Algorithm Optimization for Energy Consumption Issues in Wireless Sensor Network Nodes

Yu Wei*, Yang Sun

School of Information Engineering, Xijing University, Xi'an, China

*Corresponding author: isweiyu0625@gmail.com

Abstract

This study proposes an energy aware routing algorithm based on ant colony algorithm and Dijkstra algorithm to address the issues of uneven energy distribution and premature node death leading to information transmission obstruction in wireless sensor networks. This algorithm updates the position coordinates and remaining energy of each node in real-time, and uses ant colony algorithm to track and record the number of ant arrivals between nodes as the weight matrix of the Dijkstra algorithm. At the same time, the power consumption parameter per unit path length is taken into account in the algorithm, and a dynamic weight matrix is generated through multiple optimizations, thereby achieving dynamic routing between any node. The experimental results on the simulation platform MATLAB show that compared to the algorithm that directly optimizes the path, this improved algorithm achieves dynamic planning of wireless sensor network layer routing, solves the problem of uneven energy consumption between nodes, and extends the lifespan of the network.

Keywords

Wireless Sensor Network; Ant Colony Algorithm; Dijkstra Algorithm; Energy Consumption Balance.

---

1. Introduction

With the development of 5G, MEMS and edge computing, almost all objects can be embedded in intelligent sensors to sense the changes of the surrounding environment in real time[1-2]. These miniature sensors that already have real-time sensing, communication, and data processing capabilities are advancing with the development of technology, which has also led to unprecedented development in wireless sensor networks. Wireless sensor networks are a highly interdisciplinary and concentrated cutting-edge research field, particularly highly valued by the military, industry, and related research institutions. How to improve the lifecycle of wireless sensor networks, avoid network congestion, and improve network transmission rates has received widespread attention[3-4].

Ant colony algorithm has good applicability in the network layer protocol of wireless sensor networks. Based on knowing the number and location of network nodes, it can change the network topology structure through path exploration and crawling. The algorithm mainly relies on the weights of the weight matrix in a directed graph to find the path between two nodes. Transferring the dynamic topology implemented by ant colony algorithm to the algorithm and then injecting energy into it to obtain the fusion result of the three will have great significance for improving the routing protocol of the entire sensor network layer. Naik et al. [5] proposed an energy aware multi-objective improved Ramola optimization algorithm and multi-objective ant colony optimization (EA-MIROA-MACO) to improve the energy efficiency of WSN by eliminating node isolation problems. Jemla et al. [6] proposed a multi-objective trust aware artificial hummingbird algorithm M-TAAHA to achieve secure and reliable transmission in WSN with mobile receivers. Elyyan et al. [7] proposed an economically efficient routing method to extend the lifespan of sensor nodes in the Internet of Things. Nithya [8] proposed a multi-objective particle swallowing swarm optimization to protect data broadcasting on WSN. This optimization is a combination of moth flame and mixed swarm optimization, which considers four different parameters, such as trust, distance, energy, and hop count, for cluster head selection and routing path generation Boyineni et al. [9] used a modified version of ant colony optimization strategy to alleviate hotspot issues in WSN using data collected through mobile receivers, while improving energy efficiency, network lifespan, and throughput by reducing packet loss and latency.

The above research has done a lot of work on network routing planning [10-11] and extending network lifespan, but has not automatically incorporated energy into the algorithm to enable autonomous planning. Based on the theory of wireless sensor network protocol stack, this paper focuses on the node routing planning problem in the energy management plane when there is little remaining sensor energy. The ant colony algorithm is used to obtain the global optimal path and the number of times each two nodes can directly reach it. Then, the number of times each two nodes can directly reach it is used as the input matrix weight of the Dijkstra algorithm to solve the local optimum. Finally, an energy aware wireless sensor network routing planning is achieved by integrating path length, node remaining energy.

2. Algorithm Design

2.1 Algorithm Overall Framework Design

(1) Ant colony algorithm obtains the number of times any two reachable nodes can be reached

Using the pheromone accumulation and volatilization factors of each route in the ant colony algorithm, m ants are randomly scattered across n nodes to find the final iteration matrix Tabu(m,n) through the number of iterations iter_max. Assuming the counter count is nC.

The algorithm framework is as follows:

```
while nC$\leq iter\_max$
    Execute ant colony algorithm
    Update Tabu
end
```

(2) Process the iteration matrix to obtain the weight matrix of the algorithm

Use the for loop to iterate through both forward and backward directions to obtain the number of times ants have walked on the directed edge between two reachable nodes, and record the result in the weight matrix G for finding the adjacency vector. The algorithm updates the weight matrix every time it is executed. The false design counter is count2, and the total number of times set is number.

The algorithm framework is as follows:

```
while count2≤number
Randomly set starting and ending nodes
Remaining energy participates in updating the weight matrix for this time
Find the optimal path using the Dijkstra algorithm based on the weight matrix of this time
end
```

(3) Algorithm parameter settings

The final result of the algorithm is closely related to the setting of parameters, and how to choose parameters is one of the difficult issues in this article. Not only should we consider the number of ants, number of iterations, and information evaporation coefficient, but we also need to consider whether the power consumption coefficient of node transmission and reception can be applied to the fusion algorithm. In addition, after each node traversal, the energy of the nodes needs to be transformed. How to set the initial value of the weight matrix, the power consumption for information exchange, and the power consumption for forwarding information have become a focus issue.

2.2 Improving Ant Colony Algorithm

(1) Algorithm implementation for selecting the probability of the next node

In ant colony algorithm, each node has a different number of ants, and each ant has a different starting point. Over time, the pheromone concentration between adjacent nodes shows diversity. Except that not all ants will choose routes with relatively high pheromone concentrations. In response to this phenomenon, at a certain moment when ant k needs to choose the next node on one node, there will be different selection probabilities. The calculation formula for selecting probability $P(k)$ is:

$$P(k') = \left[Tau\big(visited(end), J(k)\big)^{Alpha}\right] * \left[Eta\big(visited(end), j(k)\big)^{Beta}\right], \tag{1}$$

$$P(k) = \frac{P(k')}{\sum_{j \in Arrive} P(k')}, \tag{2}$$

Among them, Tau represents the pheromone matrix, Eta represents heuristic information, which is set as the reciprocal of the distance in this algorithm. $Tau\big(visited(end), J(k)\big)$ represents the pheromone amount of two nodes, $Eta\big(visited(end), j(k)\big)$ represents the energy information amount of two nodes, and $Arrive$ represents the set of nodes that ants can choose from next. In addition, random numbers can be designed as needed to increase the randomness of ants selecting the next node.

(2) Implementation of pheromone update algorithm

In the well-known ant colony algorithm, the pheromones of each node are initially set to be the same, so the probability of ants selecting the next node is sufficiently random. As the number of iterations increases, the pheromones of each path will evaporate with time. At the same time, as the number of iterations increases, the pheromone concentration of some paths will become higher and higher, while the pheromone concentration of some paths will gradually evaporate due to the lack of ant selection.

So, in the algorithm, it is necessary to build a model for the dynamic volatilization and increase of pheromones. The factors that affect pheromone changes include the volatility coefficient per unit time and the consumption coefficient per unit path length. Based on this, the pheromone update algorithm established is:

$$Tau = (1 - Rho) * Tau + Delta\_Tau \tag{3}$$

Among them, Rho represents the volatility coefficient, $Delta\_Tau$ represents the increment of pheromones, and Tau represents the matrix of pheromones.

2.3 Dynamically Improving the Weight Matrix S of Dijkstra Algorithm

The Dijkstra algorithm can obtain a relatively optimal path, but it cannot consider the phenomenon of node death caused by energy consumption during node information transmission. In response to this issue, this article proposes an algorithm that uses the number of times ants reach two nodes as the weight of the directed graph G, and the weight dynamically changes according to the rule of decreasing node energy, in order to find the dynamic optimal path that changes with the number of iterations.

In the traditional Dijkstra algorithm, the weighted edges in the weight matrix G are usually pre-set and then stored in the adjacency matrix arcs, where $G.arcs[i][j]$ represents the weight of arc $<$

$v_i, v_j >$. If $< v_i, v_j >$ does not exist, set the value at this position to infinity. In this algorithm, the weight on arc $< v_i, v_j >$ is determined by the number of arrivals of the two nodes in the improved ant colony algorithm, and the weight matrix will also change accordingly due to changes in pheromones and energy during each run. The improvement steps for the weight matrix G are as follows:

(1) Record the node code passed during the last iteration based on the results of the previous iteration.

(2) Each update multiplies the remaining points of the passed nodes by the weight matrix to ensure normalization of node energy.

(3) Update the weights of the rows and columns of the nodes in sequence, with the more iterations, the smaller the weight values of the updated node matrix.

The implementation algorithm of weight matrix G is as follows:

```
$num = length(W);$
$Wpath = length(path);$
$for\ \ f = 1:Wpath$
    $pathpo\mathrm{int} = path(f);$
$G(pathpo\mathrm{int},:) = (EnergyInit(pathpo\mathrm{int})/mult)*(G(pathpo\mathrm{int},:)*0.8);+(G(pathpo\mathrm{int},:)*0.2);$
$G(:,pathpo\mathrm{int}) = (EnergyInit(pathpo\mathrm{int})/mult)*(G(:,pathpo\mathrm{int})*0.8);+(G(pathpo\mathrm{int},:)*0.2);$
$end$
$W=G$
```

2.4 The Fusion Process of Improved Ant Colony Algorithm, Improved Dijkstra Algorithm, and Node Energy Update

(1) Obtaining the Input Matrix of Dijkstra Algorithm in Ant Colony Algorithm

After improving the ant colony algorithm and Dijkstra algorithm, the fusion of the two algorithms was carried out. The setting of the number of ants and the number of iterations in the ant colony algorithm will affect the final result. However, the result used in this article is not the final route formed by the ant colony algorithm, but the optimal number of reachable nodes is obtained by seeking a balance between the number of ants and the number of iterations. After experimental verification, it was found that when the number of nodes is constant, the smaller the value of the stable number of iterations, the more paths available for selection, and the more accurate the results. The more nodes there are, regardless of the number of iterations, the more types of paths there will be, and the overall energy consumption will be more balanced. During algorithm fusion, the forward and backward intermediate paths of the ant colony are recorded in the weight matrix G, so that nodes can reach each other.

(2) Integrating energy consumption patterns into the entire algorithm

When the energy of the routing node is sufficient, point A needs to pass through points B and C to reach point D. However, after a certain number of iterations, the energy of points B and C cannot support the information transmission of other nodes in the sensor network, which can cause partial or even global paralysis of the network. This project has set a critical energy value to address this issue. When frequent communication between point A and point D causes uneven energy distribution of the routing node, other paths are found to transmit information. When the energy of the routing node is insufficient, when point A reaches point D, it will change its route and dynamically plan other paths.

To balance the issue of energy imbalance between nodes, after each algorithm run, i.e. the entire network undergoes information transmission, the nodes involved in information transmission are changed, and their energy is modified. The modification must involve both rows and columns, so that the next running route can be changed from both directions. The energy consumption algorithm passed through the nodes is as follows.

When the path length relationship between two nodes is $d \leq d_0$, the formula for calculating the remaining energy of the nodes is:

$$Enek * Elec + Enek * d^2 + Energy' \tag{4}$$

When the path length relationship between two nodes is $d \geq d_0$, the formula for calculating the remaining energy of the nodes is:

$$Enek * Elec + Enek * d^4 + Energy', \tag{5}$$

Among them, d_0 is the threshold distance between two nodes, $Energy'$ is the remaining energy of each node after the last algorithm run, $Enek$ is the number of transmitted bits, and $Elec$ is the basic power consumption of node sending and receiving messages.

3. Results

3.1 Experimental Environment

This article has conducted multiple experiments on the platform to verify the correctness of algorithm fusion. During testing, the positions of each node were randomized and remained fixed throughout the experiment. The settings of simulation parameters in the experiment are shown in Table 1.

Table 1. Parameter settings

Experimental parameters	Value	Experimental parameters	Value
Ant number	200	Evaporation coefficient	0.8
Maximum number of iterations	15	Energy weight coefficient	1
Number of message deliveries	80	Distance weight coefficient	2
Visibility factor	1/D	Two node initial threshold	3
Energy matrix multiple	50	Heuristic factor parameters	5

3.2 Analysis of Experimental Results

(1) Selection of parameters for iteration times and number of ants

In algorithm fusion, the intermediate value of ant colony algorithm is used instead of the final result. The accuracy of the results of the ant colony algorithm before improvement depends on two aspects: first, when the number of ants is fixed, the more iterations, the more accurate the results will be. Secondly, when the maximum number of iterations is fixed, the more ants there are, the more accurate the results will be. The improved algorithm seeks a balance between the number of ants and the number of iterations, rather than the ultimate optimal shortest path. The experiments were conducted by changing the number of iterations and the number of ants, and the results are shown in Tables.2. and Tables.3.

Table 2. Balance between number of iterations and number of ants

Experimental group	Ant count	Iterations	Deadline count	Route change location
1	200	15	39	1, 26, 49
2	200	18	39	1, 25, 51
3	200	20	49	1, 26, 51
4	200	25	41	1, 26, 49
5	200	30	34	1, 28, 52

Table 3. Balance between number of iterations and number of ants

Experimental group	Ant count	Iterations	Deadline count	Route change location
1	100	20	29	1, 10, 32
2	150	20	30	1, 15, 33
3	200	20	49	1, 26, 51
4	250	20	49	1, 11, 35, 46
5	300	20	53	1, 22, 34, 52

As shown in the table above, when the number of ants is the same and the number of iterations is 20, the entire network has the longest cutoff time due to energy consumption. When the number of iterations is the same, the more ants there are, the more choices of reachable paths can be obtained. Considering the impact of factors such as time complexity of this algorithm, the parameter settings for this experiment were set to 200 ants and 20 iterations.

(2) Dynamic changes in operating results

Based on the selection of the above parameters, set the number of ants to 200, the number of iterations to 20, the number of nodes to 30, the starting node to 2, the search endpoint to 8, the number of times the sensor network transmits information to 30, and the initial energy of the nodes to 50. During the process of message transmission, due to changes in node energy, the running route undergoes three changes, as shown in Figure 1.

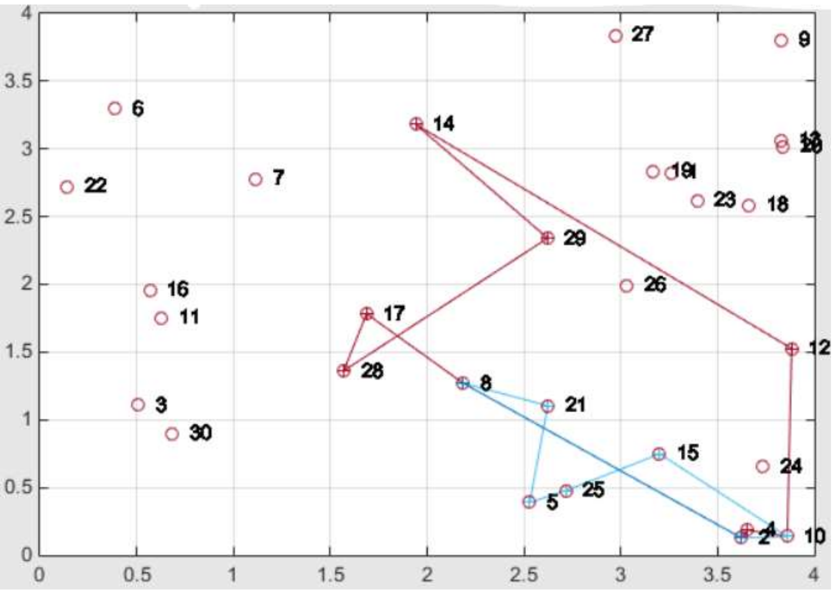

Figure 1. Node Dynamic Planning Diagram

When the energy is sufficient, the path for the algorithm to transmit information is $2 \rightarrow 8$.

When the remaining energy of a node reaches the set critical energy value after transmitting a certain number of times of information, the node changes its own route, and the selected route is: $2 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 5 \rightarrow 21 \rightarrow 8$.

When the energy is significantly insufficient, the selected route is: $2 \rightarrow 4 \rightarrow 10 \rightarrow 12 \rightarrow 14 \rightarrow 29 \rightarrow 17 \rightarrow 8$.

(2) Energy consumption balance with and without energy participation

After setting the parameters, experiments were conducted on algorithms with and without energy participation, and the data for two sets of partial nodes were obtained as shown in Tables.4 and Tables 5.

Table 4. Changes in node energy with energy participation

Nodes	Group 1	Group 2	Nodes	Group 1	Group 2	Nodes	Group 1	Group 2	Nodes	Group 1	Group 2
1	22	17	6	20	17	11	15	7	16	17	15
2	12	10	7	21	19	12	12	21	17	18	19
3	12	14	8	15	13	13	18	19	18	15	6
4	16	20	9	11	12	14	13	22	19	19	16
5	14	15	10	9	15	15	18	12	2	15	11

Table 5. Changes in node energy without energy participation

Nodes	Group 1	Group 2	Nodes	Group 1	Group 2	Nodes	Group 1	Group 2	Nodes	Group 1	Group 2
1	16	2	6	3	11	11	18	-1	16	14	3
2	1	0	7	18	-4	12	18	19	17	10	13
3	-12	15	8	9	15	13	10	-6	18	-4	-3
4	9	19	9	12	17	14	-20	-24	19	14	15
5	15	-17	10	9	11	15	-3	12	2	-13	3

From Figure 2, the remaining energy of each node iteration can be obtained. From the graph, it can be seen that nodes without energy participation have a negative residual energy value when the number of iterations reaches a certain level (nodes labeled 14 in the graph), which is not allowed in the algorithm. In path selection with energy participation, the remaining energy distribution of nodes is relatively balanced, and the network lifespan is significantly extended.

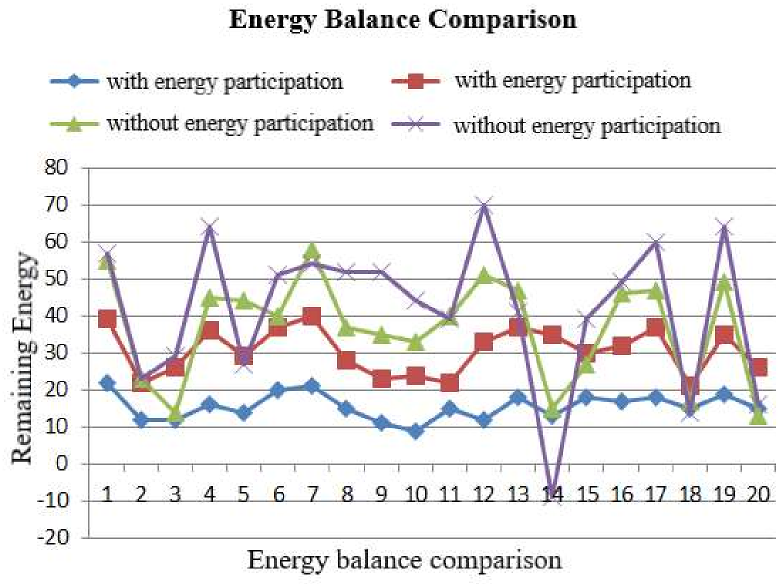

Figure 2. Energy balance comparison chart

4. Conclusion

The improved algorithm has shown outstanding performance in achieving self-organizing and dynamic protocol work in wireless sensor networks. By monitoring multiple nodes and transmitting node information into the Dijkstra algorithm, the algorithm protocol for finding routing and broadcasting paths in wireless sensor networks is completed. This article innovatively applies the intermediate value of ant colony algorithm, focusing on the balance between the number of ants and the number of iterations, and finds the intermediate variable through the balance of the two to connect with the Dijkstra algorithm. In addition, fusion algorithms use loops to simulate the transmission of each packet, simulating the entire process of a node from having life to receiving and sending packets, and then to death. In the process of forwarding packets by nodes, dynamic path planning was designed and implemented to address the limitations of path and self-energy, which extended the network lifespan.

References

[1] Vakirtzi E , Bayliss P . Towards a[J]. Journal of Philosophy of Education, 2013, 47(3):364-378.

[2] Hong Yi,Li Deying,Chen Zhibo. Constructing virtual backbone with guaranteed routing cost in Wireless Sensor Networks[J]. Ad Hoc Networks,2021,116.

[3] Lu Xinmiao,Su Yanwen,Wu Qiong,etc. An improved coverage gap fixing method for heterogenous wireless sensor network based on Voronoi polygons[J]. Alexandria Engineering Journal,2021,60(5).

[4] Shuai Mengxia,Yu Nenghai,Wang Hongxia,Xiong Ling,Li Yue. A Lightweight Three-Factor Anonymous Authentication Scheme With Privacy Protection for Personalized Healthcare Applications[J]. Journal of Organizational and End User Computing (JOEUC),2021,33(3).

[5] Naik A K J, Parameswarappa M, Ramachandra M N. Energy efficient data transmission using multiobjective improved remora optimization algorithm for wireless sensor network with mobile sink[J]. International Journal of Electrical and Computer Engineering (IJECE), 2023, 13(6): 6476-6488.

[6] Jemla Naik A K, Parameswarappa M, Ramachandra M N. Multiobjective, trust-aware, artificial hummingbird algorithm-based secure clustering and routing with mobile sink for wireless sensor networks[J]. ETRI Journal, 2024.

[7] Elyyan R R, Darabkh K A. A New IoT Power-Limited Wireless Sensor Networks Routing Protocol Utilizing Computational Intelligence[C]//2023 6th International Conference on Advanced Communication Technologies and Networking (CommNet). IEEE, 2023: 1-7.

[8] Nithya, D.A. Secure Routing using Multi-Objective Particle Swallow Swarm Optimization for Wireless Sensor Networks. 2023.

[9] Boyineni S, Kavitha K, Sreenivasulu M. Mobile sink-based data collection in event-driven wireless sensor networks using a modified ant colony optimization[J]. Physical Communication, 2022, 52: 101600.

[10] Divyashree H B, Puttamadappa C, KS N P. Hybrid Optimization algorithm for clustering and Routing in WSN[C]//2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). IEEE, 2022: 1-7.

[11] Shyjith M B, Maheswaran C P, Reshma V K. Optimized and dynamic selection of cluster head using energy efficient routing protocol in WSN[J]. Wireless Personal Communications, 2021, 116: 577-599.