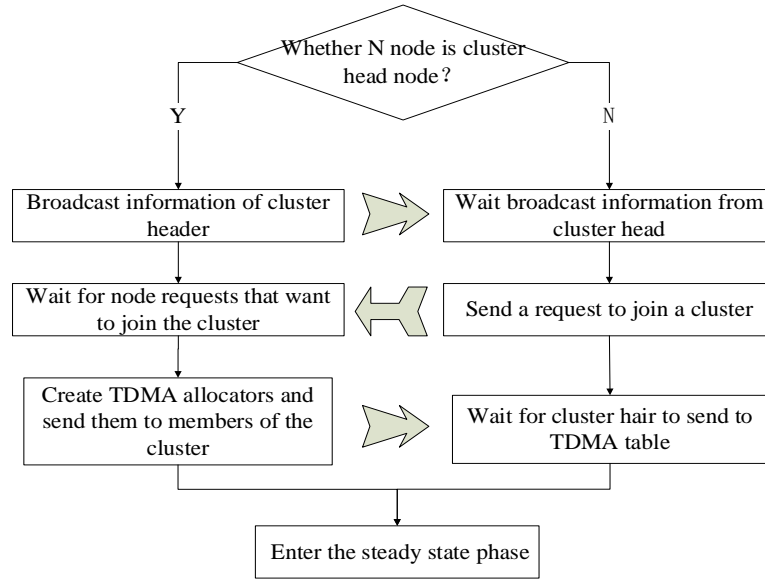


Fig. 3. The process of LEACH

Data transmission and data fusion are the main energy-consuming operations of LEACH, which can be divided into data transmission phase, data reception phase and the final data fusion phase. The free space model (the consumption is  $D^2$ ) and m-path attenuation model (the consumption is  $D^4$ ) are used in the data transmission phase to reduce the total consumption of wireless sensor network. The consumption can be expressed as:

$$E_{TX}(k, d) = \begin{cases} k \cdot E_{ele} + k \cdot \varepsilon_1 d^2 & d < D \\ k \cdot E_{ele} + k \cdot \varepsilon_2 d^4 & d \geq D \end{cases} \quad (2)$$

$E_{ele}$  is the consumption of transferring data,  $\varepsilon_1$  and  $\varepsilon_2$  are the unit of consumption for data transfer. And the  $D$  is the threshold value of distance.

The energy consumption of receiving  $k$ -bit data can be expressed as:

$$E_{RX} = k \cdot E_{elec} \quad (3)$$

$$E_{AX} = k \cdot E_{da} \quad (4)$$

$E_{elec}$  is the consumption of receiving one bit of data and  $E_{da}$  is the consumption of fusing the unit of data.

### 3.2 The Multi-objective LEACH Protocol

In LEACH, the total number of cluster head nodes determines how many clusters the entire network is divided, the number of clusters have an important impact on the overall energy. How to reduce the whole energy consumption of the network and ensure the balance of energy consumption of the nodes in the network is a multi-objective optimization problem. In this paper, the cluster-head node selection problem is transformed into the multi-objective optimization problem of the Euclidean distance between the all sensor nodes and sink node, and another objective is the distance from the clusters to sink node.

(1) The first objective  $f_1$ :

$$\min f_1 = \sum_{k=1}^m d_{k \sin k} \quad (5)$$

Where  $m$  represents the total number of sensor nodes,  $d_{k \sin k}$  represents the Euclidean distance from the node  $k$  to sink node.

(2) The second objective  $f_2$ :

$$\min f_2 = \sum_{t=1}^w \left( \sum_{i=1}^n d_{ih} + d_{k \sin k} \right) \quad (6)$$

Where  $w$  represents the total number of node clusters,  $n$  represents the total number of sensor nodes,  $d_{ih}$  represents the Euclidean distance from the member sensor node of the cluster  $i$  to the head of the cluster  $h$ .  $d_{k \sin k}$  represents the Euclidean distance from the cluster head  $h$  to the sink node.

## 4. Multi-objective Coupling Algorithm

### 4.1 Bat Algorithm

The bat algorithm is a new bionic optimization algorithm proposed by scholar Yang in 2010. The idea of the algorithm comes from the process of mimicking bat colonies to guide foraging behavior through echolocation. The basic model of bat algorithm is shown below.

(1) Initialize the bat individuals

Since the location of food is unknown, the individual positions  $x_i$  are randomly generated according to the maximum  $x_{\max}$  and minimum  $x_{\min}$  values of the region boundary.

$$x_i = x_{\min} + \phi_i \times (x_{\max} - x_{\min}) \quad (7)$$

Where  $i = 1, 2, \dots, n$ ,  $\phi_i$  is a random number between  $[0, 1]$ .

(2) Global search of the BA

All individuals generate the speed  $v_i^{t+1}$  and location  $x_i^{t+1}$  of generation  $t+1$  based on their current velocity  $v_i^t$ , location  $x_i^t$ , frequency  $f_i$  of generation  $t$  and optimal location  $x_{best}$  in the population.

$$f_i = f_{\min} + \beta_i \times (f_{\max} - f_{\min}) \quad (8)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_{best}) \cdot f_i \quad (9)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (10)$$

Where  $\beta$  is a random number between  $[0, 1]$ .  $f_{\min}$  and  $f_{\max}$  are upper and lower frequency.

(3) Local search of bat algorithm

$$x_{new} = x_{best} + \varepsilon \overline{A}^t \quad (11)$$

Where  $\varepsilon \in [-1, 1]$ ,  $\overline{A}^t$  represents the average loudness of all bats at generation  $t$ .

(4) Update of loudness and emission frequency

The loudness and emission frequency are updated with the bat individuals approach the target to more accurately locate the food location.

$$r_i^{t+1} = r_i [1 - \exp(-\gamma t)] \quad (12)$$



$$A_i^{t+1} = \alpha A_i^t \quad (13)$$

Where  $r_i$  indicates the emission frequency of current generation.  $\gamma$  and  $\alpha$  are the coefficients of the emission and loudness frequency, respectively, which are constants in  $[0, 1]$ .

## 4.2 Bacterial Foraging Optimization

Prof. K. M. Passino proposed an algorithm BFO [58] in 2001. BFO algorithm is a new bionic algorithm that imitates Escherichia Coli to devour food in human intestines. This algorithm simulates the behavior of bacterial colony, including three steps: chemotaxis, reproduction and dispersal.

(1) Chemotaxis: this operation mainly simulates the movement process of bacteria, including forward movement and steering movement, the operation is defined by the following formula:

$$x^i(n+1, o, p) = x^i(n, o, p) + C(m) \frac{\Delta(m)}{\sqrt{\Delta^T(m) * \Delta(m)}} \quad (14)$$

Where  $C(m)$  means the step size, and  $\frac{\Delta(m)}{\sqrt{\Delta^T(m) * \Delta(m)}}$  means the random direction of bacterias.

(2) Reproduction: this operation simulates the propagation process of the survival of the fittest of individual bacteria. It orders all bacteria according to the fitness value, removes half of the worse bacteria, replaces half of the superior bacteria, and ensures that the total amount of bacteria remains the same.

$$J_{health}^i = \sum_{n=1}^{N_c} f(x^m(n, o, p)) \quad (15)$$

(3) Dispersal: this operation is to improve the global search ability, because when there are multiple extremum points in the solution space, the clustering of bacteria will make the algorithm easily fall into local extremum, so the dispersal operation can disrupted cluster.

$$x = \begin{cases} x_{new}, & \text{if } q < p_{ed} \\ x, & \text{otherwise} \end{cases} \quad (16)$$

Where  $x_{new}$  denotes the new position obtained through initialization,  $q(0 < q < 1)$  is a uniformly distributed random number.

## 4.3 Glowworm Swarm Optimization

The idea behind the GSO algorithm is to use fluorescent communications to simulate firefly social behavior in nature. In addition, the standard GSO algorithm consists of four stages, namely, initialization, updating luciferin, position update and update the perceptual range stage. The five phase operation is detailed below.

(1) Initialize the glowworm population

The initial population of glowworms are randomly generated, and the initial luciferin of each glowworm is set to  $l_0$ .

(2) Update luciferin of the glowworm

The luciferin of the glowworm is depended on the position, in the solution space, the position of a glowworm represents a solution, the fitness value of the solution determines the



value of luciferin, the better the fitness value, the higher the luciferin value of the glowworm.

$$Y_m(c+1) = (1-\rho)Y_m(c) + \gamma J(x_m(c+1)) \quad (17)$$

In equation (17),  $l_m(c)$  means the size of luciferin, the luciferin value is  $\rho$  ( $0 < \rho < 1$ ),  $\gamma$  represent the luciferin enhancement factor,  $J(x_m(c+1))$  is the value of the objective function.

### (3) Roulette selection

In order to select individuals with better solutions, the algorithm adopts the roulette selection mechanism. During each iteration, the individual  $n_j$  is selected when the two conditions are satisfied. First, individual  $n_j$  exists in the perception radius of individual  $n_m$ ; secondly, the fluorescence value of individual  $n_j$  is also greater than that of individual  $n_m$ . The calculation equations are shown below:

$$G_{mn}(t) = \frac{Y_n(t) - Y_m(t)}{\sum_{k \in N_i(t)} Y_k(t) - Y_m(t)} \quad (18)$$

$$N_m(t) = \{n : d_{m,n}(t) < r_d^i; Y_m(t) < Y_n(t)\} \quad (19)$$

The  $d_{m,n}(t)$  is the Euclidean distance from the individual  $m$  to  $n$  in the  $t$  th iteration.

### (4) Updating position

The equation of updating position is defined as follows.

$$x_m(t+1) = x_m(t) + s \times \left( \frac{x_n(t) - x_m(t)}{\|x_n(t) - x_m(t)\|} \right) \quad (20)$$

Where  $d_{m,n}(t)$  is the position of individual  $m$  at time  $t$ ;  $\|\bullet\|$  represents the Euclidean norm operator and  $s(s > 0)$  is the step size.

### (5) Updating perception range

The formula for updating perception range as shown:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta \times (n_i - |N_i(t)|)\}\} \quad (21)$$

Where  $\beta$  is a constant,  $n_i$  is a coefficient.

## 4.4 The Multi-objective Coupling Algorithm based on BA, BFO and GSO

The traditional optimization algorithm has their limited ability to solve many practical problems, because the existing algorithms have their specific defects. On the basis of the no free lunch theorem, each kind of algorithm has its applicable problem, a certain algorithm cannot achieve satisfactory results in all issues. Every algorithm has their specific adaptation in a particular field. So, the coupling algorithm MBGF based on BA, GSO and BFO is proposed to remedy such problems. The design process of multi-objective coupling algorithm is as follows:

### (1) The pareto dominate relational

In single objective bacterial foraging algorithm, only one fitness value can be compared to determine the individual's superiority and inferiority. In the multi-objective optimization, the pareto dominance relationship needs to be used for comparison. Let the  $x^a = (x_1^a, x_2^a, \dots, x_D^a)^T$  and  $x^b = (x_1^b, x_2^b, \dots, x_D^b)^T$  be any individuals in the population. If and only if  $\forall i \in \{1, 2, \dots, n\} : x_i^a \leq x_i^b$  and  $\exists i \in \{1, 2, \dots, n\} : x_i^a < x_i^b$ , then  $x_i^a \preceq x_i^b$ . We adopt the normalization method to deal with it, if the individuals  $x^a$  and  $x^b$  do not have pareto dominance relationship. The process is as follows:

The proportion  $W$  of the two individuals in the fitness value was calculated respectively.

$$W_i = \frac{f_i}{f_i + f_j} \quad (22)$$

$$W_j = \frac{f_j}{f_i + f_j} \quad (23)$$

Then, the sum of the weights of the two individuals is calculated as follows:

$$F = \sum_{k=1}^M \delta_k \times |W_j - W_i| \quad (24)$$

Where  $\delta_k$  represents the weight of the objective,  $0 < \delta_k < 1$  and  $\sum_{k=1}^M \delta_k = 1$ .  $M$  represents the number of objective functions.

#### (2) Adaptive chemotaxis operator

In the chemotaxis operator, each update of the individual position is compared according to the pareto dominance relationship mentioned above. However, the original fixed step size cannot meet the requirements of convergence, so this paper gives a new definition of the original fixed step size, and the formula is as follows:

$$C_D = \frac{S_D}{j + k + l} \quad (25)$$

$S_D$  represents the initial step size on the  $D$  dimension ( $D=1,2,\dots,n$ ),  $j$  represents the current chemotactic times,  $k$  represents the current replication times, and  $l$  represents the current elimination and dispersal times.

#### (3) Simulated binary crossover

In the original BFO, the reproduction operator would sort individuals by their fitness values, and then delete half of the individuals with worse fitness values. However, in multi-objective, this operation will lead to a great reduction of population diversity, which is not conducive to the distribution of population diversity. So, this paper introduces a simulated binary crossover in order to maintain the diversity of population.

$$X'_{1j}(t) = 0.5 \times [(1 + \gamma_j) \times X_{1j}(t) + (1 - \gamma_j) \times X_{2j}(t)] \quad (26)$$

$$X'_{2j}(t) = 0.5 \times [(1 - \gamma_j) \times X_{1j}(t) + (1 + \gamma_j) \times X_{2j}(t)] \quad (27)$$

$$\gamma_j = \begin{cases} (2u_j)^{\frac{1}{\eta+1}}, & \text{if } u_j \leq 0.5 \\ (\frac{1}{2(1-u_j)})^{\frac{1}{\eta+1}}, & \text{otherwise} \end{cases} \quad (28)$$

Where  $u_j \in U(0, 1)$ ,  $\eta = 1$ .

#### (4) The dispersal operator based on polynomial mutation

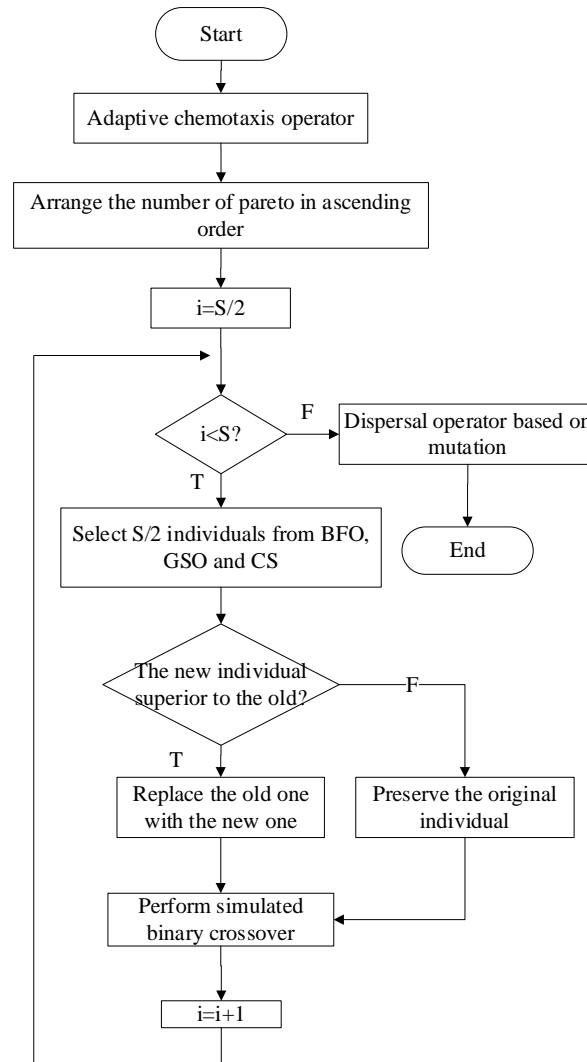
Generally speaking, the convergence speed and accuracy are only considered in the single-objective optimization algorithm, but in the multi-objective algorithm, we need to consider not only the convergence of the algorithm but also the diversity of the population. In the standard BFO algorithm, random generation is carried out for individuals satisfy the certain conditions. Although this method can improve the overall diversity to some extent, it does not make use of the convergence of the algorithm in the later stage of the multi-objective optimization. This paper introduces polynomial variation to less than dispersion probability of variation individual.

$$X_{lj}(t) = X_{lj}(t) + \beta_j \quad (29)$$

$$\beta_j = \begin{cases} (2u_j)^{\frac{1}{\eta+1}} - 1, & u_j < 0.5 \\ (1 - (2(1 - u_j))^{\frac{1}{\eta+1}}), & \text{other} \end{cases} \quad (30)$$

Where,  $u_j \in (0, 0.5)$ .

The whole algorithm no longer disperses individuals randomly through the above dispersal operator as before, but disperses individuals on a specific basis, which will be conducive to finding a better solution. **Fig. 5** shows the flow of the MBGF. The pseudo code of the MBGF algorithm is listed in **Algorithm 1**.



**Fig. 5.** The schematic flowchart of the multi-objective coupling Algorithm

#### 4.5 The Metrics of the Algorithms

Performance metrics is an important index to evaluate the performance of an algorithm. Different from single objective, not only the convergence rate of algorithm should be considered, but also the diversity of population should be considered in multi-objective optimization problem. Convergence generally refers to the ability of the algorithm to converge to the Pareto frontier, and diversity generally means that the population is distributed as evenly as possible on the Pareto frontier. There are three of the most popular algorithm performance indicators, GD (Generational short) [59] value, SP (Spacing) value [59] and IGD (Inverted Generational short) value [60]. GD is the Average minimum distance of each point in the solution set to the reference Pareto set  $P^*$ . The smaller the GD value, the better the convergence of the algorithms. Spacing is the standard deviation that measures the minimum distance between each solution point and the other solution points. The smaller the Spacing value, the more uniform the solution set. The IGD value is the average distance from each reference point to the nearest solution point. The smaller the IGD value, the better the comprehensive performance of the algorithm. In this paper, the performance evaluation adopts these three performance evaluation indexes.

$$GD = \frac{1}{n} \sqrt{\sum_{i=1}^n dist_i^2} \quad (31)$$

Where  $dist_i$  represents the minimum distance between the solution obtained by the algorithm and the pareto set,  $n$  represents the size of population.

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (32)$$

Where  $d_i$ , this is the distance between any two pareto solutions.  $\bar{d}$  is the average of  $d_i$ .

$$IGD(x, p^*) = \frac{\sum_{x \in p^*} dist(x, p)}{|P^*|} \quad (33)$$

Where  $p^*$  represents the set of points on the real pareto sets and requires uniform distribution.

**Table 1. Algorithm 1- MBGF Algorithm**

---

Step1: Initial population size $S$ , the position and other parameter;
Step2: let $j=0$ , The number of the dispersal operator $N_{ed}$ , $j=j+1$ ;
Step3: let $k=0$ , The number of the reproduction operator $N_{re}$ , $k=k+1$ ;
Step4: let $l=0$ , The number of the chemotaxis operator $N_c$ , $i=i+1$ ;
Step5: The chemotaxis operator:
I. For each dimension of each individual $i$ , the step size of each dimension is calculated according to formula (25);
II. Perform adaptive chemotaxis until each individual completes chemotaxis;
III. if $l < N_c$ , go back to Step4;
Step6: The reproduction operator:
I. The number of dominant and dominated for each individual are calculated according to Pareto relation, and then a fast non-dominant sort is carried out according to the dominated number of each individual;

---

---

```

    II. Select the first S/2 individuals, and delete the second S/2 individuals;
    III. The remaining S/2 individuals are selected from CS and GSO;
    IV. if  $k < N_{re}$ , go back to Step3;
Step7: The dispersal operator:
    I. The Simulated binary crossover and mutation process will be carried out with
    formula (26-29) When the individual reaches certain conditions.
    II. if  $j < N_{ed}$ , go back to Step2.
    Else
        Step8: Output the optimal solution set.
    end

```

---

## 5. Simulation Results and Analysis

### 5.1 Parameters and Experimental Environment

In experiments part, six multi-objective test functions are used to verify the performance of the algorithm. The test functions include SCH, ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6. These test functions are widely used in multi-objective optimization tests. SCH and ZDT1 functions have a convex Pareto front, ZDT2 has a smooth non-convex Pareto front, and ZDT3 consists of 5 slightly convex parts which constitute a discontinuous Pareto front. ZDT4 is a multimode function with multiple fronts and  $21^9$  different local fronts. ZDT6 has a non-convex front, which is very similar to ZDT2. All of the above problems have two objective functions and no constraints. In this experiment, we compared with the other three algorithms, a fast and elitist multi-objective genetic algorithm (NSGA-II), improved the strength pareto evolutionary algorithm (SPEA2) and pareto-optimal neighbor-based immune algorithm (PNIA). The experiments are performed in processor of Intel 5, memory of 4G and Matlab 2016a of Windows 7 environment. In the experiment, each algorithm will run independently on each test function for 30 times and iterate for 100 times. In addition, in order to the convenience of comparison, the population size of each algorithm in this section is uniformly set to 100, and other corresponding parameter Settings are shown in [Table 2](#).

**Table 2.** Parameters setting of the algorithms

Algorithm	Parameters
NSGA-II	$pc = 1, pm = \frac{1}{D}$
SPEA2	$pc = 1, pm = \frac{1}{D}$
PNIA	$pc = 1, pm = \frac{1}{D}$
MBGF	$N_{re}=4, N_{re}=4, N_{ed}=2, \beta=0.08, \rho=0.4, \gamma=0.6$

In order to improve the accuracy of the algorithm, we select the threshold value of the dispersal probability  $P_{ed}$  in BFO algorithm by golden ratio. The golden ratio is also known by many other names such as the golden proportion, golden mean, golden section, golden

number et al. In this experiment, we set parameters  $Ped \in [0,1]$ . The experiment will be completed If the experimental results are less than 0.01, and the experimental process is shown in [Table 3](#).

It can be seen from [Table 3](#), that the range of the interval becomes 0, meeting our previous termination condition and stopping the cycle when the method is carried out at the 9<sup>th</sup> time. At this time, the value  $Ped=0.206$  according to the experimental results.

**Table 3.** The results of  $Ped$  with golden section method

Number	$Ped$	Rank	abridged general view
1	0.000	2.79	
	0.382	1.89	
	0.618	2.18	
	1.000	3.14	
2	0.000	3.13	
	0.236	2.23	
	0.382	2.25	
	0.618	2.39	
3	0.000	3.05	
	0.146	3.02	
	0.236	1.91	
	0.382	2.02	
4	0.146	3.23	
	0.236	2.00	
	0.292	2.46	
	0.382	2.30	
5	0.146	3.18	
	0.202	2.11	
	0.236	2.18	
	0.292	2.54	
6	0.146	3.16	
	0.180	2.48	
	0.202	2.21	
	0.236	2.14	
7	0.202	2.43	
	0.215	2.34	
	0.223	2.61	
	0.236	2.63	
8	0.202	2.46	
	0.215	2.36	
	0.210	2.79	
	0.223	2.39	

9	0.202	2.00	
	0.206	1.84	
	0.206	1.84	
	0.210	2.16	

## 5.2 Comparative analysis of experimental results

In this section, in order to demonstrate the performance of the algorithms, four different algorithms were tested on the test set, and the experimental results are shown in [Table 4](#). In the [Table 4](#), mean(GD) value and menn(SP) value represent the average value obtained on the test set after 30 runs of each algorithm, while STD (GD) value and STD (SP) value represent the variance value, respectively.

**Table 4.** 4 Comparison results of GD and SP values of four different algorithms

Problems	Metrics	SPEA2	PNIA	NSGA-II	MBGF
SCH	mean(GD)	4.4126E-02	<b>3.8135E-02</b>	6.1711E-02	4.3512E-02
	std(GD)	<b>1.1456E-03</b>	1.9312E-03	2.9614E-03	3.2164E-02
	mean(SP)	7.0138E-03	<b>4.9647E-03</b>	1.7536E-02	1.3475E-02
	std(SP)	<b>1.6157E-03</b>	3.6638E-03	1.0234E-02	2.2137E-02
ZDT1	mean(GD)	2.8047E-05	6.5647E-04	2.5837E-04	<b>4.8936E-15</b>
	std(GD)	1.3450E-05	1.6468E-04	1.1750E-04	<b>1.2649E-14</b>
	mean(SP)	1.9426E-03	<b>1.0445E-03</b>	5.0953E-03	3.2539E-03
	std(SP)	3.3835E-04	<b>1.0654E-03</b>	3.9228E-03	7.3022E-03
ZDT2	mean(GD)	1.2700E-05	8.7648E-04	8.4237E-05	<b>2.5641E-14</b>
	std(GD)	1.2165E-05	1.1735E-03	1.3228E-04	<b>4.1038E-14</b>
	mean(SP)	<b>2.1343E-11</b>	2.3628E-03	2.8140E-03	8.3670E-04
	std(SP)	8.8300E-04	3.1781E-03	4.3652E-03	<b>1.2986E-03</b>
ZDT3	mean(GD)	4.4327E-06	1.8804E-04	1.4068E-04	<b>1.5397E-14</b>
	std(GD)	1.7122E-06	1.0106E-04	9.9837E-05	<b>2.9026E-14</b>
	mean(SP)	1.8927E-03	<b>4.0853E-04</b>	5.5670E-03	1.2427E-03
	std(SP)	5.9920E-04	<b>5.4742E-04</b>	3.8153E-03	1.5903E-03
ZDT4	mean(GD)	6.9563E-02	1.2526E-02	2.6805E-01	<b>1.1837E-04</b>
	std(GD)	4.1956E-02	<b>1.0338E-02</b>	1.3024E-01	2.3959E-02
	mean(SP)	4.3747E-03	1.4658E-03	5.0957E-03	<b>1.3680E-03</b>
	std(SP)	4.6448E-03	3.2069E-03	1.1158E-02	<b>1.5423E-03</b>
ZDT6	mean(GD)	1.3328E-03	1.7897E-03	5.4921E-03	<b>1.2379E-03</b>
	std(GD)	1.8487E-04	2.3154E-04	<b>1.2740E-03</b>	1.4587E-03
	mean(SP)	1.4538E-03	1.3569E-03	4.4326E-03	<b>4.6549E-04</b>
	std(SP)	<b>5.7432E-04</b>	9.8746E-04	2.0548E-03	2.0230E-03

As can be seen from [Table 4](#), in terms of convergence, we propose MBGF algorithm is superior to the NSGA-II, SPEA2 and PNIA algorithms in SCH, ZDT1, ZDT2, ZDT3 ZDT4 and ZDT6 test set. In terms of diversity, we propose the MBGF algorithms have good diversity except for ZDT1 and ZDT3. From ZDT1 and ZDT3, the PNIA algorithm shows good diversity.

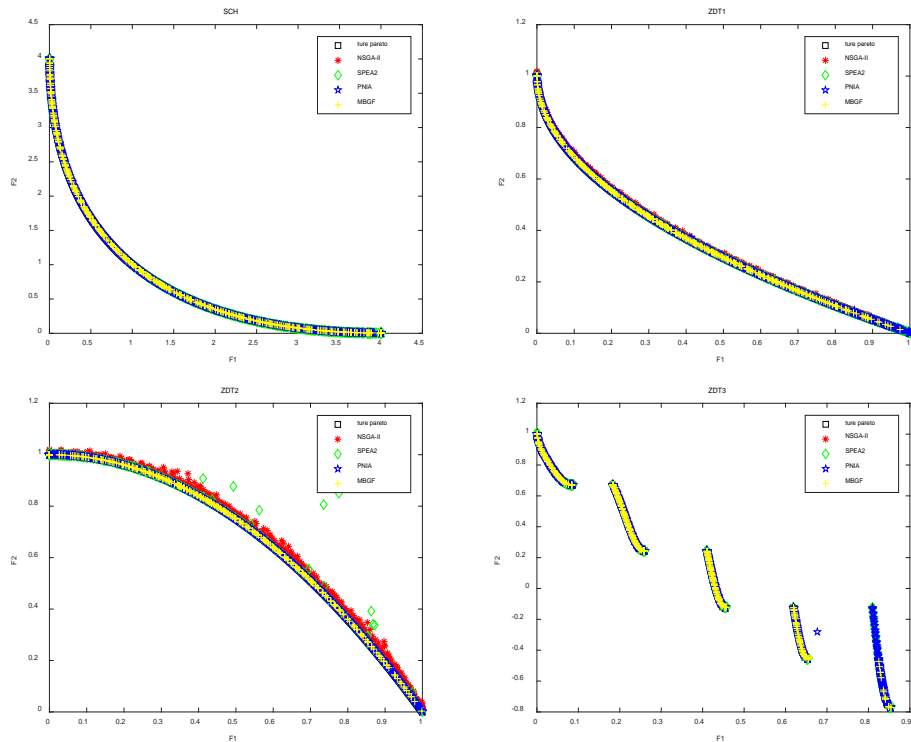


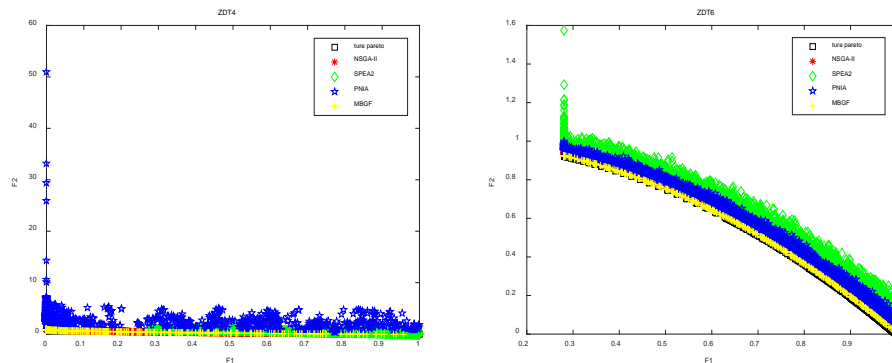
In order to further demonstrate the effectiveness of the MBGF algorithm, we also carried out these algorithms in IGD value. The experimental results are shown in [Table 5](#).

**Table 5.** 4 Comparison results of IGD values of four different algorithms

Problems	Metrics	SPEA2	PNIA	NSGA-II	MBGF
SCH	mean(IGD)	2.7448	1.5049	2.7305	<b>1.3564</b>
	Std(IGD)	12.2751	8.2428	12.2112	<b>7.9458</b>
ZDT1	mean(IGD)	0.6175	0.5368	0.5861	<b>0.4327</b>
	Std(IGD)	2.7615	2.9403	2.6210	<b>2.5421</b>
ZDT2	mean(IGD)	0.7002	0.5396	0.7706	<b>0.4076</b>
	Std(IGD)	3.1314	2.9515	3.4461	<b>2.5437</b>
ZDT3	mean(IGD)	<b>0.2384</b>	0.5388	0.2445	0.2964
	Std(IGD)	<b>1.0661</b>	2.9512	1.0981	2.3268
ZDT4	mean(IGD)	2.1738	1.2983	3.4946	<b>0.4374</b>
	Std(IGD)	9.7216	7.1113	15.6284	<b>2.5419</b>
ZDT6	mean(IGD)	0.6646	0.5026	0.6357	<b>0.4243</b>
	Std(IGD)	2.9720	2.7527	2.8431	<b>2.2304</b>

As can be seen from [Table 5](#), the algorithm proposed in this paper performs well in most test functions. This shows that our proposed algorithm has great performance improvement in both diversity and convergence. The true Pareto sets of the six test functions and the Pareto sets of four comparison algorithms on the six test sets are shown in [Fig. 6](#).





**Fig. 6.** The dynamic performance comparison results of four algorithms on six test sets

**Fig. 6** shows the true Pareto front of the six test functions and the Pareto sets of four comparison algorithms, where  $f_1$  and  $f_2$  are objective functions in the standard benchmarks. The converging to the true Pareto front and the diverse solutions in the front can be clearly seen in **Fig. 6**. For SCH, ZDT1, ZDT2, ZDT3 test functions, NSGA-II, SPEA2, PNI and MBGF indicate the similarity in convergence and distribution. It can be seen from the performance diagram of ZDT4 and ZDT6 that our algorithm can well converge to its real front surface, especially in ZDT4, NSGA-II and PNI algorithms may be trapped in local optimization and cannot well converge to the real front.

### 5.3 The multi-objective LEACH protocol Algorithm Based on MBGF

In order to verify the MBGF algorithm, We apply it to the multi-objective LEACH protocol by this paper proposed.

In this experiment part, the region to be monitored in the constructed wireless sensor networks model is set to  $100 \times 100$ , and the location of the base station is set to (50,50). In this model, some assumptions are adopted, that is, interference and delay are not considered in the experiment, and the base station has infinite energy. The parameters set in the experiment are shown in the following table:

**Table 6.** Experimental parameters settings

Node Numbers	100
Initial Energy for each Node	0.5J
Iterations	2000
The Packs of Data	4000bits
Consumption for Unit Data (Send or Receive)	50nJ/bit
Consumption for Fusion Unit Data	5nJ/bit
Probability of Cluster-head Node	0.05

In order to verify the reliability of our multi-objective LEACH protocol, we compare the LEACH protocols of the following algorithms:

- (1) The LEACH protocol(LEACH)
- (2) The LEACH protocol based on standard Bat Algorithm (SBA-LEACH)

- (3) The LEACH protocol based on Cuckoo Search(CS-LEACH)
- (4) The multi-objective LEACH protocol Algorithm based on MBGF(MBGF-LEACH)

This time we compared the number of surviving nodes and total residual energy of nodes. The number of surviving nodes for the four algorithms is shown in Fig. 7. We can clearly see from Fig. 7 that the performance of each algorithm is almost indistinguishable before 1000 rounds. When the algorithm turns from 1000 to 1500 rounds, the number of surviving nodes of each algorithm decreases rapidly, especially. However, our algorithm is more prominent than other surviving nodes after 1500 rounds.

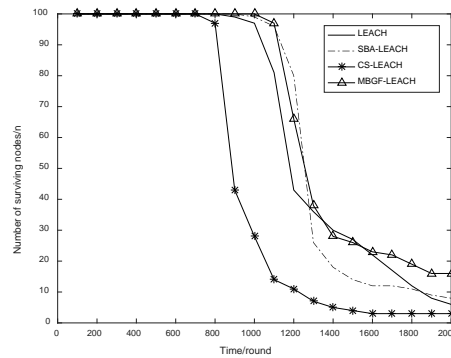


Fig. 7. Comparison of surviving nodes of each algorithm

Total residual energy of nodes is shown in Fig. 8. We can clearly see from Fig. 8 that the initial energy of each algorithm are the same. The residual energy of MBGF-LEACH can achieve a superior result than others protocol under the same number of rounds. Experimental results show that our algorithm saves the most energy among all the comparison algorithms.

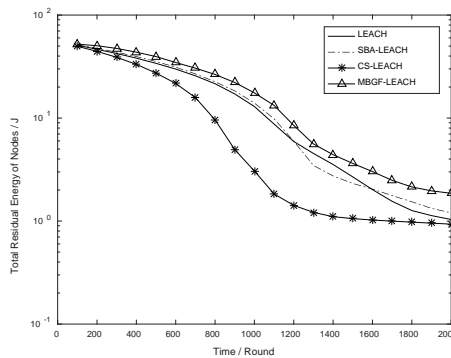


Fig. 8. Comparison of node residual energy

Table 7 represents the experimental data results for each LEACH protocol. After 2000 rounds, as for the number of Surviving Nodes, the MBGF-LEACH algorithm is the most, and the Total Remaining Energy in MBGF-LEACH is also much more than that in other protocols. Overall, the MBGF-LEACH algorithm is far better than the other three algorithms in the results, we can see that the multi-objective LEACH protocol proposed is valid.

In conclusion, experimental results show that our algorithm is effective in all comparison algorithms in terms of node inventory and residual energy.

**Table 7.** Comparison results of four protocols

Algorithm	Surviving Nodes	Total Remaining Energy
LEACH	6	1.035
SBA-LEACH	8	1.2013
CS-LEACH	7	1.1376
MBGF-LEACH	14	1.8024

## 5. Conclusion

As well known, the LEACH protocol belongs to cluster routing protocol in WSN. Many swarm intelligence optimization algorithms have been proposed to optimize the LEACH. However, the protocol does not specify how the number of cluster-head nodes should be distributed across the network. Therefore, it is likely that the cluster head nodes are concentrated in a certain area of the network. So that there are no cluster head nodes around some nodes, resulting in uneven distribution of network energy consumption. LEACH assumes that the first cluster-head round selects all the nodes with the same energy, and that each cluster-head node consumes roughly the same energy. Therefore, the protocol is not suitable for the network with unbalanced node energy. Aiming at this shortcoming, this paper proposes multi-objective LEACH protocol to improve performance of algorithm. In addition, in order to improve the performance of multi-objective algorithm, we also propose a multi-objective coupling algorithm based on BA, GSO and BFO Algorithm. Experimental results show that our proposed coupling algorithm greatly improves the performance of the algorithm. What's more, we also apply the multi-objective coupling algorithm to the multi-objective LEACH protocol, and the results show that the protocol greatly improves the number of surviving nodes and the total residual energy. In our future work, the different coupling algorithms and LEACH protocol are improved to solve the engineering problems, and the recently proposed protocols [61] are also comparing with our improved protocol.

## Acknowledgement

This work is sponsored by the Natural Science Foundation of Zhejiang Province under Grant NO.LY18F030010.

## References

- [1] D. Xin Ma, J. ma, P. Min Xu, and Y. Pang, "The Application Research Progress of Wireless Sensor Networks," *Applied Mechanics and Materials*, vol. 475-476, pp. 520-523, December, 2013. [Article \(CrossRef Link\)](#)
- [2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, August, 2008. [Article \(CrossRef Link\)](#)
- [3] W. Huang, Y. Wang, and H. Guan, "The Current Situation and Prospect of Localization in Wireless Sensor Network," in *Proc. of 2009 Second International Workshop on Computer Science and Engineering*, vol. 1, pp. 483-487, October, 2009. [Article \(CrossRef Link\)](#)
- [4] Z. Cui, L. Du, P. Wang, X. Cai, and W. Zhang, "Malicious code detection based on CNNs and multi-objective algorithm," *Journal of Parallel and Distributed Computing*, vol. 129, pp. 50-58, July, 2019. [Article \(CrossRef Link\)](#)
- [5] Y. Cao, Z. Cui, F. Li, C. Dai, and W. Chen, "Improved Low Energy Adaptive Clustering Hierarchy Protocol Based on Local Centroid Bat Algorithm," *Sensor Letters*, vol. 12, pp. 1372-1377, September, 2014. [Article \(CrossRef Link\)](#)

- [6] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210-219, December, 2017. [Article \(CrossRef Link\)](#)
- [7] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, "Robust Graph Learning From Noisy Data," *IEEE Transactions on Cybernetics*, pp. 1-11, December, 2019. [Article \(CrossRef Link\)](#)
- [8] X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang, and J. Chen, "Multi-objective 3-Dimensional DV-Hop Localization Algorithm with NSGA-II," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10003-10015, 2019. [Article \(CrossRef Link\)](#)
- [9] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51-58, May, 2000. [Article \(CrossRef Link\)](#)
- [10] A. Ray and D. De, "Energy efficient clustering protocol based on K-means (EECPK-means)-midpoint algorithm for enhanced network lifetime in wireless sensor network," *IET Wireless Sensor Systems*, vol. 6, no. 6, pp. 181-191, December, 2016. [Article \(CrossRef Link\)](#)
- [11] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "Smart Dust," in *Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, Washington, USA, pp. 271-278, August, 1999. [Article \(CrossRef Link\)](#)
- [12] Y. Zeng and L. Wang, "Energy-saving routing protocol for Wireless Sensor Networks," in *Proc. of The 26th Chinese Control and Decision Conference (2014 CCDC)*, pp. 4868-4872, June, 2014. [Article \(CrossRef Link\)](#)
- [13] X. Cai, Y. Sun, Z. Cui, W. Zhang, and J. Chen, "Optimal LEACH Protocol with Improved Bat Algorithm in Wireless Sensor Networks," *KSI Transactions on Internet and Information Systems*, vol. 13, no. 5, pp. 2469-2490, May, 2019. [Article \(CrossRef Link\)](#)
- [14] S. K. Singh, P. Kumar, and J. P. Singh, "A Survey on Successors of LEACH Protocol," *IEEE Access*, vol. 5, pp. 4298-4328, February, 2017. [Article \(CrossRef Link\)](#)
- [15] Z. Cui, Y. Cao, X. Cai, J. Cai, and J. Chen, "Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 217-229, October, 2019. [Article \(CrossRef Link\)](#)
- [16] I. Sahmoudi, "Arabic language and knowledge reduction in formal contexts," *International Journal of Computing Science and Mathematics*, vol. 10, no. 1, pp. 71-82, January, 2019. [Article \(CrossRef Link\)](#)
- [17] L. Ma, X. Wang, H. Shen, and M. Huang, "A novel artificial bee colony optimiser with dynamic population size for multi-level threshold image segmentation," *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 32-44, February, 2019. [Article \(CrossRef Link\)](#)
- [18] E. Amiri and M. N. Dehkordi, "Dynamic data clustering by combining improved discrete artificial bee colony algorithm with fuzzy logic," *International Journal of Bio-Inspired Computation*, vol. 12, no. 3, pp. 164-172, September, 2018. [Article \(CrossRef Link\)](#)
- [19] G. Saranya, H. K. Nehemiah, and A. Kannan, "Hybrid particle swarm optimisation with mutation for code smell detection," *International Journal of Bio-Inspired Computation*, vol. 12, no. 3, pp. 186-195, September, 2018. [Article \(CrossRef Link\)](#)
- [20] Z. Cui, Y. Chang, J. Zhang, X. Cai and W. Zhang, "Improved NSGA-III with selection-and-elimination operator," *Swarm and Evolutionary Computation*, vol. 49, pp. 23-33, September, 2019. [Article \(CrossRef Link\)](#)
- [21] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, November, 1999. [Article \(CrossRef Link\)](#)
- [22] J. Hu, G. Yu, J. Zheng, and J. Zou, "A preference-based multi-objective evolutionary algorithm using preference selection radius," *Soft Computing*, vol. 21, no. 17, pp. 5025-5051, September, 2017. [Article \(CrossRef Link\)](#)
- [23] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, November, 2007. [Article \(CrossRef Link\)](#)

- [24] J. D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proc. of the 1st International Conference on Genetic Algorithms*, pp. 93-100, 1985.  
[Article \(CrossRef Link\)](#)
- [25] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *TIK-Report*, vol. 103, pp. 1-21, January, 2001. [Article \(CrossRef Link\)](#)
- [26] Z. Cui et al., "A pigeon-inspired optimization algorithm for many-objective optimization problems," *Science China Information Sciences*, vol. 62, no. 7, p. 70212, January, 2019.  
[Article \(CrossRef Link\)](#)
- [27] G. Zhou, R. Zhao, and Y. Zhou, "Solving large-scale 0-1 knapsack problem by the social-spider optimisation algorithm," *International Journal of Computing Science and Mathematics*, vol. 9, no. 5, pp. 433-441, September, 2018. [Article \(CrossRef Link\)](#)
- [28] S. Sahoo and M. Pal, "Modular and homomorphic product of intuitionistic fuzzy graphs and their degree," *International Journal of Computing Science and Mathematics*, vol. 8, no. 5, pp. 395-404, November, 2017. [Article \(CrossRef Link\)](#)
- [29] X. Cai, J. Zhang, H. Liang, L. Wang, and Q. Wu, "An Ensemble Bat Algorithm for Large-scale Optimization," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 10, pp. 3099–3113, 2019. [Article \(CrossRef Link\)](#)
- [30] G. Wang, X. Cai, Z. Cui, G. Min, and J. Chen, "High Performance Computing for Cyber Physical Social Systems by Using Evolutionary Multi-Objective Optimization Algorithm," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 20-30, 2020.  
[Article \(CrossRef Link\)](#)
- [31] X. Cai, S. Geng, D. Wu, L. Wang, and Q. Wu, "A unified heuristic bat algorithm to optimize the LEACH protocol," *Concurrency and Computation: Practice and Experience*, 2019.  
[Article \(CrossRef Link\)](#)
- [32] M. Zhang, H. Wang, Z. Cui, and J. Chen, "Hybrid multi-objective cuckoo search with dynamical local search," *Memetic Computing*, vol. 10, no. 2, pp. 199-208, 2018. [Article \(CrossRef Link\)](#)
- [33] Z. Cui, B. Sun, G. Wang, Y. Xue, and J. Chen, "A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems," *Journal of Parallel and Distributed Computing*, vol. 103, pp. 42-52, May, 2017. [Article \(CrossRef Link\)](#)
- [34] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, no. 2, pp. 87-124, June, 2009. [Article \(CrossRef Link\)](#)
- [35] H. Gao, Y. Du, and M. Diao, "Quantum-inspired glowworm swarm optimisation and its application," *International Journal of Computing Science and Mathematics*, vol. 8, no. 1, pp. 91-100, Mar, 2017. [Article \(CrossRef Link\)](#)
- [36] W. Yechuang, Z. Cui, and L. Wuchao, "A Novel Coupling Algorithm Based on Glowworm Swarm Optimization and Bacterial Foraging Algorithm for Solving Multi-Objective Optimization Problems," *Algorithms*, vol. 12, no. 3, March, 2019. [Article \(CrossRef Link\)](#)
- [37] Z. Cui, M. Zhang, H. Wang, X. Cai and W. Zhang, "A hybrid many-objective cuckoo search algorithm," *Soft Computing*, vol. 23, no. 21, pp. 10681–10697, April, 2019.  
[Article \(CrossRef Link\)](#)
- [38] Y. Wang et al., "A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization," *Mathematics*, vol. 7, p. 135, February, 2019. [Article \(CrossRef Link\)](#)
- [39] X. Cai et al., "An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search," *Concurrency and Computation Practice and Experience*, vol. 32, no. 5, 2020. [Article \(CrossRef Link\)](#)
- [40] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, USA*, pp. 174-185, August, 1999. [Article \(CrossRef Link\)](#)
- [41] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowledge-Based Systems*, vol. 163, pp. 510-517, January, 2019. [Article \(CrossRef Link\)](#)



- [42] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu, "Clustering with similarity preserving," *Neurocomputing*, vol. 365, pp. 211-218, November, 2019. [Article \(CrossRef Link\)](#)
- [43] Z. Wang and J. Dai, "Separated vehicle scheduling optimisation for container trucking transportation based on hybrid quantum evolutionary algorithm," *International Journal of Computing Science and Mathematics*, vol. 8, no. 5, pp. 405-413, November, 2017. [Article \(CrossRef Link\)](#)
- [44] P. Li, J. Zhao, Z. Xie, W. Li, and L. Lv, "General central firefly algorithm based on different learning time," *International Journal of Computing Science and Mathematics*, vol. 8, no. 5, pp. 447-456, November, 2017. [Article \(CrossRef Link\)](#)
- [45] X. Zhang, X.-t. Li, and M.-h. Yin, "Hybrid cuckoo search algorithm with covariance matrix adaption evolution strategy for global optimisation problem," *International Journal of Bio-Inspired Computation*, vol. 13, no. 2, pp. 102-110, March, 2019. [Article \(CrossRef Link\)](#)
- [46] X. Cai, X.-z. Gao, and Y. Xue, "Improved bat algorithm with optimal forage strategy and random disturbance strategy," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 4, pp. 205-214, August, 2016. [Article \(CrossRef Link\)](#)
- [47] T. Alam and Z. Raza, "Batch scheduling model for distributed systems," in *Proc. of 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 79-83, December, 2016. [Article \(CrossRef Link\)](#)
- [48] J. Fan, Y. Li, L. Yu Tang, and G. Kun Wu, "RoughPSO: rough set-based particle swarm optimisation," *International Journal of Bio-Inspired Computation*, vol. 12, p. 245, January, 2018. [Article \(CrossRef Link\)](#)
- [49] Z. Cui, F. Xue, X. Cai, Y. Cao, G. Wang, and J. Chen, "Detection of Malicious Code Variants Based on Deep Learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187-3196, July, 2018. [Article \(CrossRef Link\)](#)
- [50] H. Liang, Y. Liu, F. Li, and Y. Shen, "A multiobjective hybrid bat algorithm for combined economic/emission dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 101, pp. 103-115, October, 2018. [Article \(CrossRef Link\)](#)
- [51] A. Latif, I. Ahmad, P. Palensky, and W. Gawlik, "Multi-objective reactive power dispatch in distribution networks using modified bat algorithm," in *Proc. of 2016 IEEE Green Energy and Systems Conference (IGSEC)*, pp. 1-7, November, 2016. [Article \(CrossRef Link\)](#)
- [52] W. Chen and W. Xu, "A Hybrid Multiobjective Bat Algorithm for Fuzzy Portfolio Optimization with Real-World Constraints," *International Journal of Fuzzy Systems*, vol. 21, no. 1, pp. 291-307, February, 2019. [Article \(CrossRef Link\)](#)
- [53] B. Wu, C. Qian, W. Ni, and S. Fan, "The improvement of glowworm swarm optimization for continuous optimization problems," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6335-6342, June, 2012. [Article \(CrossRef Link\)](#)
- [54] Z. Tang, Y. Zhou, and X. Chen, "An Improved Glowworm Swarm Optimization Algorithm Based on Parallel Hybrid Mutation," in *Proc. of ICIC 2013: Intelligent Computing Theories and Technology, Berlin, Heidelberg*, pp. 198-206, July, 2013. [Article \(CrossRef Link\)](#)
- [55] Y. Chen and W. Lin, "An Improved Bacterial Foraging Optimization," in *Proc. of 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guilin, China, pp. 2057-2062, January, 2010. [Article \(CrossRef Link\)](#)
- [56] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy," *Applied Intelligence*, vol. 49, no. 4, pp. 1283-1305, April, 2019. [Article \(CrossRef Link\)](#)
- [57] N. A. Okaeme and P. Zanchetta, "Hybrid Bacterial Foraging Optimization Strategy for Automated Experimental Control Design in Electrical Drives," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 668-678, May, 2013. [Article \(CrossRef Link\)](#)
- [58] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52-67, August, 2002. [Article \(CrossRef Link\)](#)
- [59] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," *Cellular Immunology*, vol. 37, no. 1, pp. 1-13, August, 1995. [Article \(CrossRef Link\)](#)



- [60] A. Mohammadi, M. N. Omidvar, and X. Li, "A new performance metric for user-preference based multi-objective evolutionary algorithms," in *Proc. of 2013 IEEE Congress on Evolutionary Computation*, pp. 2825-2832, June, 2013. [Article \(CrossRef Link\)](#)
- [61] A. Paul, A. Banerjee, and S. P. Maity, "Residual Energy Maximization in Cognitive Radio Networks With Q-Routing," *IEEE Systems Journal*, pp. 1-10, 2019. [Article \(CrossRef Link\)](#)



**Wuzhao Li** received his PhD in Control Science and Engineering from Tongji University, China in 2018, and MSc in Automation Device from Jiangnan University in 2006. His research interest includes computational intelligence, intelligent control, and complex optimization.



**Yechuang Wang** received his M.D. in Complex System and Computational Intelligence Laboratory at Taiyuan University of Science and Technology, China. He is now an algorithm engineer at XCMG.



**Youqiang Sun** is an academic master of Computer Science and Technology from Taiyuan University of Science and Technology, China. His research interest includes computational intelligence and deep learning.



**Jie Mao** received her BSc in Liaoning University of Technology, she is a professor of Mechanical design, Manufacturing and Automation. Her research interest includes intelligent manufacturing, intelligent control and manufacturing execution system.