J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 5511--5531

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Energy Efficiency Regulation Strategy of Distributed Sensor Networks Based on Graph Optimization Algorithm in Intelligent Buildings

Feifei Gao[1], Benyang Dou[2, ✉]

[1] *Department of Photovoltaic, Xuancheng Vocational & Technical College, Xuancheng, Anhui, 242000, China*

[2] *Administration of Technical Education, Xuancheng Vocational & Technical College, Xuancheng, Anhui, 242000, China*

ABSTRACT
Wireless sensor networks, which integrate a variety of technologies such as sensors, microelectromechanical systems, wireless communications, and distributed information processing, have become a cutting-edge field for studying the behavior of intelligent autonomous self-governing systems in groups. This paper explores distributed sensor networks in intelligent buildings, uses QoS routing algorithm based on ant colony optimization to implement the strategy of energy efficiency regulation of distributed sensor networks, and conducts experimental analysis on the performance of the algorithm as well as distributed sensor networks. Compared with the PCCAA algorithm, the node degree variance and channel percentage variance of this paper's algorithm are smaller, the network link distribution and channel allocation are more balanced, and the topology is better. Meanwhile, the average power of this paper's algorithm is slightly larger than that of the PCCAA algorithm, which is able to increase the robustness of the network while reducing the energy consumption and BER to ensure the network performance. In addition, the variance of the node energy consumption of this paper's algorithm in different networks is smaller than that of the PCCAA algorithm, which indicates that this paper's algorithm can make the node energy consumption of the whole network more balanced, and then improve the energy efficiency of the whole network. Simulation experiments prove that the algorithm in this paper effectively allocates node bandwidth through the quantization mechanism, thus reducing the amount of inter-node communication, while the corresponding sampling interval extension strategy can save the overall energy consumption of the network. The algorithm proposed in this paper has important practical value for energy efficiency regulation of sensor networks in intelligent buildings.

Keywords: Ant colony optimization algorithm, QoS routing algorithm, Distributed sensor network, Energy efficiency regulation

✉ Corresponding author.
E-mail address: dby_123@126.com (B. Dou).
Received 10 January 2024; Revised 15 February 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-308

1. Introduction

With the acceleration of urbanization and the continuous expansion of building scale, monitoring and remote control of building electrical equipment have become increasingly important [1]. However, traditional monitoring systems often have problems such as few monitoring points, difficult data collection, inconvenient remote control, and are difficult to meet the needs of large-scale buildings [2-3]. To solve this problem, IoT technology becomes a feasible solution. With its wide range of applications and flexible deployment methods, IoT technology brings new possibilities for monitoring and remote control of building electrical equipment [4].

Smart sensor networks have a wide range of application areas, such as industrial production, agricultural cultivation, environmental monitoring, etc., to realize informatization and intelligent management by sensing and interacting with the physical world [5-7]. Combining smart sensor network technology with building electrical systems is important for enhancing building energy efficiency, improving living experience, and realizing sustainable building development [8-10]. Smart sensor network is a distributed network system composed of a large number of miniature sensor nodes with functions such as data acquisition, information processing and wireless communication [11]. The sensor nodes form a multi-hop mesh topology through self-organization and transmit the collected data to the aggregation node for fusion processing [12]. The convergence node runs the data fusion algorithm, detects and corrects the abnormal data, and uploads the processed data to the cloud platform through the network to realize real-time monitoring and intelligent regulation of the building environment [13-15]. In the smart building electrical system, the wireless sensor network collects environmental data in real time, the central control system processes the information and issues commands, the actuators and control devices respond to the commands to regulate the building equipment, and the user interface provides human-computer interaction [16-19]. These units work closely together to ensure efficient operation of the system for building automation and energy optimization [20].

In this paper, we propose a sensor network multicast model based on sensor node graph location information, design a QoS routing algorithm based on ant colony optimization, and achieve energy efficiency regulation of distributed sensor networks by introducing MMAS model and energy warning model. To verify the effectiveness of the algorithm model, the performance of the algorithm is evaluated in terms of average node degree, node degree variance, percentage of channels occupied, average power of the network, and average energy consumption of the nodes, respectively. In addition, simulation experiments on distributed sensor networks demonstrate that the algorithm in this paper can effectively allocate node bandwidth and realize the regulation of the overall energy consumption of the network.

2. A model for graph optimization algorithms in distributed sensor network systems

In order to achieve the energy efficiency regulation of distributed sensor network system, this paper applies the QoS routing algorithm based on ant colony optimization to optimize the graph of distributed sensor network, and at the same time, in order to improve the algorithm's effect, it is improved by introducing the MMAS model and energy warning model.

2.1 Distributed Sensor Network System Model

The system model of the distributed sensor network is shown in Fig. 1. Assume that the distributed cluster exists n edge computing nodes with unknown states, whose set is $D=\{D_1, D_2, \dots, D_n\}$. These edge computing nodes assume the role of management nodes in the wireless sensor network, and each management node manages the training data collection of one or more sensor nodes, and at the same time, may contain multiple working nodes in the training state. In addition, there exists a set of independent indivisible heterogeneous tasks $Q=\{Q_1,Q_2,...,Q_m\}$ in the cluster that need to be scheduled, where m is the number of heterogeneous tasks and $m<n$, and for the j th heterogeneous task Q_i, its resource requirement can be denoted by r_j.

The distributed cluster considered in this study utilizes an online non-preemptive scheduling strategy, therefore, these heterogeneous tasks need to be scheduled for execution on edge computing nodes that have free resources. It is assumed that the heterogeneous tasks are the smallest execution units that cannot be further divided and are independent of each other, which means that these tasks do not have certain dependencies on each other and their inputs and outputs do not affect each other, so the execution process of the tasks can be parallel. In this study, a binary vector $a_j=\{a_{j,1},a_{j,2},...,a_{j,n}\}$ is used to represent the scheduling strategy of the j nd heterogeneous task Q_j in the distributed cluster, where when $a_{j,i}=1$ means that the j th heterogeneous task Q_j is executed by the i th edge computing node D_i, and when $a_{j,i}=0$ means that the i th edge computing node D_i is not the execution location of the j th heterogeneous task Q_j. Without loss of generality, this study assumes that the execution resources required for these computing tasks include, but are not limited to, processor resources, power energy, storage resources, and so on.

In this study, the binary vector $c=\{c_1,c_2,...,c_n\}$ is used to represent the result vector of all edge computing nodes performing the training task, where the binary variable c_i represents the result of the i rd edge computing node D_i performing the training task, when $c_i=1$ means that the i th edge computing node D_i received the training request from the parameter server and no packet error occurred during the parameter communication, and vice versa $c_i=0$.

In the distributed training process, for the i st edge computing node D_i, this study uses s_i to denote the size of its locally collected training data from the sensor nodes, and if this node receives a training request from the parameter server, the weight vector obtained from the training of its local model can be denoted as $w_i=\{w_{i,1},w_{i,2},...,w_{i,s_i}\}$. This study adopts the distributed training framework, and assumes that the parameter server in the training process of the global model outputs the model parameters as w, due to the existence of some edge computing nodes in the distributed cluster that do not perform training tasks, let:

$$g_i=1-\sum_{j=1}^{m}a_{j,i} \tag{1}$$

Then, for the distributed machine learning framework mentioned in this study, the global model parameters in this study can be expressed as follows, under the consideration of heterogeneous task scheduling strategies:

$$w(a)=\frac{\sum_{i=1}^{n}s_ic_iw_ig_i}{\sum_{i=1}^{n}s_ic_ig_i} \tag{2}$$

where $a=\{a_1,a_2,...,a_m\}$ denotes the vector of scheduling strategies for all heterogeneous tasks. Under certain other factors such as the parameter update algorithm, the convergence efficiency of distributed training is related to the loss function in the learning task, implying that it can be approximated that the maximization of the training performance represents the minimization of the value of the loss function of the global model at the time of global model convergence. Therefore, after obtaining the global model parameters of the distributed machine learning system, the distributed training performance optimization objective of this paper regarding the heterogeneous task scheduling strategy a can be initially written as:

$$\min_a \frac{1}{s}\sum_{i=1}^{n}\sum_{k=1}^{s_i} c_i f(w_i(a_i)) \tag{3}$$

Where, $s=\sum_{i=1}^{n} s_i$ denotes the scale size of the data volume collected by all edge nodes of this distributed cluster, $w_i(a_i)=s_i c_i w_i g_i$ denotes the training parameters of the local model of each edge node, and $f(w_i(a_i))$ is the loss function of the local model.

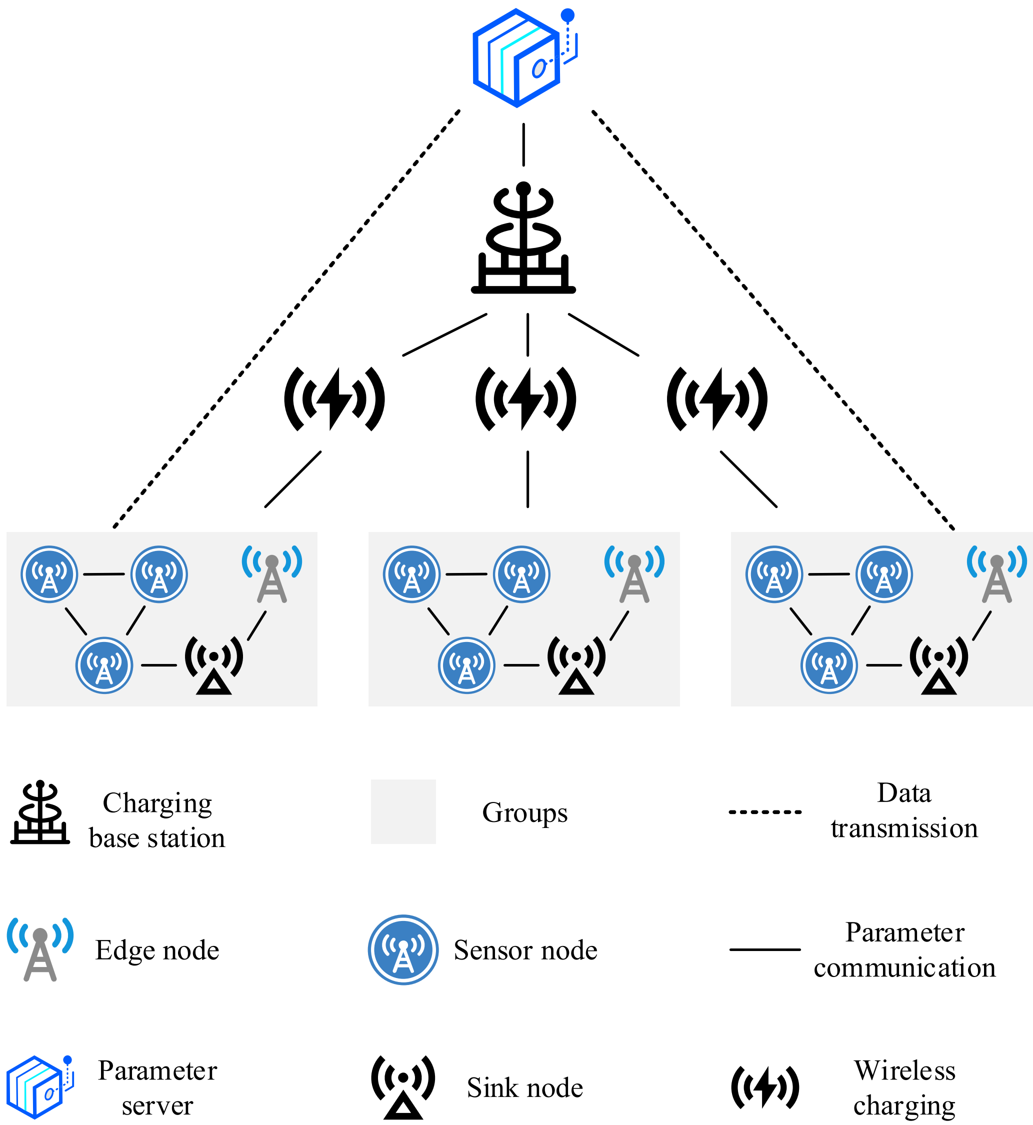


Fig. 1. System model

2.1.1. Communication systems. When the parameter server sends a training task to a specified working node in the distributed cluster, the probability of successful transmission is directly related to the probability that the parameter server selects that node with the network signal-to-noise ratio, network bandwidth and other factors. For the i st edge computing node D_i, this study first analyzes the transmission rate between the edge node and the parameter server, i.e.:

$$v_i = b_i \ln\left(1+\frac{\alpha_i p_i}{\chi^2}\right) \tag{4}$$

Where, α_i and p_i denote the channel gain and communication power, respectively, for communication between the edge computing node D_i and the parameter server, χ is additive Gaussian white noise, and b_i denotes the communication bandwidth between the edge computing node D_i and the parameter server.

In this study, we consider φ_i to denote the probability of D_i successfully performing the training task, i.e:

$$c_i = \begin{cases} 1, & 1-\varphi_i \\ 0, & \varphi_i \end{cases} \tag{5}$$

Assuming that the probability that the parameter server is choosing edge node D_i is δ_i, there is:

$$\varphi_i = \delta_i \left\{1-\exp\left[-\upsilon(\varpi_i+\chi^2)\right]\right\} \tag{6}$$

Among them:

$$\varpi_i = \frac{\sum_{i=1}^{n} \partial_i p_i a_{j,i}}{\partial_i p_i} \tag{7}$$

That is, the proportion factor of noise interference caused by other edge computing nodes performing heterogeneous tasks to the parameter communication process in the distributed machine learning system, with υ being the waterfall threshold.

2.1.2. Computing systems. The processing capability of an edge node for a heterogeneous task is related to the node's CPU clock frequency, clock period, and the size of the amount of resources it can provide, and the size of the amount of data processed by the heterogeneous task also has a large impact on the computational latency of the task. Assuming that the j st heterogeneous task Q_j is executed on the i rd edge node D_i, for D_i the task execution time $t_{j,i}^{he}$ is given:

$$t_{j,i}^{he} = \frac{\tau_j}{r_{j,i}} \tag{8}$$

where τ_j denotes the number of CPU clock cycles required to execute the heterogeneous task Q_j and $r_{j,i}$ denotes the computational resources D_i provided to Q_j. For the energy loss $E_{j,i}$ required for Q_j to execute on D_i, there is:

$$E_{j,i} = \vartheta_i \tau_j r_{j,i}^2 \tag{9}$$

where ϑ_i is the effective capacitance factor associated with the chip structure of the edge node.

Assuming that the i 2nd edge node D_i performs a training task, the execution time $t_{0,i}^{tr}$ of a single iteration of the training task on D_i can be expressed as:

$$t_{0,i}^{tr} = \frac{s_i}{r_{0,i}} \tag{10}$$

where $r_{0.i}$ denotes the computational resources provided by D_i to the training task. For the energy loss $E_{0,i}^{tr}$ required for a single iteration of the training task on D_i, there is:

$$E_{0,i}^{'r} = \vartheta_i s_i r_{0,i}^2 \tag{11}$$

For distributed machine learning algorithms with loss functions that are strongly convex, a general upper bound on the number of global iterations can be expressed as:

$$K(\gamma,\varepsilon)=\frac{O(\log(1/\gamma))}{1-\varepsilon} \tag{12}$$

where γ and ε are the training accuracies of the global and local models, respectively. From (12), the number of global model iterations after regularization can be approximated as $\frac{1}{1-\varepsilon}$ for a certain γ. And the number of local model iterations in a global iteration is related to its training accuracy ε, which can be approximated as $\log\left(\frac{1}{\varepsilon}\right)$. Thus, for the computational energy consumption $E_{0.i}$ required for the training task to be executed on D_i, there is:

$$E_{0.i}=\frac{\log\left(\frac{1}{\varepsilon}\right)E_{0.i}^{tr}}{1-\varepsilon} \tag{13}$$

2.1.3. Convergence efficiency analysis. The local model parameter updating algorithm used in this study is the standard gradient descent (SGD) algorithm. Therefore, denoting the learning rate of the training model in the distributed machine learning system as λ, and assuming that in the 1st global iteration, $\nabla f_k(w^{'}(a_i))$ is the gradient of the global model $f_k(w^{'}(a_i))$ with respect to the global model weights $w^{'}(a_i)$, the local model parameter updating method in this study can be expressed as:

$$w_i^{'+1}=w_i^{'}-\frac{\lambda}{s_i}\sum_{k=1}^{s_i}\nabla f_k(w^{'}(a_i)) \tag{14}$$

From the above, it can be seen that in a distributed machine learning system, the global model can be represented as $F(w)=\frac{1}{s}\sum_{i=1}^{n}f_i(w)$ and the local model can be represented as $f_i(w)=\sum_{k=1}^{s_i}f_k(w)$, where w is a shorthand form of $w(a_i)$. It can be seen that in the ith global iteration, the way of updating the parameters of the global model can be denoted as:

$$w^{'+1}=w^{'}-\lambda(\nabla F(w)-\xi) \tag{15}$$

Among them:

$$\xi=\nabla F(w^t)-\frac{\sum_{i=1}^{n}s_ic_iw_ig_i}{\sum_{i=1}^{n}s_ic_ig_i} \tag{16}$$

2.2 Multicast model for sensor networks based on graph position information

In the multicast model for sensor networks based on graph position information, we use the total number of hops required for data delivery as an evaluation metric for routing performance. Instead of using addresses as identification IDs, nodes in the sensing network are addressed based on the data that the node can provide. That is, a Sink broadcasts a message in the network consisting of a certain data format asking for the monitoring data it is interested in, which is referred to as interest. The node that matches this interest (called the source node) responds to this query and sends back monitoring data to the Sink. Since there may be more than one Sink with the same interest and there may be more than one node that matches the same interest, these nodes that match the same interest send back their own monitoring data to all the Sinks that have that interest.

Is shown in Fig. 2. Using this multicast model, the specific implementation process of multicast routing is as follows: assume that the destination nodes sink1 and sink2 have the same interest Q, and through the information exchange between neighboring nodes, the sensing nodes B and C learn that they match the interest Q, where B and C belong to the same cluster, whose cluster head is A. Then B and C first use the sensing unit to collect the raw data of the monitored object, and then both send the data to A. A, after aggregating and merging all the collected raw data using the processing unit, then calculates the location of node D using the location information, and then uses the ant colony algorithm to distributedly find an actual path from A to D as well as from D to Sink1 and Sink2, respectively, such that the sum of the number of hops from A to D hops plus the sum of hops from D to nodes Sink1 and Sink2 is minimized. A then sends the aggregated data uniformly to D. D then, after making a copy of the data, sends the two copies of the data to Sink1 and Sink2, respectively, along the actual paths found from D to Sink1 and sink2. Finally, nodes Sink1 and Sink2 are then transmitted via the Internet or by satellite to the user data processing center.

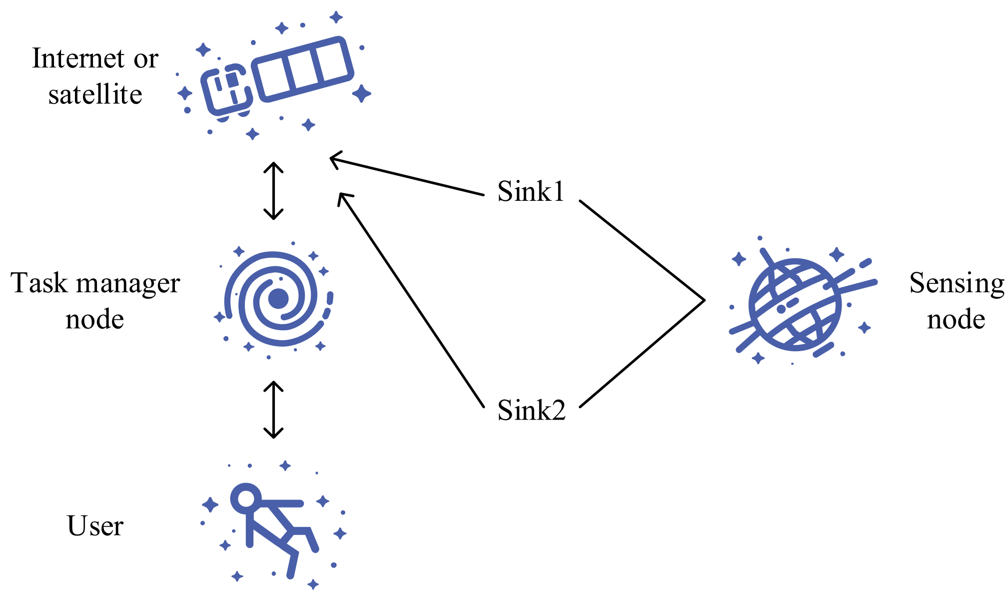


Fig. 2. Multicast model of sensor network based on location information

Based on the above description, a formal description of the multicast model for sensor networks based on graph location information is given below. Assuming that Source node Source in the sensing network has data to send to Sink node $\text{Sin}\,k_1, \text{Sin}\,k_2, \ldots \text{Sin}\,k_m$, then based on the total number of hops required for data sending as the evaluation index of routing performance, it can be seen that our goal is to find an optimal intermediate point P such that P satisfies the condition:

$$Minimum\{Hops(Source, P) + \sum_{j=1}^{m} Hops(P, Sinkj)\} \tag{17}$$

where $Hops(A, B)$ denotes the minimum number of hops between nodes A, B.

Assuming that the Sink node and the sensing node have the same effective communication radius r. Since the nodes of the sensing network are densely distributed, it can be assumed that node A can reach node B along a straight line path, so $Hops(A, B)$ can be estimated according to equation (18):

$$Hops(A, B) = Dist(A, B) / r \tag{18}$$

where $Dist(A, B)$ denotes the straight line distance between nodes A, B.

Since the sensing nodes can use the GPS positioning function to get their own geographic coordinates location information, assuming that the coordinates of nodes A, B are $A(x_1, y_1), B(x_2, y_2)$, respectively, the straight line distance $Dist(A,B)$ between A, B can be derived according to equation (19):

$$Dist(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{19}$$

From the above description, it is easy to see that the crux of the problem is how to find the coordinates of the optimal intermediate point P. Abstracting the problem, we can see that it is equivalent to the following problem: that is, for a given M points $A_1(x_1, y_1), A_2(x_2, y_2), \ldots A_M(x_M, y_M)$, to find the point $P(x,y)$ so that $\sum_{i=1}^{M} Hops(P, A_j)$ is minimized.

Obviously, the center of gravity of $A_1(x_1, y_1), A_2(x_2, y_2), \ldots A_M(x_M, y_M)$ is an approximate solution that satisfies the condition, so the coordinates of point $P(x,y)$ can be determined based on the following equations (20)~(21):

$$x = (\sum_{j=1}^{M} x_j) / M \tag{20}$$

$$y = (\sum_{j=1}^{M} y_j) / M \tag{21}$$

2.3 Energy efficiency regulation of QoS routing algorithms based on ant colony optimization

2.3.1. Ant colony optimization algorithm and its process. Ant colony optimization is an intelligent optimization model modeled after the biological characteristics of ant colony foraging, which can efficiently and reliably deal with complex combinatorial optimization problems [21]. With its high search efficiency and low time complexity, it is one of the best algorithms to find the optimal solution, and nowadays it has been widely applied and promoted in many fields, such as combinatorial optimization, artificial intelligence, communication data mining, and robot path planning. Since the structure of this algorithm has the same characteristics as the wireless sensor network structure, such as pan-self-organization, randomness, dynamics and distribution, many researchers are keen to introduce the ant colony optimization technique into wireless sensor network applications.

The steps to run the ant colony optimization algorithm are as follows:

Step 1: Variable initialization.

$n_c = 0$ (initialize the number of iteration steps n_c).

$\tau_{ij}(0) = c, \Delta\tau_{ij} = 0$ (initialize pheromone concentration on all paths).

All ants are randomly distributed across cities.

Step 2: Place m ants on n cities. $tabuk(s)$ Record the initial position of each ant}.

Step 3: m ants choose the next target city by probability function.

For any ant $k(k = 1, 2, \ldots., m)$ select the next hop with $p_i^k(t)$ probability.

Add $tabuk(s)$ path information.

Step 4: Record the best route for this iteration

The k-ant completes all traversals, calculates the total path L_k based on $tabuk(s)$, and updates the current shortest path.

Step 5: Update pheromone

Update the path pheromone $\tau_{ij}(++n)$ according to $tabuk(s)$ and L_k.

All ants complete the traversal, set all edges to $\Delta\tau_{ij} = 0; n_c = n_c + 1$, and one iteration is completed.

Step 6: Output shortest path

When $n_c < NCMAX$ jump to step 2. When $n_c \geq NCMAX$, output the result and finish.

2.3.2. QoS Routing Model for Wireless Sensor Networks. The WSN topology is available as a weighted undirected graph $G(V,E)$, the set of nodes: $V = \{v_1, v_2, \ldots, v_n\}$, the number of nodes is

$n, delay(v_i)$ is the node QoS metric, $loss(e_i)$ denotes the link QoS metric, $E=\{e_1,e_2,...,e_m\}$ is the set of edges, m is the number of edges, and the path from the sensor node to the sink node is $p(s,\sin k)$ [22]:

$$delay(p(s,\sin k)) = \sum_{v\in p(s.\sin k)} delay(v) \tag{22}$$

$$loss(p(s,\sin k)) = \sum_{e\in p(s,\sin k)} less(e) \tag{23}$$

The QoS constraints are: path delay constraint, packet loss rate constraint.

Condition Requirements:

1) $packetloss(p(s,\sin k)) \le L$, L is the maximum loss rate.
2) $delat(p(s,\sin k)) \le D$, D are the maximum delay values.

Assuming that all sensor nodes have sink as their destination node, L denotes the number of neighboring nodes of node a. All nodes jointly maintain pheromone table K_n, then:

$$\sum_{i/L} P_i = 1 \quad L=\{neightbors(a)\} \tag{24}$$

p_i denotes the probability of selecting the next-hop node as i when node a reaches the sink, and the set of neighboring nodes $neightbors(a)$.

2.3.3. QoS routing algorithms. $s(s\in V)$ Node periodically sends artificial $F_{s\to\sin k}$ ants to the sink $F_{s\to\sin k}$ with the same packet priority and the same routing search path rules; $F_{s\to\sin k}$ record the delay and cost, $F_{s\to\sin k}$ when it reaches the sink, $F_{s\to\sin k}$ ants save the delay information on the path to the stack:

$$\{(s, delay(j_1), loss(h_1)), ..., (j_k, delay(j_k), loss(h_k))\} \tag{25}$$

where $(h_1,h_2,...,h_k)$ is the set of links between nodes and $p(s,\sin k)=\{s,j_1,j_2,...,j_k,\sin k\}$ is a node on the $F_{s\to\sin k}$ path.

Correction $F_{s\to\sin k}$ ant the link:

When $k\in p(s,\sin k)$, the next hop $h, h\in(s,j_1,j_2...,j_{k-2})$ is selected, all data between nodes h and k is removed, $F_{s\to\sin k}$ forwarded to node h, and the next hop is selected.

When $k\in p(s,\sin k)$, the next jump points $h, h\in\{neighbors(n)\}$, $n\in\{s,j_1,j_2,...,j_{k-2}\}$ are selected, removing the data between n and k, and the path is $\{s,j_1,...,n,h,...,j_k,\sin k\}$.

When $F_{s\to\sin k}$ reaches the sink node, immediately release the backward $B_{\sin k\to s}$ ants, $B_{\sin k\to s}$ acquire the path information stack, and $F_{s\to\sin k}$ release the resources when the ants die. The highest priority $B_{\sin k\to s}$ returns to the starting position along the path table, sequentially updating the pheromone table of the nodes along the path.

Suppose $B_{\sin k\to s}$ ants choose the set of nodes on the path $p(s,\sin k)=\{s,j_1...,a,j_i,j_{i+1},...,j_k,\sin k\}$, T^a is the delay estimation table for a nodes, and D^a is the cost estimation table for evaluating the path $p(a,\sin k)=\{a,j_i,j_{i+1},...,j_k,\sin k\}$. Wanting to make the delay estimation table T^a and the cost estimation table D^a efficient, a balanced approach is used for the computation of D^a and T^a. $B_{\sin k\to s}$ comes to node a and updates T^a and D^o:

$$T^a(j_i) = (1+\alpha)T^a(j_i) + \alpha * delay(p(a,\sin k)) \tag{26}$$

$$D^a(j_i) = (1+\beta)D^a(j_1) + \beta * loss(p(a,\sin k)) \tag{27}$$

The queue length determines the waiting delay, which is a response to link congestion and is typically $0.7\le\beta\le 0.9$. The learning factor α,β, the transmission capacity of the link and the processing speed of the node determine the transmission delay and is typically $0.2\le\alpha\le 0.4$.

$B_{\sin k\to s}$ updates all pheromone tables on the path on the way back to the source node.

The pheromone increment of node a:

$$\Delta p = \frac{e^{-\delta \times loss(p(a,\sin k))/D^a(a)}}{1+\varepsilon \times \ln(delay(p(a,\sin k))/T^a(a))} \tag{28}$$

In Eq. (28), the influence factor is δ,ε, taken as $3 \le \delta \le 5, 0.8 \le \varepsilon \le 1.2$, and there are ratios of $loss(p(a,\sin k))$ and $D^a(a)$ that are inversely proportional to the pheromone increment. The ratio of $delay(p(s,\sin k))/T^a(a) < 1, loss(p(a,\sin k))$ and $D^a(a)$ is proportional to the pheromone increment.

Update the pheromone table equation for a node:

The a node selects the neighbor J_1 node to the convergence node:

$$p_{j_1} = \frac{p_{j_1} + \Delta p}{1+\Delta p} \tag{29}$$

When $j \in \{neighbors(a)\}$, and $j \ne j_1$, there is:

$$p_j = \frac{p_j}{1+\Delta p} \tag{30}$$

It can be shown that the constraints are still satisfied after the pheromone table is updated.

2.3.4. Improvement of the algorithmic model:

1) Introduction of MMAS model. The optimal path is often affected by the wireless network characteristics and changes, but the artificial ants in the original algorithm are still stuck in the original equilibrium, and it is difficult to jump out and carry out the search for the optimal path in a short period of time. The MMAS model [23] can exactly solve this problem, which is introduced in this paper and the algorithm is improved as follows:

(1) Limit the maximum selection probability by decreasing the pheromone concentration by making the following judgment before $B_{\sin k \to x}$ updating the pheromone table for the data link: if $p_{j_1} > p_0, p_0 \in (0,1)$ that indicates that the selection probability for the link increases by increasing the pheromone concentration, then only $B_{\sin k \to s}$ updating the two tables of the cost estimation and the delay estimation is allowed, and no updating can be made to the pheromone table.

(2) Suboptimal path diversion method

After the packet reaches the aggregation node (sink) from any source node, it selects the next hop node probability:

$$p_j = \begin{cases} 1/L_a, p < p_1 \\ K_a \end{cases} \tag{31}$$

Where: a is any source node. $j, j \in \{neighbors(a)\}$ is the next hop node. p_1 is a random factor, $p_1 \in (0,1)$. L_a is the number of neighbor nodes of node a. p is any number between (0, 1).

Set the random factor p_1 to control some artificial ants to explore the new path randomly.

Conclusions are drawn from simulation experiments:

(1) $p_0 \ge 0.9$: The data traffic in the network is concentrated towards one link, and the chance of the algorithm forming a locally optimal solution increases.

(2) $p_0 \le 0.7$: The data traffic in the network is dispersed to different links, but it is close to the average probability distribution, and the selected path does not reach the optimal solution.

(3) $0.7 \le p_0 \le 0.9$: The data traffic in the network is reasonably dispersed to various links, and when the network state is changed, the most existing paths can be quickly abandoned, and the existing paths are replaced by suboptimal paths.

2) Introduction of energy warning model. Due to the limited, and non-replenishable energy of sensor nodes, the nodes will fail early when they are overused or when they run out of energy due to other reasons. The residual energy of a node is the best predictor of the network life cycle, in order to ensure

the smooth operation of the network, we must protect the residual energy of a node before it runs out of energy, so we introduce the energy warning model.

Energy Alert Model:

The probability formula for selecting the next hop node after any source node reaches the aggregation node is as follows:

$$p_j = \begin{cases} p_j * 0.95, w_j \le w_0 \\ p_j, \text{other} \end{cases} \tag{32}$$

In Eq. (32), a is any source node. j is the next hop node, $j \in \{neighbors(a)\}$. w_j is the residual energy value of node j, and w_0 is the node energy threshold.

If a node on the link is found to have a residual energy less than the given warning threshold, then the data transmission traffic of the link in the later time period is reduced by lowering the pheromone of the corresponding node on the link as a way of protecting the nodes in the network whose energy is going to be exhausted from failing.

3. Model simulation experiments and performance evaluation

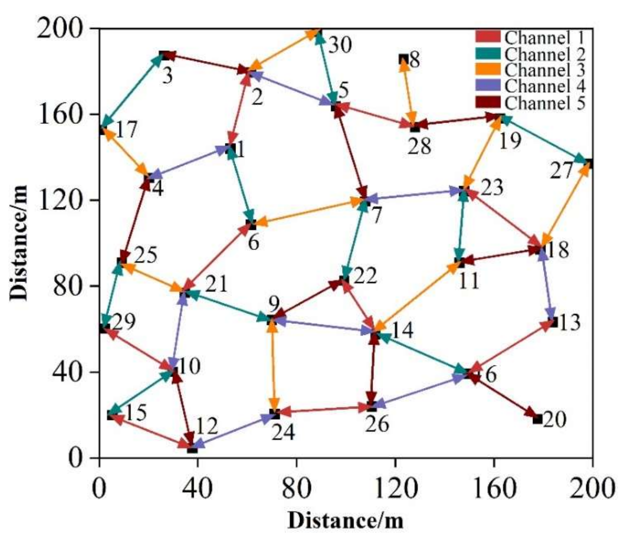
In this chapter, the proposed QoS routing algorithm model based on ant colony optimization is simulated and experimented with performance evaluation in MATLAB environment, whereas the PCCAA algorithm comprehensively considers the joint optimization of node's power control and channel allocation, and optimally selects the power and channel of all the nodes in the network, and the final results show that the algorithm effectively reduces the network interference and energy consumption. Therefore, the PCCAA algorithm is used as the comparison algorithm of this paper, and the performance of the PCCAA algorithm is analyzed and compared with the performance of this paper, so as to validate the reasonableness, feasibility, and effectiveness of the distributed algorithm proposed in this paper. The experimental simulation area of the WSN is a square of 200m×200m, and the experimental parameters of the two algorithms are the same, and the other parameter settings are shown in Table 1. Other parameter settings are shown in Table 1.

Table 1. Simulation parameters

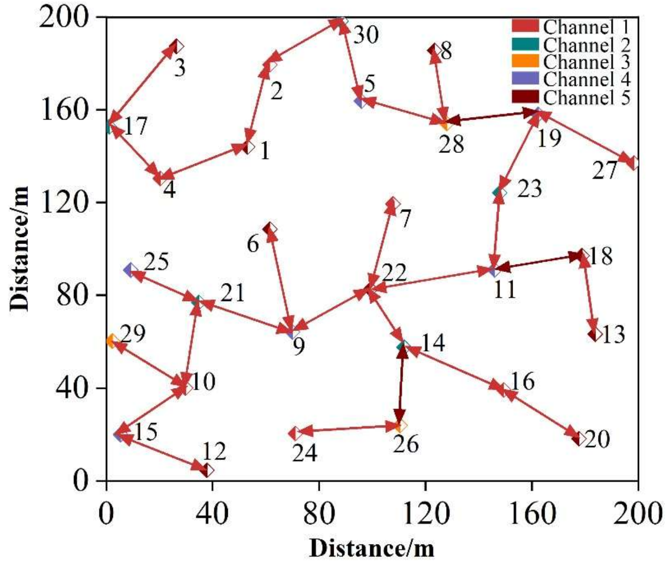
Parameters	Value	Parameters	Value
Transmission power threshold $p_{th}(w)$	7.2×10^{-7}	Maximum transmission radius $d_{\max}(m)$	60
Channel bandwidth $B(kHz)$	25	Signal to noise ratio threshold $r_{th}(dB)$	7
Noise bandwidth $B_N(kHz)$	40	White noise $N_0(w)$	1.1×10^{-7}
Data generation rate $r(kb/s)$	25	The total number of available channels	5
Maximum number of iterations K	300	Population size pop	80
Inertia factor w	0.9	EA size	25
Learning factor c_1	0.8	Learning factors c_2	0.5

In order to get the topology graph and the final channel allocation of the network, 30 nodes are randomly arranged in an area of 200m×200m, and the algorithm of this paper and the PCCAA algorithm are executed to get the topology graph and the channel allocation state of the network, respectively. The final results of topology and channel allocation are shown in Fig. 3, where Fig. (a) and Fig. (b) represent the topology and channel allocation of this paper's algorithm and PCCAA algorithm, respectively.

From the channel allocation state obtained by the PCCAA algorithm in Fig. 3(b), it can be seen that neighboring nodes in the network are allocated different channels, this is because the algorithm comprehensively considers and analyzes the information of the neighboring nodes, but the algorithm is designed to allocate channels to the nodes, so the switching of the channels consumes more energy of the nodes during the process of sending and receiving data. In addition, the topology graph constructed by the PCCAA algorithm is sparse, and the whole network is more likely to be disconnected when the network encounters interference and attacks. Meanwhile, from Fig. 3 (a), the topological links constructed by this paper's algorithm are denser and more uniform than the PCCAA algorithm, so that when a node in the network is unable to communicate due to the reduction of its energy, the other links can communicate efficiently in order to avoid the failure of data transmission in the WSN. This is due to the fact that in the process of channel allocation and power control, the algorithm in this paper takes into account the topological state of the network, at this time, if the neighboring links choose the same channel, the BER function value and the energy consumption function value of the nodes in the network will increase, in order to reduce the function value of the nodes, in this paper's algorithm, the neighboring links will try to avoid choosing the same channel, and the channel allocation is more balanced.



(a) This algorithm



(b) PCCAA algorithm

Fig. 3. Topology and channel allocation

In addition, the network performance of this paper's Qos routing algorithm based on ant colony optimization is evaluated by experimental simulation for networks with consistent deployment node profiles, and compared and analyzed with the PCCAA algorithm. MATLAB is used to realize the simulation comparison of each performance. These network performances include network capacity, BER and energy consumption, in addition to this the node degree and channel equalization of the network is also analyzed.

3.1 Comparison of mean node degree and node degree variance

Randomly distributing 30 to 100 nodes in a 200m×200m square with a channel number of 5, and then executing the PCCAA algorithm and this paper's QoS routing algorithm based on ant colony optimization, respectively, the comparative results of the network's average node degree and node degree variance can be obtained as shown in Fig. 4. In Fig. 4, the lengths of the purple and green lines indicate the magnitude of the node degree variance in the network. The size of the average node degree in the network represents the denseness of the communication links and the degree of damage resistance of the network. In the topology construction process of WSN, the smaller the node degree variance indicates that the node degree of all nodes in the network has a smaller difference, the closer they are to each other, and the more uniform the constructed topology is.

From Fig. 4, it can be seen that the average node degree of this algorithm is larger than that of the PCCAA algorithm, which is due to the fact that the algorithm in this paper takes into account the performance of the network when constructing the topology. Therefore, when the network external interference is too large and some nodes fail, the algorithm of this paper with a larger average node degree can ensure that the network is not partitioned and communicates normally, thus prolonging the network lifetime. In addition, from the comparison results, it can be seen that this paper's algorithm has a smaller variance of node degree than the comparison algorithm PCCAA, the network link distribution is more balanced, and the topology structure is better.

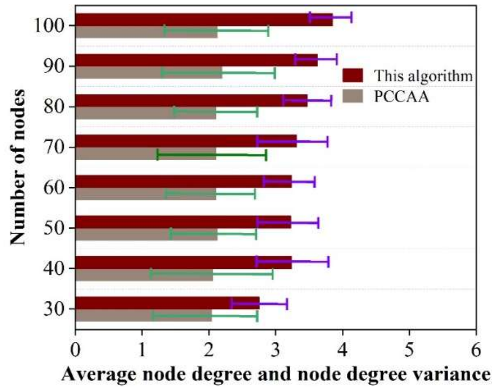
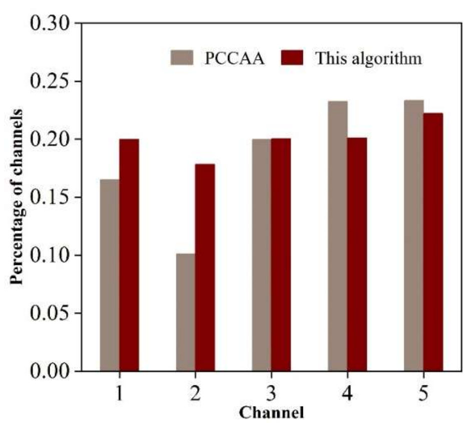


Fig. 4. Comparison between average node degree and node degree variance

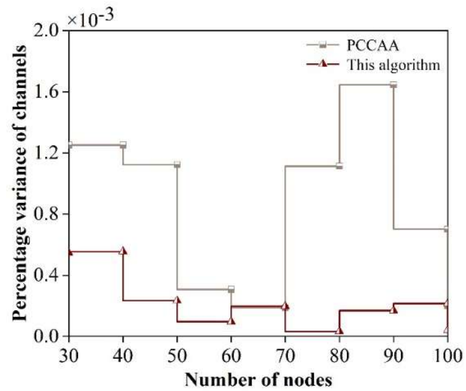
3.2 Comparison of channel equalization

In the process of channel allocation for WSNs, the magnitude of channel equalization is an important criterion to measure whether the channel allocation in the network is reasonable or not. In the case of limited channel resources, if the channel allocation is not balanced, then it will lead to multiple links or nodes will use the same channel for data transmission, in this case, the nodes will interfere with each other, conflict with each other, affect each other, and the interference of the whole network will increase, which will lead to more energy loss as well as data loss. So in order to achieve balanced utilization of spectrum resources, the utilization of channels needs to be balanced to enhance the interference resistance of the network. The results of channel equalization comparison between this paper's algorithm and comparing PCCAA are shown in Fig. 5. Fig. 5(a) represents the percentage of channels of this paper's algorithm and PCCAA algorithm in this experimental phase, setting the number of network nodes as 40 and the number of available channels as 5. Fig. 5(b) represents the channel occupied percentage variance of this paper's algorithm and PCCAA algorithm during the number of channels is 5 and the number of nodes increases from 30 to 100. The channel percentage share variance indicates the equalization of channel allocation in the network.

From Fig. 5(a), it can be seen that the difference in the percentage of channels in this paper's algorithm is not large and is relatively smooth, and the channel allocation is well balanced. According to Fig. 5(b), it can be seen that the percentage variance of channels in this paper's algorithm is smaller than that of the PCCAA algorithm in the process of channel allocation, and the experimental results show that the channel allocation in the network topology formed by this paper's algorithm is more balanced, and the effectiveness of the spectrum resources is stronger.



(a) Percentage of channels



(b) Percentage variance of channels

Fig. 5. Channel balance comparison

3.3 Comparison of network average power

In the simulation phase of this experiment, the number of channels is set to 5, and the average power of the whole network can be obtained when the number of nodes in the network is changed from 30 to 100, and the results of the comparison of the average power between the algorithm in this paper and the PCCAA algorithm are shown in Fig. 6. The shaded part of the figure indicates the difference between the average power of the two algorithms.

In WSN, the size of the average power of the network directly affects the performance of the network, when the average power is too large, the nodes consume too much energy and the life cycle will be shortened. When the average power of the network is small, it leads to decrease in network capacity and poor robustness. Therefore, the appropriate transmit power can effectively improve the performance index of the network. As can be seen from Fig. 6, the average power of this paper's algorithm is slightly larger than that of the PCCAA algorithm, which is due to the fact that this paper's algorithm comprehensively investigates the objectives of network energy consumption and BER, and takes into account the interference of the network and the robustness of the network. Also due to the fact that the PCCAA algorithm assigns power and channels to nodes based on their residual energy, the resulting power is the minimum power that maximizes the benefit function while ensuring network connectivity. Therefore it can be concluded that the algorithm in this paper is able to increase the robustness of the network while reducing the energy consumption and BER and ensuring the performance of the network.

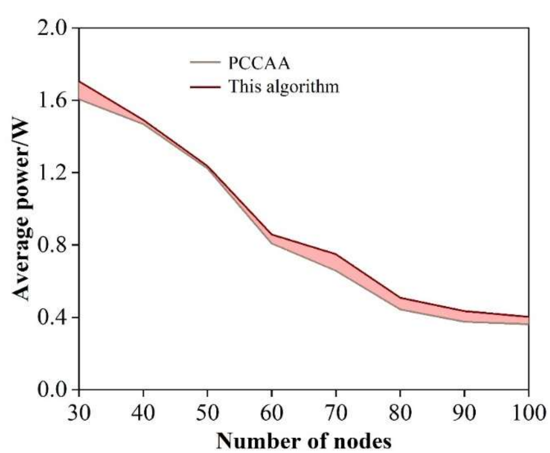


Fig. 6 Average power comparison

3.4 Comparison of energy consumption

This experiment compares the energy consumption of each node in the network under the PCCAA algorithm and this paper's algorithm and the variance of energy consumption under each algorithm, showing the ability of the two algorithms to reduce and equalize the energy consumption. 40 nodes are randomly deployed in a simulation area of 200m×200m, with the number of channels being 5, and the results of the comparison of the node's energy consumption and the comparison of the variance of the energy consumption are shown in Fig. 7 and Fig. 8, respectively.

Figure 7 shows that the energy consumption of some nodes in the network under this paper's algorithm is smaller than that of the node under the PCCAA algorithm, because the topology constructed by this paper's algorithm is denser, so the transmit power is relatively slightly larger, and the node energy consumption is also relatively slightly larger. As can be seen from Fig. 8, the variance of the node energy consumption of this paper's algorithm is smaller than that of the PCCAA algorithm in different networks with 40-80 nodes, and the energy consumption of each node in the network is more balanced. Therefore, this paper's algorithm can make the node energy consumption of the whole network more balanced, which in turn improves the energy efficiency of the whole network.

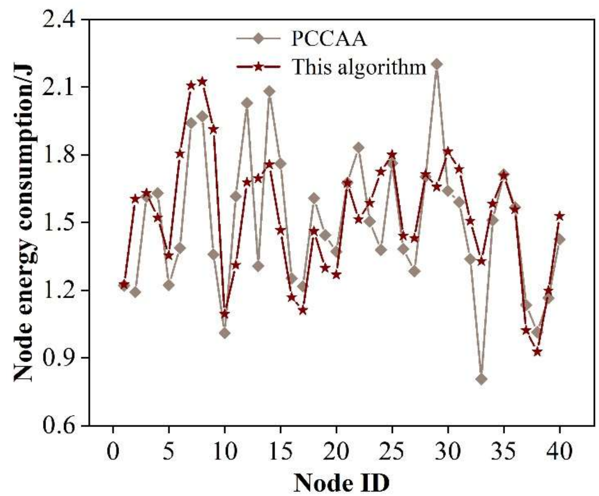


Fig. 7. Node energy consumption comparison

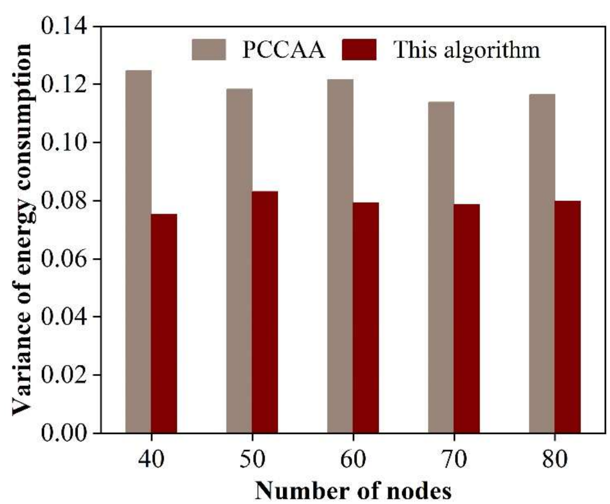


Fig. 8. Comparison of variance of energy consumption

4. Simulation analysis of energy efficiency regulation of distributed sensor networks

In order to illustrate the effect of quantization mechanism and sampling interval on the tracking performance of wireless sensor networks during target tracking, this paper simulates a certain target tracking with ranging sensor nodes randomly deployed in a monitoring area of 60×40m^2, and the main simulation parameters are shown in Table 2.

Table 2. Simulation parameters

Parameter name	Numerical value
Number of nodes	120
Node perception radius	9
Node noise variance	0.02
Test duration	24.9s
Sampling interval	[0.1s,0.5s]
Φ_0^1	1
Φ_0^2	1
M	1

It is assumed that the target is detected by the sensor nodes in the monitoring area at time $t=0$ and the initial state is $x_0=(30,-2,30,0)^T$, where the symbols denote the direction of velocity and the target travels along the "sickle-shaped" trajectory.

The tracking effect and root mean square error comparison between quantized and unquantized node observations are shown in Fig. 9 and Fig. 10, respectively. Due to the limited energy of wireless sensor network nodes, it is necessary to minimize the energy consumption under the premise of guaranteeing the accuracy. The purpose of saving energy can be realized by quantifying the observations of nodes and reducing the amount of data transmission between networks.

From Fig. 9 and Fig. 10, it can be seen that under the constraint of the tracking threshold, the tracking effect after quantization is not as good as when there is no quantization, both in terms of the estimation of the trajectory and the root-mean-square error of the tracking. However, in order to save energy, the observations are first constrained to be quantized during the tracking process, mainly to reduce energy consumption at the expense of accuracy.

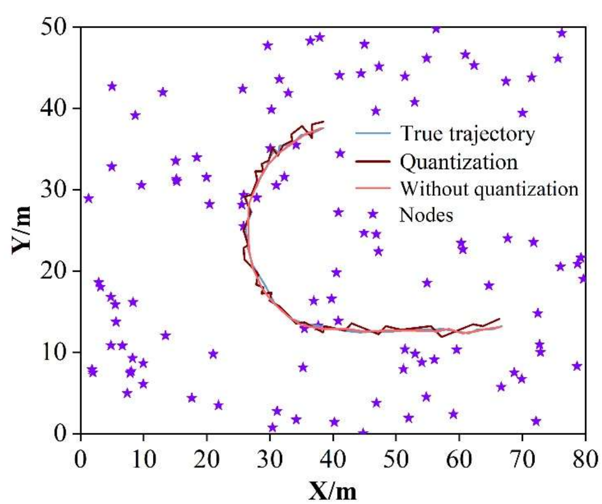


Fig. 9. Target tracking trajectories with quantization and without quantization

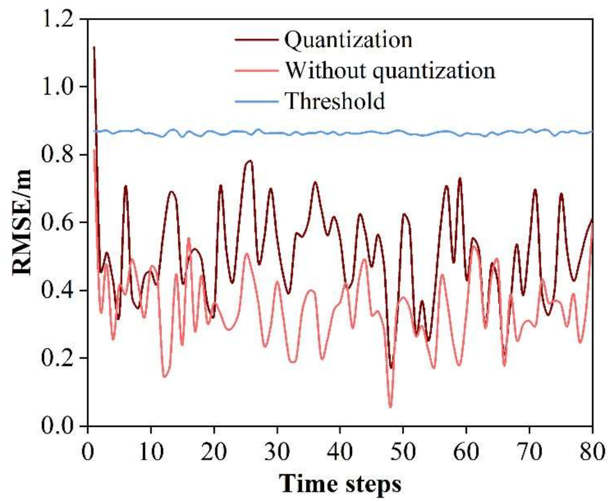


Fig. 10. Root mean square error with quantization and without quantization

The tracking sampling interval under the quantization constraint is shown in Fig. 11. In the initialization stage, the smallest sampling interval of 0.1 s is used.Since a small sampling interval results in a large tracking accuracy and high energy consumption, the sampling interval is adjusted to 0.45 s according to the tracking accuracy under the constraints.In addition, the average sampling intervals of the nonlinear part of the trajectory are lower than those of the linear part, which are 0.3126 s and 0.4135 s, respectively, since the tracking errors of the nonlinear part are larger than the tracking error of the linear part, thus the sampling interval needs to be reduced to meet the accuracy requirement of the tracking.

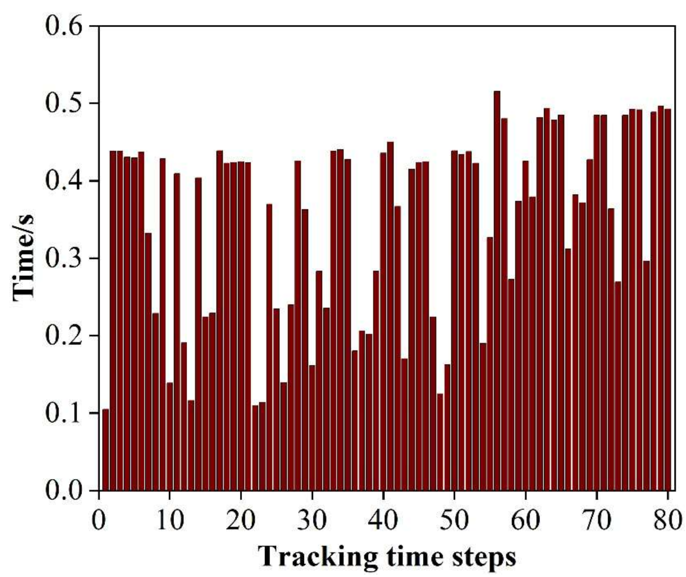


Fig. 11. Sampling intervals with quantization

The energy consumption of node accumulation under different conditions is shown in Fig. 12. $t = 0.1$ denotes the energy consumption of single node tracking without quantization when the fixed adoption interval duration is 0.1s. t=adaptive denotes the energy consumption of multi-node collaborative tracking with adaptive sampling interval. Due to the short sampling interval and no quantization mechanism, the energy consumption in single node tracking is much larger than that of multi-node collaborative tracking strategy. Under the adaptive sampling interval strategy, the observations are quantized and the communication between the wireless sensor networks is reduced, thus reducing the energy consumption of the network. And from Fig. 12, it can be seen that the quantized energy consumption is 47.56% less than the energy utilization efficiency of the energy consumption minus the complex without quantization, which achieves the goal of energy saving. Therefore, in the tracking process, the quantization technique can effectively reduce the network energy consumption, improve the energy utilization efficiency of the network, and realize the energy efficiency regulation of the distributed sensor network.

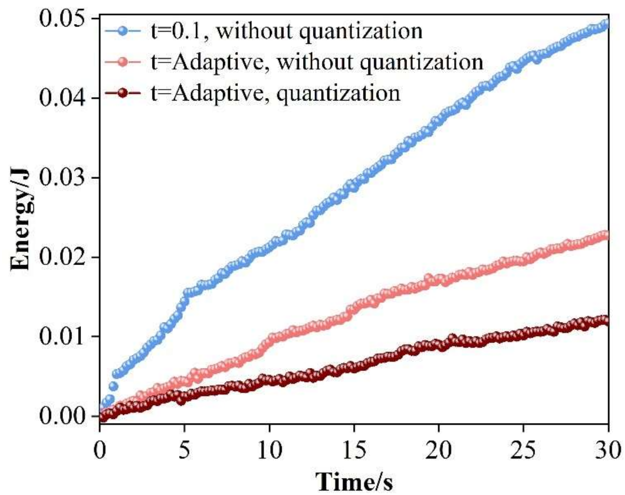


Fig. 12. Accumulated energy consumption under different conditions

5. Conclusion

In this paper, we construct a multicast model of sensor network based on sensor node graph location information, and design an energy efficiency regulation strategy for distributed sensor networks using QoS routing algorithm based on ant colony optimization.

Compared with the PCCAA algorithm, this paper's algorithm has a larger average node degree, smaller node degree variance, more balanced network link distribution, and better topology. And the percentage variance of channels in this paper's algorithm is smaller than that of the PCCAA algorithm, which indicates that the channel distribution in the network topology formed by it is more balanced, and the effectiveness of spectrum resources is stronger. In terms of the average power of the network, the average power of this paper's algorithm is slightly larger than that of the PCCAA algorithm, which increases the robustness of the network while reducing the energy consumption and BER to ensure the network performance. At the same time, the variance of node energy consumption of this paper's algorithm in different networks is smaller, and the energy consumption of each node in the network is more balanced, which indicates that this paper's algorithm can make the energy consumption of nodes in the whole network more balanced, and then improve the energy efficiency of the whole distributed sensor network.

In addition, this paper carries out target tracking simulation experiments on distributed sensor networks, taking into account the transmission amount of information and tracking sampling interval. The experimental results show that the algorithm in this paper reduces the communication between nodes through the quantization mechanism and extends the tracking sampling interval, which significantly reduces the energy consumption of the network and achieves the purpose of energy saving.

Funding

This work was supported by Key Projects of Scientific Research in Higher Educational Institutions in Anhui Province in 2022 (Natural Science): Development of logistics information collection system based on microcontroller (Item No. 2022AH052782); Anhui Province Construction Department Science Program Key Projects: Current Situation and Prospect Analysis of the Development of Construction Industrialization in Xuancheng City in 2023 (Item No. 2023-RK046); Faculty Quality "Curriculum Civics Teaching Team" in 2024 (Item No. 2024kcszjxtd01).

References

[1] Jin, X., Wang, G., Song, Y., & Sun, C. (2018). Smart building energy management based on network occupancy sensing. Journal of International Council on Electrical Engineering, 8(1), 30-36.

[2] Noye, S., North, R., & Fisk, D. (2016). Smart systems commissioning for energy efficient buildings. Building Services Engineering Research and Technology, 37(2), 194-204.

[3] Su, B., & Wang, S. (2020). An agent-based distributed real-time optimal control strategy for building HVAC systems for applications in the context of future IoT-based smart sensor networks. Applied Energy, 274, 115322.

[4] Dener, M., & Bostancıoğlu, C. (2015). Smart technologies with wireless sensor networks. Procedia-Social and Behavioral Sciences, 195, 1915-1921.

[5] Gupta, D., de Albuquerque, V. H. C., Khanna, A., & Mehta, P. L. (2021). Smart sensors for industrial internet of things. Springer International Publishing, 10, 978-3.

[6] Mahbub, M. (2020). A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. Internet of Things, 9, 100161.

[7] Ullo, S. L., & Sinha, G. R. (2020). Advances in smart environment monitoring systems using IoT and sensors. Sensors, 20(11), 3113.

[8] Morales-Velazquez, L., de Jesus Romero-Troncoso, R., Herrera-Ruiz, G., Morinigo-Sotelo, D., & Osornio-Rios, R. A. (2017). Smart sensor network for power quality monitoring in electrical installations. Measurement, 103, 133-142.

[9] Kazmi, A. H., O'grady, M. J., Delaney, D. T., Ruzzelli, A. G., & O'hare, G. M. (2014). A review of wireless-sensor-network-enabled building energy management systems. ACM Transactions on Sensor Networks (TOSN), 10(4), 1-43.

[10] Sofi, A., Regita, J. J., Rane, B., & Lau, H. H. (2022). Structural health monitoring using wireless smart sensor network–An overview. Mechanical Systems and Signal Processing, 163, 108113.

[11] Prabhu, B., Balakumar, N., & Antony, A. (2017). Wireless sensor network based smart environment applications. Wireless Sensor Network Based Smart Environment Applications (January 31, 2017). IJIRT, 3(8).

[12] Pradeep, K. G. M., Venkatesan, S., & Rajan, S. (2021). A neural network based smart building monitoring system using wireless sensor network. INFORMATION TECHNOLOGY IN INDUSTRY, 9(1), 232-239.

[13] Ali, A. I., & Zorlu Partal, S. (2022). Development and performance analysis of a ZigBee and LoRa-based smart building sensor network. Frontiers in Energy Research, 10, 933743.

[14] Dong, B., Prakash, V., Feng, F., & O'Neill, Z. (2019). A review of smart building sensing system for better indoor environment control. Energy and Buildings, 199, 29-46.

[15] Kumar, A., Singh, A., Kumar, A., Singh, M. K., Mahanta, P., & Mukhopadhyay, S. C. (2018). Sensing

technologies for monitoring intelligent buildings: A review. IEEE Sensors Journal, 18(12), 4847-4860.

[16] Zand, M., Nasab, M. A., Padmanaban, S., & Khoobani, M. (2022). Big data for SMART Sensor and intelligent Electronic devices–Building application. In Smart buildings digitalization (pp. 11-28). CRC Press.

[17] Haldorai, A., Ramu, A., Murugan, S., Haldorai, A., Ramu, A., & Murugan, S. (2019). Smart sensor networking and green technologies in urban areas. Computing and Communication Systems in Urban Development: A Detailed Perspective, 205-224.

[18] Naji, N., Abid, M. R., Benhaddou, D., & Krami, N. (2020). Context-aware wireless sensor networks for smart building energy management system. Information, 11(11), 530.

[19] Verma, A., Prakash, S., Srivastava, V., Kumar, A., & Mukhopadhyay, S. C. (2019). Sensing, controlling, and IoT infrastructure in smart building: A review. IEEE Sensors Journal, 19(20), 9036-9046.

[20] Wang, J., Yeh, W. C., Xiong, N. N., Wang, J., He, X., & Huang, C. L. (2019). Building an improved Internet of Things smart sensor network based on a three-phase methodology. IEEE Access, 7, 141728-141737.

[21] Zuchao Shen, Walter Leite, Huibin Zhang, Jia Quan & Huan Kuang. (2025). Using Ant Colony Optimization to Identify Optimal Sample Allocations in Cluster-Randomized Trials. The Journal of Experimental Education(1),167-185.

[22] Let G. Shine, Pratap C. Benin, Jagannath D. J., Dolly D. Raveena Judie & Evangeline L. Diana. (2023). Software-Defined Networking Routing Algorithms: Issues, QoS and Models. Wireless Personal Communications(3),1631-1661.

[23] Abdulaziz Elwalda, İsmail Erkan, Mushfiqur Rahman & Deniz Zeren. (2022). Understanding mobile users' information adoption behaviour: an extension of the information adoption model. Journal of Enterprise Information Management(6),1789-1811.