An Artificial Bee Colony-Based Green Routing Mechanism in WBANs for Sensor-Based E-Healthcare Systems

Jian Yan [ID], **Yuhuai Peng ***, **Dawei Shen, Xinxin Yan and Qingxu Deng**

College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China; jianyan@stumail.neu.edu.cn (J.Y.); shendw@stumail.neu.edu.cn (D.S.); 1310387@stu.neu.edu.cn (X.Y.); dengqx@mail.neu.edu.cn (Q.D.)

* Correspondence: pengyuhuai@mail.neu.edu.cn

Abstract: At present, sensor-based E-Healthcare systems are attracting more and more attention from academia and industry. E-Healthcare systems are usually a Wireless Body Area Network (WBANs), which can monitor or diagnose human health by placing miniaturized, low-power sensor nodes in or on patient's bodies to measure various physiological parameters. However, in this process, WBAN nodes usually use batteries, and especially for implantable flexible nodes, it is difficult to accomplish the battery replacement, so the energy that the node can carry is very limited, making the efficient use of energy the most important problem to consider when designing WBAN routing algorithms. By considering factors such as residual energy of node, the importance level of nodes, path cost and path energy difference ratios, this paper gives a definition of Optimal Path of Energy Consumption (OPEC) in WBANs, and designs the Optimal Energy Consumption routing based on Artificial Bee Colony (ABC) for WBANs (OEABC). A performance simulation is carried out to verify the effectiveness of the OEABC. Simulation results demonstrate that compared with the genetic algorithm and ant colony algorithm, the proposed OEABC has a better energy efficiency and faster convergence rate.

Keywords: E-Healthcare system; wireless body area networks; energy consumption; Artificial Bee Colony Algorithm

1. Introduction

Sensor-based E-Healthcare systems are a new technology developed in the field of health care in recent years. Wireless Body Area Networks (WBANs) [1] are mainly used in emergency rescue, telemedicine, home care and other occasions. Medical sensor nodes are worn on the body surface or implanted in the human body to monitor the patient's medical data, these physiological data are transferred to a sink through the WBAN, and the sink processes the data or sends the data to a medical monitoring center [2]. The nodes are mostly powered by microbatteries, the energy is very limited, so if some nodes are frequently used for data forwarding, it is likely to lead to the early depletion of these nodes, affecting the network connectivity, and eventually leading to a reduced global network lifetime [3–5]. Compared with the traditional data-centric WSN, WBANs' human-centered features make their energy requirements more demanding [6] is several aspects: (1) Faster signal transmission attenuation: the specificity of the human tissue structure and the shadow effects make the signal in the transmission process have a great path loss [7], resulting in WBANs' communication energy consumption being much higher than that of other normal networks of the same size. (2) Varying network communication link time: the network topology of WBANs is closely related to the changes in

the posture of the human body (especially the activities of the human limbs). The data retransmissions caused by the frequent interruptions of the communication links and the reconstruction of the network topology can cause a waste of energy [8]. (3) Limited node energy: The network lifetime in WBANs is defined as the time interval between when the network starts working to the time when the first node dies, which leads to network partitioning in such a way that the destination cannot be reached. As battery replacement and charging is not feasible in implant medical devices, each node must use limited energy to maximize its own life. Network lifetime is of more importance in WBANs compared to WSNs. Since about 80% of the total energy is consumed only for communication purposes [9], routing algorithms play a vital role to make the communication effective and prolong the lifetime of the WBANs, and how to use the residual energy of nodes effectively to extend the network lifetime becomes the key problem of designing WBAN routing algorithms. In the survey of routing protocols in wireless sensor networks [10,11], the path selection algorithm is based on the principle of minimizing the total energy consumption of the path, the global optimal solution is solved by using the greedy algorithm, which does not consider the protection of individual nodes with small energy, nor the energy consumption ratio of the whole path. It is easy to for some nodes to be utilized with high frequency, and if the important nodes suffer premature death, this often leads to the paralysis of the entire network [12]. Moreover, the time and spatial complexity of these algorithms will usually increase exponentially as the size of the problem grows. In the past decade, the group intelligent optimization methods have attracted more and more attention [13], as they are effective tools to address the problem of combinatorial optimization.

In this paper, we fully consider the energy consumption of the path, the residual energy of the nodes, the importance level of the nodes and the energy ratio of the whole path, and give the definition of Optimal Path of Energy Consumption (OPEC), and by means of the idea of artificial bee colony [14–16], design Optimal Energy Consumption Artificial Bee Colony Algorithm-Based WBANs (OEABC). The simulation results show that the OEABC converges faster, and has a better ability to find the optimal solution.

2. Related Works

In recent years, several routing protocols have been proposed for WBANs. According to the different routing mechanisms, the energy efficient strategy of WBANs in the network layer is mainly divided into single hop routing, multi-hop routing and cooperative routing. Early research on WBANs mainly focused on the construction of end-to-end network structures, and energy consumption was not a primary concern, so the data was transmitted by using star topologies and single-hop communication modes. The multi-hop routing protocol was proposed and divided into cluster-based routing and cross-layer routing. AnyBody [17], and HIT [18,19] are cluster-based routing protocols that after dividing the nodes into multiple clusters and selecting a cluster head, forward the data to the sink through the cluster head. These protocols aim to minimize the number of direct transmissions from nodes to the sink. However, the large amount of overhead required for cluster selection is the main drawback of these protocols. WASP [20], CICADA [21,22] are cross-layer routing protocols, that by controlling the timing of sending data for every node, lower the energy consumption of WBANs, but require exchanging a lot of control information between nodes. After weighing the advantages and disadvantages of single hop routing and multi-hop routing, collaborative routing [23–25] became widely used because it can distribute the transmission load of the whole network. According to the works presented in [23,24], the distance between the source node and sink node can be divided into high and low level. When a source node is located at a high level, multiple hops routing will be used. Otherwise, when it located at a low level, it will choose one hop routing, so the life-cycle of the whole network will be greatly improved. The authors in [25] proposed a decode-and-forward (DF) protocol which uses half duplex mode and builds two communication channels (S-D and S-R-D) between the source node and the sink node. Simulations show this protocol can effectively reduce the energy waste caused by packet retransmission. The pairs of different routing mechanisms are summarized in Table 1.

Table 1. Comparative Analysis of Different Routing Mechanisms.

Routing Protocol	Routing Mechanism	Advantage	Disadvantage	Energy Consumption
Star topology	Single Hop	Low delay	High energy consumption	High
AnyBody [17]	Cluster-Based Multi Hop	Reliable, low load	More complex	Medium
HIT [18,19]	Cluster-Based Multi Hop	Reliable, low load	More complex	Medium
WASP [20]	Cross-Layer Multi Hop	Reliable, low load	Large network delay	Medium
CICADA [21,22]	Cross-Layer Multi Hop	Reliable, low load	Large network delay	Medium
[23–25]	Collaborative	Low packet loss rate and low network delay	Unable to adapt to frequent network topology changes	Low

Certainly, there are many other routing mechanisms. Opportunistic routing [26], PSR [27], PRPL [28], OBSFR [29] are cost-effective routing protocols which select the lowest cost path by monitoring the cost of energy consumption for each path, so the cost-effective relationships between all nodes need to be periodically updated and stored, and a large amount of transmissions and overhead is required to find routes, which also adds complexity to the system. LOCALMOR [30], DMQOS [31] are based on QoS [32] routing protocols, according to the data type collected by the node to select different path metrics to meet the node's demand for data transmission reliability, delay and energy consumption, but QoS routing protocols require too much information that leads to high energy consumption and huge overhead. TARA [33], LTR [34], LTRT [35] are temperature-based protocols, which are mainly designed to minimize the local or overall system temperature rise. The idea behind these protocols is to route data along different paths to avoid a dramatic temperature rise in some nodes leading to human tissue damage and depletion of the node. In fact, each classification of routing protocols only tries to satisfy a specific requirement in WBANs. Moreover, when choosing a path, the global network lifetime [36] is considered to be insufficient, and the greedy algorithm is used when selecting the path, which easily falls into a local optimum.

3. Network Model

3.1. Mathematical Description of WBANs

Definition 1. *The set $W = \{V, C, \alpha, E, A\}$ represents WBANs, where $V = \{v_0, v_1, v_2, \ldots, v_n\}$ is the set of nodes; $C = \{c_0, c_1, c_2, \ldots, c_n\}$ is the set of residual energy of nodes, $c_i \geq 1$; $\alpha = \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n\}$ is the set of importance level of nodes, $\alpha \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$; $E = \{e_0, e_1, e_2, \ldots, e_m\}$ is the edge sets, $E \subseteq \{< i, j > | i, j = 0, 1, 2, \ldots, n\}$; A is the adjacency matrix of nodes, when $< i, j > \in E$, a_{ij} represents the energy required to transmit a packet from node i to node j, when $< i, j > \notin E$, $a_{ij} = 0$.*

3.2. The Important Level Factor α

There are many types of medical sensor collection activities. Some are essential, and must always be guaranteed, such as ECGs or blood pressure; some can be experienced interruption or delay, such as body temperature; and some nodes even only have forwarding functions, without the acquisition function. According to these characteristics, the importance level factor α of the node is defined, so the higher the importance level of the node, the greater the factor α. The maximum α is 9, and the lowest is 1.

In Figure 1 [29], the WBANs has 15 various types of nodes, the node set V is $\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}\}$, where v_0 is EEG, v_1 is hearing aid cochlear implant, v_2 is positioning, v_3 is motion sensor, v_4 is blood pressure, v_5 is Blood pump ECG, v_6 is insulin injection, v_7 is glucose, v_8 is blood oxygen, v_9 is lactic acid, v_{10} is artificial knee1, v_{11} is artificial knee2, v_{12} is pressure sensor1, v_{13} is pressure sensor2, v_{14} is sink [30]. The initial value of the residual energy of all nodes is 50, the node's important level factor α varies with the sensor node, and the corresponding network topology as shown in Figure 2, where each node includes the node name, the residual energy and the importance level factor α. The number on the link between nodes is the transmission energy consumption.
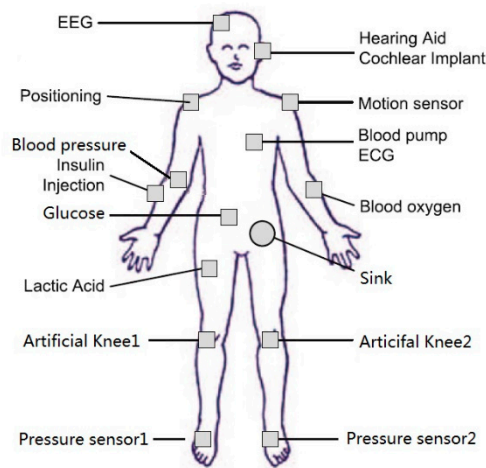
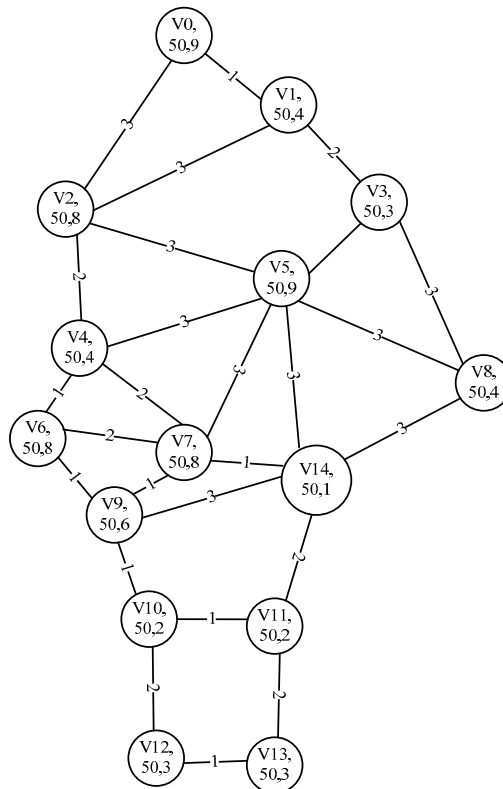**Figure 1.** Distributed sample of wireless body area network sensors.

Figure 2. The network topology of Figure 1.

4. The Optimal Path of Energy Consumption (OPEC)

4.1. Mathematical Description of Path Energy Consumption

Definition 2. *Path Set, denoted as P. The source node is v_0, the destination node is v_n, the alternate sequence v_0 ... $v_i v_j$ ... v_n is a path from v_0 to v_n, let P= {r_1, r_2, r_3, ... , r_k}, so P is the collection of all paths from v_0 to v_n.*

Definition 3. *Path Transmission Energy Consumption, denoted as E. The power consumption required to transfer a packet through r_i is formulated as:*

$$E(r_i) = \sum_{0}^{n} a_{ij} \tag{1}$$

The larger the $E(r_i)$, the greater the energy required to transfer the data using the path, and vice versa.

Definition 4. *Path Cost, denoted as C. The cost of the node v_i is formulated as:*

$$f(v_i) = \frac{1}{c_i} \qquad (2)$$

The cost of the path r_i is the sum of the cost of all nodes in the path and is formulated as:

$$C(r_i) = \sum_{i=0}^{n} f(v_i) \qquad (3)$$

With the decrease of the residual energy of the node, the cost of transmitting the data of the node will be increased, the $C(r_i)$ when the path r_i containing the node v_i also increases, and vice versa.

Definition 5. *Path Energy Difference Ratio, denoted as η. In the nodes of path r_i, the node with the largest remaining energy is max and the node with the smallest energy is min, the path r_i energy difference ratio η is formulated as:*

$$\eta(r_i) = \frac{c_{\min}}{\alpha_{\min} \times c_{\max}} \qquad (4)$$

$\eta(r_i)$ indicates the degree of energy difference between the maximum energy node v_{max} and the minimum node v_{min} in the path r_i, and the larger the η, the smaller the difference is, the better the energy consumption of the path r_i, otherwise the worse. $\alpha_{\min}$ is the importance level factor of v_{min}, used to adjust η, when v_{min} is very important or the energy will be exhausted, you can increase $\alpha_{\min}$, making η smaller. At the same time, in order to protect the node which residual energy has reached its may easily cause some nodes to run out of energy, the path will not be utilized.

4.2. Definition of the OPEC

The OPEC is denoted by r_{opt}, choosing r_{opt} can use the energy of nodes reasonably and efficiently, so that the energy of the whole network can be steadily reduced and the survival time of the whole network can be prolonged. Therefore, when calculating r_{opt} we need to consider the energy difference ratio η, path cost C and transmission energy E, but the weight of each parameter is different. First we consider η, select the maximum path of η from all paths that conform to $\eta_r > \eta_{Th}$, and try not to use less energy or very important nodes to transmit data. If there are multiple paths with the same η, the path with the lowest C is selected, to reduce the usage of nodes with less residual energy. If there are still multiple paths with the same η and C, select the path with the lowest E.

Definition 6. *Calculation equation of the OPEC:*

$$r_{opt} = \begin{cases} \max(\eta(r_i) | \eta_{r_1} \neq \eta_{r_2} \neq \ldots \neq \eta_{r_k}) \\ \min(C(r_i) | \eta_{r_1} = \eta_{r_2} = \ldots = \eta_{r_k}) \\ \min(E(r_i) | (\eta_{r_1} = \eta_{r_2} = \ldots = \eta_{r_k}) || (C_{r_1} = C_{r_2} = \ldots = C_{r_k})) \end{cases} \qquad (5)$$

5. Introduction to the Artificial Bee Colony (ABC) Algorithm

The ABC algorithm is inspired by the behavior of bee colonies, which is an optimization method mimicking the behavior of bees, proposed by the Karaboga team in 2005 to optimize the algebra problem [14]. It is an application of cluster intelligence concepts. Through the local optimization of artificial bees, the global optimal value emerges. In the ABC algorithm, the colony of artificial bees contains three groups of bees: worker bees, onlookers and scouts. A bee waiting on the dance

area for making decision to choose a food source, is called an onlooker and a bee going to the food source visited by itself previously is named a worker bee, a bee carrying out random search is called a scout. The position of a food source represents a possible solution of the optimization problem and the amount of nectar of a food source corresponds to the quality (fitness) of the associated solution. In the first step, the ABC generates a randomly distributed initial population of *SN* solutions, where *SN* denotes the size of the population. Each solution X_i (i = 1, 2, ... , *SN*) is a D-dimensional vector. Here, D is the number of optimization parameters. In order to produce an initial food position, the ABC uses the following expression:

$$x_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{6}$$

where $x_{\max}$ and $x_{\min}$ are the upper and lower bounds of search space.

After initialization, the worker bee will start the neighborhood search. It generates a new candidate position based on the local information in its memory and checks the amount of nectar in the new location. If the new location is better than the original position, the bee remembers the new location and forgets the original location. The new candidate position equation is:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{7}$$

The above is called the ABC algorithm search equation, where $k \in \{1, 2, \ldots, BN\}$, $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $\phi_{i,j}$ is a random number between $[-1, 1]$.

After the worker bees complete the search process, they will share the memory of the nectar with onlookers through the dance. An onlooker bee chooses a food source depending on the probability p_i associated with that food source, p_i is calculated by the following expression:

$$p_i = \frac{fit_i}{\sum\limits_{j=1}^{SN} fit_j} \tag{8}$$

where fit_i is the fitness value of the solution *i* evaluated by its employed bee. When a nectar source is selected, the onlooker bee will use Equation (8) to produce a new position. By checking the amount of nectar in the new position, if the position is better than the one in memory, it will be replaced, otherwise the original position is left. In order to prevent the algorithm from falling into a local optimum, if the amount of nectar does not improve after the limit cycle, the colony will abandon the position, whereby the worker bee becomes a scout and randomly generates a new nectar according to Equation (7).

6. The Optimal Energy Consumption Artificial Bee Colony-Based WBAN (OEABC)

6.1. The Fitness Function of the Path

The general algebra problem can construct the fitness function directly according to the requirement of the problem. It can be found that the optimal path of energy needs to consider the Path Energy Difference Ratio η, Path Cost C, Path Transmission Energy Consumption E, so the higher the fitness of the path, the better the energy consumption. We define the fitness of the path r_k according to Equation (9):

$$f(r_k) = \alpha \times \eta(r_k) + \frac{\beta}{C(r_k)} + \gamma \times E(r_k) \tag{9}$$

where α, β, γ are the adjustment parameters to ensure that the η is the maximum weight, followed by *C*. *E* is minimum, and its value depends on the specific network environment.

6.2. Initial Path Generation Formula

In the initial stage, the ABC algorithm generates *SN* initial solutions for the worker bees randomly by using Equation (6). In the algebraic optimization problem, the solution space is continuous, so the randomly generated solutions are feasible solutions, but each solution of the path planning problem is a path in a particular network topology, its path is discrete, so not any combination of nodes is a path, therefore, we cannot directly apply Equation (6) to generate the initial path.

In order to find a random path, our method is to randomly select a node from neighbors of node v_0 as the next node of the path, and then continue to generate the next node of the path in the same way. If one cannot find the neighbor node, then one goes back to the previous step to select another neighbor node as the next node until the next node is the path end point v_n, to obtain a random path.

Let the function *Adj(i)* return the set of all adjacencies of node *i*. Then Equation (10) for generating the node *t* in path r_i is:

$$r_{i,t} = Rand(Adj(t-1)) \tag{10}$$

where $t \in \{0, 1, \dots, m-1\}$, *m* is the number of nodes of path *i*, *t* is a randomly selected node in path *i*, and $t-1$ is the predecessor node of node t.

6.3. The Generation Formula of the New Path

Definition 7. *In two paths, if there are sections which have the same starting and ending points, it is said that same segments are Same Area (SA) for the two paths.*

Conclusion. $r_i =< v_0 \dots v_r \dots v_s \dots v_n > r_j =< v_0 \dots v_p \dots v_q \dots v_n >$, *$r_i$ and r_j have multiple SA, randomly select an SA, denoted as SA_α. SA_α in r_i is <r \dots s>, denoted as $SA_{\alpha i}$; in r_j is <p \dots q>, denoted as $SA_{\alpha j}$. If $SA_{\alpha i}$ in path r_i is replaced by $SA_{\alpha j}$, so the new path is $< v_0 \dots v_{r-1}, v_p \dots v_q, v_{s+1} \dots v_n >$ and is still connected.*

Proof. r_i is a connected path, and its sub-paths $< v_0 \dots v_{r-1} >$ and $< v_{s+1} \dots v_n >$ are also connected in accordance with the characteristics of the connected path.

Using the $SA_{\alpha j}$ of the r_j path to replace the $SA_{\alpha i}$ of the r_i path, the r_i becomes $< v_0 \dots v_{r-1}, v_p \dots v_q, v_{s+1} \dots v_n >$, denoted as $r_i{}^*$.

$p = r$, then $< v_0 \dots v_{r-1}, v_p >$ is connected.

$q = s$, then $< v_q, v_{s+1} \dots v_n >$ is connected.

Since r_j is connected, so the $SA < v_p \dots v_q >$ of its must be connected.

In summary, the sub-paths $< v_0 \dots v_{r-1}, v_p >, < v_p \dots v_q >, < v_q, v_{s+1} \dots v_n >$ are connected, so their order combination $r_i{}^* =< v_0 \dots v_{r-1}, v_p \dots v_q, v_{s+1} \dots v_n >$ must be connected. □

When path *i* and *j* exist *k* SA, then SA_t in the path *i* and *j* are denoted as SA_{ti} and SA_{tj} respectively, so path $i =< v_0 \dots v_{r-1} > SA_{ti} < v_{s+1} \dots v_n >$. The function $GetSA(i, j, t)$ returns SA_{tj}, then the equation for generating the new path $r_i{}^*$ is:

$$r_i{}^* = \begin{cases} < v_0 \dots v_{r-1} > GetSA(i,j,t) < v_{s+1} \dots v_n > \\ r_i(when\ r_i\ and\ r_j\ have\ no\ SA) \end{cases} \tag{11}$$

Equation (11) is the search equation for the OEABC algorithm, $i, j \in \{1, 2, \dots, SN\}$, $t \in \{0, 1, \dots, m-1\}$, *m* is the number of *SA* of path *i* and path *j*.

For example, the WBANs with 31 nodes, v_0 is the starting node, v_{30} is the sink, there are already two paths r_0 and r_1, $r_0 = <v_0, v_1, v_6, v_{12}, v_{14}, v_{17}, v_{24}, v_{27}, v_{30}>$, $r_1 = <v_0, v_4, v_6, v_{11}, v_{13}, v_{16}, v_{17}, v_{23}, v_{29}, v_{30}>$, as shown in Figure 3. According to Definition 7, there are 5 *SAs* for r_0 and r_1, in r_0 is $<v_0, v_1, v_6>$, $<v_0, v_1, v_6, v_{12}, v_{14}, v_{17}>$, $<v_6, v_{12}, v_{14}, v_{17}>$, $<v_6, v_{12}, v_{14}, v_{17}, v_{24}, v_{27}, v_{30}>$, $<v_{17}, v_{24}, v_{27}, v_{30}>$, in r_1 is $<v_0, v_4, v_6>$, $<v_0, v_4, v_6, v_{11}, v_{13}, v_{16}, v_{17}>$, $<v_6, v_{11}, v_{13}, v_{16}, v_{17}>$, $<v_6, v_{11}, v_{13}, v_{16}, v_{17}, v_{23}, v_{29},$

$v_{30}>$, $<v_{17}, v_{23}, v_{29}, v_{30}>$. If $t = 2$, then $GetSA(i, j, t) = <v_6, v_{11}, v_{13}, v_{16}, v_{17}>$, according to Equation (11), the new paths $r_2 = <v_0, v_1, v_6, v_{11}, v_{13}, v_{16}, v_{17}, v_{24}, v_{27}, v_{30}>$ generated by r_0, as shown in Figure 4.
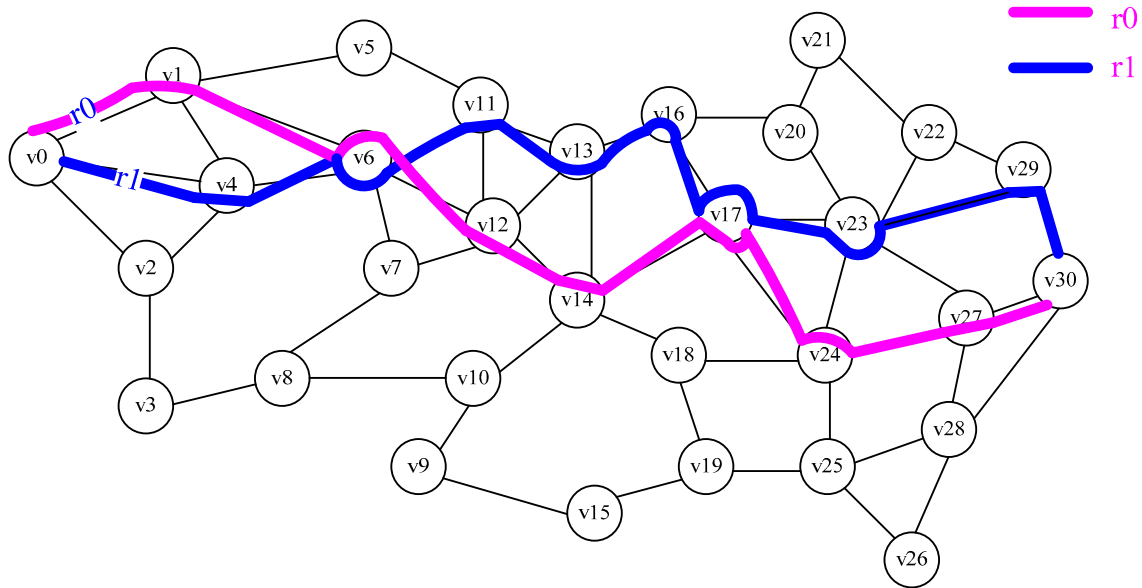

Figure 3. Two paths of r_0 and r_1.
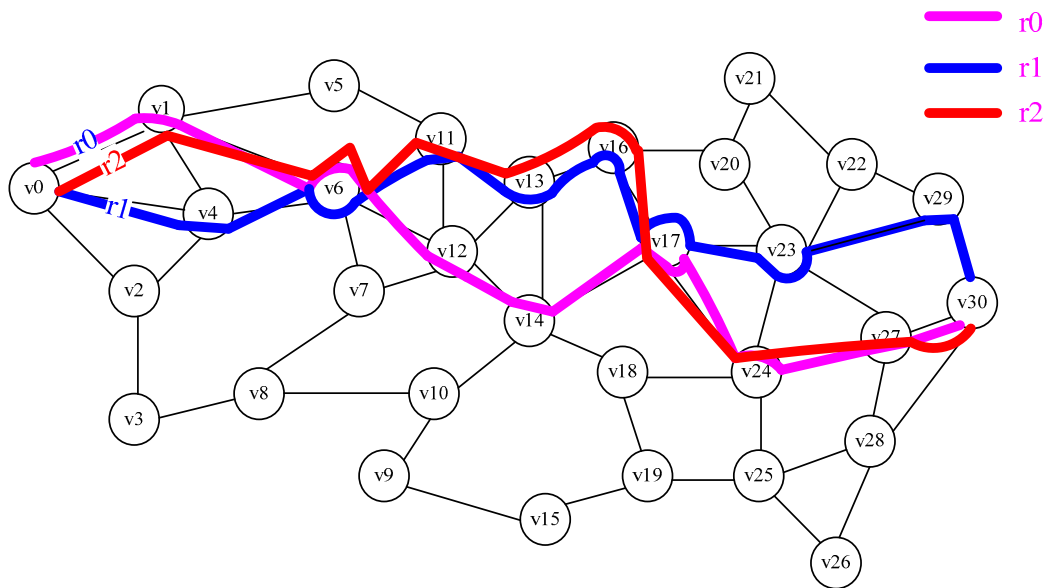

Figure 4. The new paths r_2 generated by r_0.

OEABC is implemented using the following algorithm (Algorithm 1):

Algorithm 1. OEABC Algorithm

Input: Network topology, starting point, end point
Output: Optimal energy consumption path

1.   Generate the data structure of the WBANs' topology;
2.   Initialize the population of bees according to Equation (10), generate the initial path of *SN*;
3.   **For** (num = 0→maxCycle){
4.   The employed bee search better path[i] in the neighborhood according to Equation (11);
5.   Calculate the fitness[i] of path[i] according to Equation (9);
6.   Using prob[i] = (0.9*(fitness[i]/maxfit)) + 0.1, calculate the selection probability of path[i];
7.   **For** (*t* = 0→SN){
8.   randvalue = ((double)rand()/((double)(RAND_MAX) + (double)(1)));
9.   **If** randvalue < prob[i] **Then**
10.   t++;
11.   r = ((double)rand()/((double)(RAND_MAX) + (double)(1)));
12.   neighbor = (int)(r*FoodNumber);
13.   **While** (neighbour==i){
14.   r = ((double)rand()/((double)(RAND_MAX)+(double)(1)));
15.   neighbor = (int)(r*FoodNumber);}
16.   The onlooker produce new path between path[i] and path[neighbour] according to Equation (11);
17.   Calculate the fitness of the path[new] according to Equation (9);
18.   **If** fitness[new] > fitness[i] **Then**
19.   path[i] = path[new];}
20.   **End If**
21.   **End If**
22.   }
23.   **If** path[i] is still not optimized by the maxtrial round search **Then**
24.   send ScoutBees, using Equation (10) to reinitialize path[i];
25.   **End If**
26.   }

---

7. Experiments and Results Analysis

We set the body area network node set $V = \{v_0, v_2, v_3, \ldots, v_{150}\}$ for the simulation environment, the data sending node is v_0, and the sink node is v_{150}; the node remaining energy $c_i \in \{1, 2, \ldots, 50\}$; the node service factor $\alpha_i \in \{1, 2, 3\}$; $E = \{e_1, e_2, e_3, \ldots, e_{1525}\}$ is an edge set. When $<i, j> \in E$, node i transmits a packet to node j, energy consumption $a_{ij} \in \{1, 2, 3\}$. In order to facilitate the verification result, it is assumed that the node topology is a hierarchical network, and each layer node is only connected to the upper layer and the next layer. The adjustment parameters for setting the fitness function are: $\alpha = 50$, $\beta = 5$, $\gamma = 0.1$.

We implemented the OEABC algorithm in C++ programming and tested it in the above simulation environment. In our simulation configuration, we adopt the following simulation settings: the data collection node is v_0 and the sink node is v_{150}. The communication link is established between neighbor nodes, and each node has 10–15 predecessors and successors. The data packet is sent out from v_0, the data packet is exchanged by the successor node, the transmission path from the data collection node to the network sink node is established, and the data packet is sent to the sink node v_{150}. The main simulation parameter values are listed in Table 2 below.

Table 2. Simulation Parameter Settings.

Parameter	Value
Simulation area:	100 m × 100 m
Energy model:	Generic radio energy model
Attenuation model:	Two ray
Signal transmission range:	10 m
Signal interference range:	20 m
Packet size:	512 Byte
Output queue type:	First-In First-Out (FIFO)
Channel capacity:	1 Mbit/s
Cache capacity:	50 packets

7.1. Fitness Analysis of the Path

The OEABC parameters are set as: *SN* = 5, *Limit* = 10, *Cycle* = 50, and the running process of the algorithm is as follows:

The first step is to initialize the population. Randomly generate five paths and calculate the fitness of each path. The results are shown in Table 3. Path 4 has the best fitness and is the optimal path.

Table 3. The stage of initialization with five paths.

No.	Path	η	C	E	Fitness
1	$v_0, v_6, v_{19}, v_{35}, v_{53}, v_{67}, v_{88}, v_{100}, v_{112}, v_{133}, v_{140}, v_{150}$	0.57	0.63	26	46.97
2	$v_0, v_5, v_{28}, v_{37}, v_{60}, v_{75}, v_{84}, v_{100}, v_{118}, v_{132}, v_{138}, v_{150}$	0.49	0.56	29	45.26
3	$v_0, v_{13}, v_{28}, v_{35}, v_{49}, v_{71}, v_{81}, v_{97}, v_{107}, v_{123}, v_{136}, v_{150}$	0.67	0.48	31	57.43
4	$v_0, v_{10}, v_{23}, v_{43}, v_{48}, v_{74}, v_{85}, v_{93}, v_{114}, v_{134}, v_{139}, v_{150}$	0.68	0.45	22	58.42
5	$v_0, v_7, v_{27}, v_{41}, v_{56}, v_{73}, v_{90}, v_{93}, v_{109}, v_{131}, v_{149}, v_{150}$	0.49	0.44	19	49.13

The second stage is the worker bee stage. The results are shown in Table 4. It can be seen that paths 1 and 4 are optimized, the path fitness is improved, and the optimal path is path 4.

Table 4. The stage of worker bees.

No.	Path	η	C	E	Fitness
1	$v_0, v_6, v_{19}, v_{35}, v_{53}, v_{67}, v_{88}, v_{100}, v_{112}, v_{133}, v_{140}, v_{150}$	0.63	0.61	23	50.19
2	$v_0, v_5, v_{28}, v_{37}, v_{60}, v_{75}, v_{84}, v_{100}, v_{118}, v_{132}, v_{138}, v_{150}$	0.49	0.56	29	45.26
3	$v_0, v_{13}, v_{28}, v_{35}, v_{49}, v_{71}, v_{81}, v_{97}, v_{107}, v_{123}, v_{136}, v_{150}$	0.67	0.48	31	57.43
4	$v_0, v_{10}, v_{23}, v_{43}, v_{48}, v_{74}, v_{85}, v_{93}, v_{114}, v_{134}, v_{139}, v_{150}$	0.69	0.43	20	59.76
5	$v_0, v_7, v_{27}, v_{41}, v_{56}, v_{73}, v_{90}, v_{93}, v_{109}, v_{131}, v_{149}, v_{150}$	0.49	0.44	19	49.13

The third stage is the onlooker stage. As shown in Table 5, path 3 is optimized and the optimal path is still path 4. Since it is the first operation of the algorithm, it does not meet the requirements of Limit, so the scout phase is not performed.

Table 5. The stage of onlookers.

No.	Path	η	C	E	Fitness
1	$v_0, v_6, v_{19}, v_{35}, v_{53}, v_{67}, v_{88}, v_{100}, v_{112}, v_{133}, v_{140}, v_{150}$	0.63	0.61	23	50.19
2	$v_0, v_5, v_{28}, v_{37}, v_{60}, v_{75}, v_{84}, v_{100}, v_{118}, v_{132}, v_{138}, v_{150}$	0.51	0.56	29	46.26
3	$v_0, v_{13}, v_{28}, v_{35}, v_{49}, v_{71}, v_{81}, v_{97}, v_{107}, v_{123}, v_{136}, v_{150}$	0.66	0.45	23	57.52
4	$v_0, v_{10}, v_{23}, v_{43}, v_{48}, v_{74}, v_{85}, v_{93}, v_{114}, v_{134}, v_{139}, v_{150}$	0.69	0.43	20	59.76
5	$v_0, v_7, v_{27}, v_{41}, v_{56}, v_{73}, v_{90}, v_{93}, v_{109}, v_{131}, v_{149}, v_{150}$	0.49	0.44	19	49.13

The results of the operations after 50 iterations, are shown in Table 6. It can be seen that the five paths have been optimized, the optimal path has changed from path 4 to path 3, and the fitness has also increased from the initial 58.42 to 71.46.

Table 6. Results after 50 iterations.

No.	Path	η	C	E	Fitness
1	$v_0, v_6, v_{19}, v_{35}, v_{53}, v_{67}, v_{88}, v_{100}, v_{112}, v_{133}, v_{140}, v_{150}$	0.69	0.51	21	56.21
2	$v_0, v_5, v_{28}, v_{37}, v_{60}, v_{75}, v_{84}, v_{100}, v_{118}, v_{132}, v_{138}, v_{150}$	0.51	0.42	27	52.01
3	$v_0, v_{13}, v_{28}, v_{35}, v_{49}, v_{71}, v_{81}, v_{97}, v_{107}, v_{123}, v_{136}, v_{150}$	0.75	0.31	17	71.46
4	$v_0, v_{10}, v_{23}, v_{43}, v_{48}, v_{74}, v_{85}, v_{93}, v_{114}, v_{134}, v_{139}, v_{150}$	0.73	0.33	18	68.60
5	$v_0, v_7, v_{27}, v_{41}, v_{56}, v_{73}, v_{90}, v_{93}, v_{109}, v_{131}, v_{149}, v_{150}$	0.54	0.39	17	54.34

7.2. Parameter Analysis

Before running the OEABC, one needs to set the *SN*, *Limit*, and *Cycle* parameters. *SN* is the population size. *Limit* is when to give up with the current solution required for the maximum number of searches. *Cycle* refers to the number of iterations of the algorithm. By setting different *SN*, *Limit* and *Cycle* parameters, run the OEABC 30 times, record and analyze the fitness of the generated path.

7.2.1. The Analysis of SN

Set *Limit* = 30, *Cycle* = 50. When the *SN* is 20, 30, 40, respectively, the fitness of the path generated by the OEABC is shown in Figure 5. As can be seen from Figure 5, When *SN* = 20, in the 30 paths generated by the algorithm, a total of four paths have a fitness change greater than 10% of the average fitness, compared with the generated paths, the ratio of paths greater than 10% of the average fitness is 13%. This is because the initial paths generated by OEABC are stochastic, and when the number of iteration *Cycle* is greater than the limit, the new path generated by the scout bee is also stochastic.
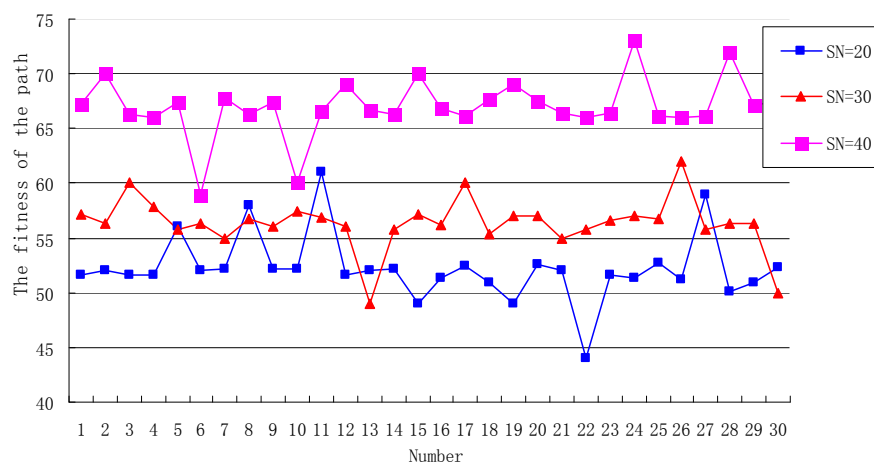

Figure 5. The fitness of the path when the *SN* is different.

When *SN* = 30, there are three points with an amplitude of change greater than 10% of the average fitness, and compared with the generated paths, only 6% of the paths are more than 10% of the average fitness, and the variation amplitude of the generated solution tends to converge with the increase of *SN*.

When *SN* = 40, there is only one point with a variation amplitude greater than 10% of the average fitness, and compared with the generated paths, only 3% of the paths are more than 10% of the average fitness, the fluctuation range of the fitness curve becomes smaller and tends to be stable.

Therefore, the bigger the *SN* is, the higher the fitness of the path generated by the algorithm, the closer we are to the optimal solution. At the same time, it can also be seen that the path ratio of the

fitness change amplitude greater than 10% of the average fitness decreases from 13% to 3%. Therefore, with the increase of *SN*, the variation amplitude of the generated solution tends to converge, and the stability of the algorithm is gradually improved. This is because *SN* becomes larger, which means that the larger the solution space is, the higher the probability of obtaining the optimal solution is. However, the increase of *SN* will lead to the synchronous increase of the overhead of the algorithm, which cannot increase the *SN* indefinitely. It is necessary to select the *SN* which is suitable for the scale of the problem.

7.2.2. The Analysis of Limit

Set *SN* = 30, *Cycle* = 50. When the *Limit* is 5, 30, 50, respectively, the fitness of the path generated by the OEABC is shown in Figure 6.
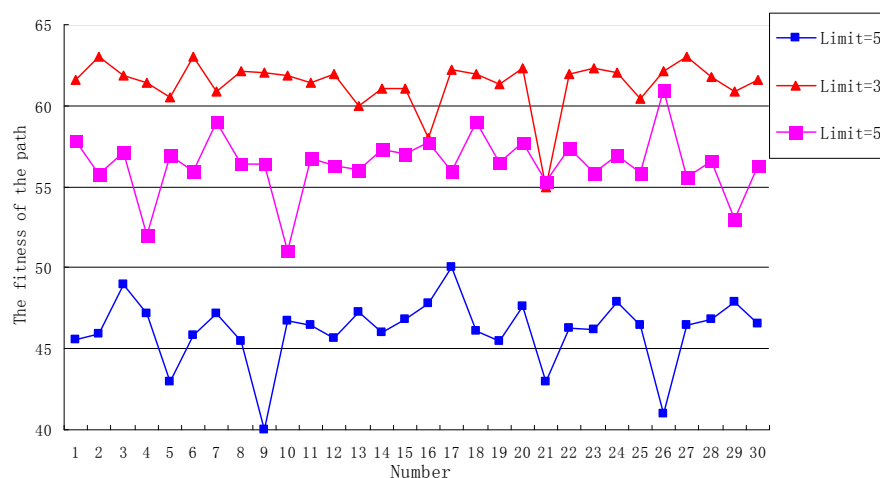

Figure 6. The fitness of the path when the *Limit* is different.

As can be seen from Figure 6, when *Limit* = 5, the overall level of the generated solution is low, and the average fitness of the path is only 46. This is because the *Limit* value is small, meaning that it will be possible to release the scout bee multiple times to generate random paths. Randomness enhancement of the algorithm generation solution and the probability of finding the optimal solution decreases obviously because the experience obtained before cannot be fully utilized to search for a new solution.

When *Limit* = 30, although the *Limit* is not the largest, the overall level of the fitness curve is the highest, and because of the reasonable value of the *Limit*, the algorithm does not fall into a local optimum, nor is it completely dependent on stochastic generation, giving full play to the advantages of both local and global optimization of the ABC algorithm.

When *Limit* = 50, the *Limit* becomes larger, but the fitness of the generation path decreases. When the value of *Limit* is larger, the generated path can easily to fall into a local optimum, and one cannot achieve global optimization, which leads to a low level of fitness of the path, so a bigger *Limit* is not the best one, and there is a suitable *Limit* size to get a better path fitness. The size of the *Limit* needs to be proportional to the *SN*, the *Cycle* and other parameters of OEABC to avoid both complete stochasticity and local optimization.

7.2.3. The Analysis of Cycle

Set *SN* = 30, *Limit* = 30, when the *Cycle* number is 30, 60, 90, respectively, the fitness of the path generated by the OEABC is shown in Figure 7.
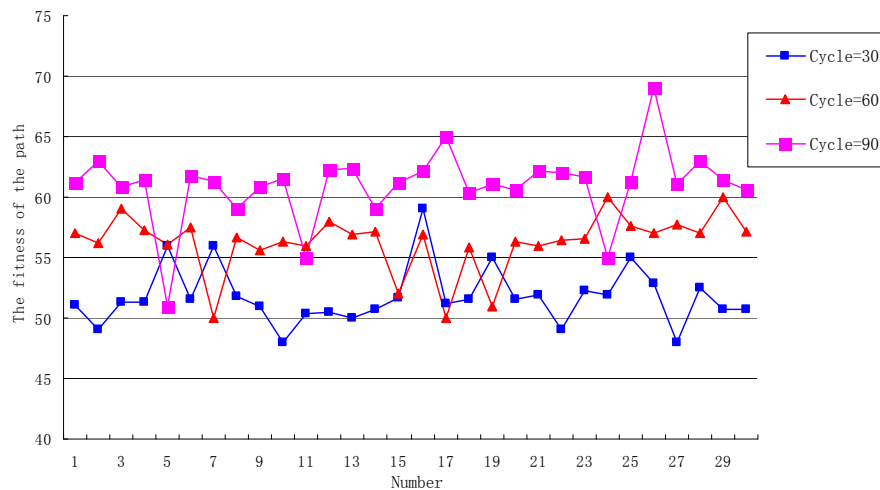
Figure 7. The fitness of the path when the number of *Cycle* is different.

As can be seen from Figure 7, When *Cycle* = 30, in the 30 paths generated by the algorithm, the change amplitude of fitness is greater than 10% of the average (that is, the fitness is more than 57 or less than 47), which is only the 16th path with a value of 60, merely 3% of the paths are more than 10% of the average fitness. The reason is that *Cycle* is too small and there are not enough iterations of the solution, so it is difficult to generate the path with higher fitness.

When *Cycle* = 60, compared with the previous round, the average fitness of the current round increased by nearly 10% from 52 to 58, because the *Cycle* value is large enough to enable the solution to be fully iterated.

When *Cycle* = 90, compared with *Cycle* = 60, the average fitness rose from 58 to 61, an increase of about 5%. Although the further increase of the *Cycle* enables the solution to be further optimized and the best solution 70 appears, the improvement decreases gradually. This is because the initial solution of the ABC algorithm and the solution generated by the scout bee are both stochastic, if the random solution is not good, it is difficult to generate the solution with high fitness by simply increasing the number of iterations.

Therefore, the bigger the *Cycle* is, the higher the fitness of the algorithm generating path is. It can also be seen that when the *Cycle* rises from 30 to 60, the average fitness increase is significantly greater than when the *Cycle* rises from 60 to 90. That is to say, when the *Cycle* reaches a certain value, the effect of improving path fitness will be worse and worse simply by adding to the *Cycle* number. The *Cycle* number not only maintains a proportional relationship with the *SN*, *Limit* and other parameters, but also considers the computational ability of nodes to ensure that the solutilon can be quickly iterated.

In summary, through the analysis of the *SN*, *Limit* and *Cycle* parameters, it can be seen that the fitness of the path generated by OEABC with different parameters changes greatly, which has a great impact on the performance of the OEABC. The *SN* represents the size of the bee colony, and a larger colony will be able to cover a wider range of solution space, effectively improving the probability of finding an optimal solution. Through the roulette mechanism, OEABC implements positive feedback on the path of higher fitness, by conducting the *Cycle* number of rounds of search in the neighborhood, iterate to find the path with higher fitness, but the problem is that it is easy to fall into a local optimum. The scout bee mechanism is used to solve this problem, and when a path cannot be optimized after the *Limit* iterations, it will be discarded and a random path will be generated, ensuring an opportunity to continue searching for the optimal solution globally. In the specific application scenario, selecting parameters not only considers the performance of the OEABC algorithm itself, but also should take into account the size of the body area network, the computational ability of nodes and specific requirements that need to be addressed.

7.3. Comparative Analysis

In order to further verify the performance of the algorithm, OEABC, the genetic algorithm [37] and ant colony [38] algorithms are used to generate the optimal energy consumption path in the above network environment. Two experiments are designed to compare and analyze the three heuristic algorithms. In Experiment 1, we compared the percentage for the appearance of optimal solution, while in Experiment 2 is the convergence rate of the algorithm.

Experiment 1. *The population size and the number of iterations of the three algorithms are the same, 20 and 100, respectively, other parameter settings also choose the most commonly used settings, details as follows: OEABC: The Limit parameter is 30. Genetic Algorithm: crossover probability Pc is 0.8, mutation probability Pm is 0.2. Ant Colony Algorithm: the proving factor ALPHA is 1.0, the expected factor BETA is 2.0, the pheromone volatility coefficient ROU is 0.5.*

We run the above three algorithms 50, 100 and 150 times, respectively, then calculate the percentage for the appearance of the optimal solution. The calculation method is the number of optimal solutions divided by the total number of runs, and the results are shown in Table 7.

Table 7. The percentage for the appearance of optimal solution.

Run Times	OEABC	Genetic Algorithm	Ant Colony Algorithm
50	16.00%	14.00%	18.00%
100	15.00%	16.00%	11.00%
150	16.67%	13.33%	17.33%
Average	15.89%	14.44%	15.44%

Experiment 2. *In order to verify the convergence speed of the algorithm, in the above network environment, set the parameters SN = 30, Limit = 20, Cycle = 50, by recording the fitness of five paths generated by OEABC that form a convergence curve. The algorithm runs five times, and the corresponding convergence curve shown in Figure 8.*
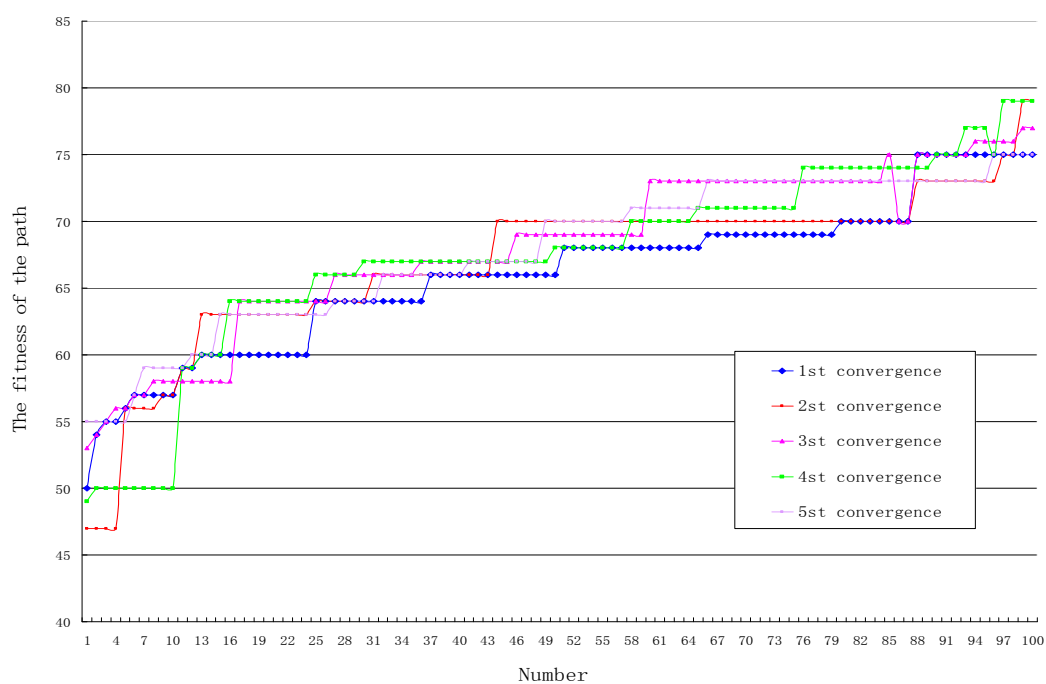

Figure 8. The convergence curve of the algorithm.

The convergence rate *Cv* is calculated as:

$$Cv = \frac{f(r_{opt}) - f(r_{init})}{Cycle} \tag{12}$$

where r_{init} is the initial path, r_{opt} is the optimal energy consumption path.

As can be seen from Figure 8, after 50 iterations, the 1st convergence is from 50 to 75, *Cv* is 0.5; the 2th from 47 converge to 79, *Cv* is 0.64; the 3th from 53 converge to 77, *Cv* is 0.48; the 4th from 49 converge to 79, *Cv* is 0.6; the 5th from 55 converges to 75, *Cv* is 0.4, so the arithmetic average of 50 times *Cv* is 0.52.

Comparing the convergence rates of the above three algorithms in the case of population size 20, 30, 40, the convergence rate *Cv* is calculated using Equation (12). The number of iterations are 50, the other parameters are the same as in Experiment 1. The experimental results are shown in Table 8.

Table 8. The convergence rate.

Population Size	OEABC	Genetic Algorithm	Ant Colony Algorithm
20	0.81	0.76	0.83
30	0.89	0.92	0.89
40	1.17	1.08	1.01
Average	0.96	0.92	0.91

Through the comparison results, we can see the OEABC has an obvious advantage in solving the problem of optimal energy consumption path of WBANs, whether the percentage for the appearance of optimal solution or the convergence rate of the algorithm are considered.

8. Conclusions

By considering the factors that affect the energy consumption of the WBANs, the definition of the optimal energy consumption path is given. Combined with the idea of ABC, using the method of exchanging path SA, the OEABC algorithm is designed and implemented, and the ABC is successfully applied to solve the problem of discrete solution space. The experimental results show that the OEABC can effectively solve the NP problem of the optimal energy consumption path in WBANs. Compared with the ant colony algorithm and genetic algorithm, it is further shown that the OEABC has a good performance in obtaining the optimal path of energy consumption and convergence rate.

References

1. IEEE. *IEEE Standard for Local and Metropolitan Area Networks—Part 15.6: Wireless Body Area Networks*; IEEE Std.: Piscataway Township, NJ, USA, 2012; pp. 1–271.
2. Otto, C.; Milenkovi, A.; Sanders, C.; Jovanov, E. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *J. Mob. Multimed.* **2006**, *1*, 307–326.
3. Hanson, M.A.; Powell, H.C.; Barth, A.T.; Ringgenberg, K.; Calhoun, B.H.; Aylor, J.H.; Lach, J. Body Area Sensor Networks: Challenges and Opportunities. *Computer* **2009**, *42*, 58–65. [CrossRef]
4. Patel, M.; Wang, J. Applications, challenges, and prospective in emerging body area networking technologies. *IEEE Wirel. Commun.* **2010**, *17*, 80–88. [CrossRef]

5. Ning, Z.; Xia, F.; Hu, X.; Chen, Z.; Obaidat, M.S. Social-oriented adaptive transmission in opportunistic Internet of smartphones. *IEEE Trans. Ind. Inform.* **2017**, *13*, 810–820. [CrossRef]

6. Scanlon, W.G.; Conway, G.A.; Cotton, S.L. Antennas, propagation considerations for robust wireless communications in medical body area networks. *Proc. IET Semin. Antennas Propag. Body Cent. Wirel. Commun.* **2007**, *11803*, 37.

7. Kiourti, A.; Nikita, K.S. A Review of Implantable Patch Antennas for Biomedical Telemetry: Challenges and Solutions [Wireless Corner]. *IEEE Antennas Propag. Mag.* **2012**, *54*, 210–228. [CrossRef]

8. Quwaider, M.; Biswas, S. DTN routing in body sensor networks with dynamic postural partitioning. *Ad Hoc Netw.* **2010**, *8*, 824–841. [CrossRef] [PubMed]

9. Tachtatzis, C.; Franco, F.D.; Tracey, D.C.; Timmins, N.; Morrison, J. An energy analysis of IEEE 802.15.6 scheduled access modes. In Proceedings of the 2010 IEEE Globecom Workshops, Miami, FL, USA, 6–10 October 2012; pp. 1270–1275.

10. Movassaghi, S.; Abolhasan, M.; Lipman, J. A Review of Routing Protocols in Wireless Body Area Networks. *J. Netw.* **2013**, *8*, 559–575. [CrossRef]

11. Bangash, J.I.; Abdullah, A.H.; Anisi, M.H.; Khan, A.W. A Survey of Routing Protocols in Wireless Body Sensor Networks. *Sensors* **2014**, *14*, 1322–1357. [CrossRef] [PubMed]

12. Ning, Z.; Liu, L.; Xia, F.; Jedari, B.; Lee, I.; Zhang, W. CAIS: A copy adjustable incentive scheme in community-based socially-aware networking. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3406–3419. [CrossRef]

13. Ning, Z.; Xia, F.; Ullah, N.; Kong, X.; Hu, X. Vehicular social networks: Enabling smart mobility. *IEEE Commun. Mag.* **2017**, *55*, 49–55. [CrossRef]

14. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]

15. Karaboga, D.; Akay, B. A comparative study of Artificial Bee Colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132. [CrossRef]

16. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [CrossRef]

17. Watteyne, T.; Auge-Blum, S.; Dohler, M.; Barthel, D. Anybody: A self-organization protocol for body area networks. In Proceedings of the ICST 2nd international conference on Body area networks, Florence, Italy, 11–13 June 2007.

18. Culpepper, B.J.; Dung, L.; Moh, M. Design and analysis of hybrid indirect transmissions (HIT) for data gathering in wireless micro sensor networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2004**, *8*, 61–83. [CrossRef]

19. Moh, M.; Culpepper, B.; Dung, L.; Moh, T.-S.; Hamada, T.; Su, C.F. On data gathering protocols for in-body biomedical sensor networks. In Proceedings of the IEEE Global Telecommunications Conference, St. Louis, MO, USA, 28 November–2 December 2005.

20. Braem, B.; Latre, B.; Moerman, I.; Blondia, C.; Demeester, P. The wireless autonomous spanning tree protocol for multihop wireless body area networks. In Proceedings of the Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services, San Jose, CA, USA, 17–21 July 2006; pp. 1–8.

21. Braem, B.; Latre, B.; Blondia, C.; Moerman, I.; Demeester, P. Improving reliability in multi-hop body sensor networks. In Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications, Cap Esterel, France, 25–31 August 2008; pp. 342–347.

22. Latre, B.; Braem, B.; Moerman, I.; Blondia, C.; Reusens, E.; Joseph, W.; Demeester, P. A low-delay protocol for multihop wireless body area networks. In Proceedings of the Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking Services, Philadelphia, PA, USA, 6–10 August 2007; pp. 1–8.

23. Braem, B.; Latr, B.; Moerman, I.; Blondia, C.; Reusens, E.; Joseph, W. The Need for Cooperation and Relaying in Short-Range High Path Loss Sensor Networks. In Proceedings of the International Conference on Sensor Technologies and Applications, Valencia, Spain, 14–20 October 2007; pp. 566–571.

24. Joseph, W.; Braem, B.; Reusens, E. Design of Energy Efficient Topologies for Wireless On-Body Channel. In Proceedings of the Seventeenth Wireless Conference-Sustainable Wireless Technologies, Vienna, Austria, 27–29 April 2011; pp. 1–7.

25. Ferrand, P.; Maman, M.; Goursaud, C. Performance evaluation of direct and cooperative transmissions in body area networks. *Ann. Telecommun.* **2011**, *66*, 213–228. [CrossRef]

26. Maskooki, A.; Soh, C.B.; Gunawan, E.; Low, K.S. Opportunistic routing for body area network. In Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2011; pp. 237–241.

27. Liang, X.; Li, X.; Shen, Q.; Lu, R.; Lin, X.; Shen, X.; Zhang, W. Exploiting Prediction to Enable Secure and Reliable Routing in Wireless Body Area Networks. In Proceedings of the Thirty-First IEEE International Conference on Computer Communications (INFOCOM), Orlando, FL, USA, 25–30 March 2012; pp. 388–396.

28. Quwaider, M.; Biswas, S. Probabilistic routing in on-body sensor networks with postural disconnections. In Proceedings of the Seventh ACM International Symposium on Mobility Management and Wireless Access, Tenerife, Canary Islands, Spain, 26–27 October 2009; pp. 149–158.

29. Quwaider, M.; Biswas, S. On-body packet routing algorithms for body sensor networks. In Proceedings of the First International Conference on Networks and Communications (NETCOM '09), Washington, DC, USA, 27–29 December 2009; pp. 171–177.

30. Djenouri, D.; Balasingham, I. New QoS and geographical routing in wireless biomedical sensor networks. In Proceedings of the Sixth International Conference on Broadband Communications, Networks, and Systems (BROADNETS), Madrid, Spain, 14–16 September 2009; pp. 1–8.

31. Razzaque, M.A.; Hong, C.S.; Lee, S. Data-centric multiobjective QoS-aware routing protocol for body sensor networks. *Sensors* **2011**, *11*, 917–937. [CrossRef] [PubMed]

32. Zuhra, F.; Bakar, K.; Ahmed, A.; Tunio, M. Routing protocols in wireless body sensor networks: A comprehensive survey. *J. Netw. Comput. Appl.* **2017**, *99*, 73–97. [CrossRef]

33. Takahashi, D.; Xiao, Y.; Hu, F.; Chen, J.; Sun, Y. Temperature aware routing for telemedicine applications in embedded biomedical sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2008**, 26. [CrossRef]

34. Shah, R.; Yarvis, M. TARA: Thermal-aware routing algorithm for implanted sensor networks. In *International Conference on Distributed Computing in Sensor Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 206–217.

35. Bag, A.; Bassiouni, M.A. Energy efficient thermal aware routing algorithms for embedded biomedical sensor networks. In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Vancouver, BC, Canada, 9–12 October 2006; pp. 604–609.

36. Mitra, R.; Sharma, S. Proactive data routing using controlled mobility of a mobile sink in Wireless Sensor Networks. *Comput. Electr. Eng.* **2018**, *70*, 21–36. [CrossRef]

37. Goldberg, D.E.; Holland, J.H. Genetic algorithms and machine learning. *Mach. Learn.* **1988**, *3*, 95–99. [CrossRef]

38. Dorigo, M.; Blum, C. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278. [CrossRef]