

Article

The Improvement of DV-Hop Model and Its Application in the Security Performance of Smart Campus

Aimin Yang^{1,2,3,4,5}, Qunwei Zhang^{1,2,3,4,5}, Yikai Liu^{1,2,3,4,5} and Ji Zhao^{6,*} 

¹ Hebei Engineering Research Center for the Intelligentization of Iron Ore Optimization and Ironmaking Raw Materials Preparation Processes, North China University of Science and Technology, Tangshan 063210, China; aimin@ncst.edu.cn (A.Y.); zhangqw@stu.ncst.edu.cn (Q.Z.); liuyk@stu.ncst.edu.cn (Y.L.)

² Hebei Key Laboratory of Data Science and Application, North China University of Science and Technology, Tangshan 063210, China

³ The Key Laboratory of Engineering Computing in Tangshan City, North China University of Science and Technology, Tangshan 063210, China

⁴ Tangshan Intelligent Industry and Image Processing Technology Innovation Center, North China University of Science and Technology, Tangshan 063210, China

⁵ College of Science, North China University of Science and Technology, Tangshan 063210, China

⁶ Modern Technology and Education Center, North China University of Science and Technology, Tangshan 063210, China

* Correspondence: zhaoji@ncst.edu.cn

Abstract: In the smart campus, sensors are the basic units in the whole the Internet of Things structure, which play the role of collecting information and transmitting it. How to transmits more information at a certain power level has attracted the attention of many researchers. In this paper, the DV-Hop algorithm is optimized by combining simulated annealing-interference particle swarm optimization algorithm to improve the node localization of wireless sensor networks and enhance the security performance of smart campus. To address the problem that particle swarm optimization easily falls into local optimum, a perturbation mechanism is introduced in the basic particle swarm optimization algorithm. Meanwhile, the acceptance probability P is introduced in the simulated annealing algorithm to determine whether a particle is accepted when it “flies” to a new position, which improves the probability of finding a global optimal solution. Comparing the average localization error and optimization rate of the DV-Hop algorithm, PSO-DV-Hop algorithm, and the optimized algorithm. The results show a greater advantage of the algorithm. This will greatly enhance the safety performance and efficiency of the smart campus.

Keywords: improved DV-Hop model; smart campus; wireless sensor network; partial least squares regression; perturbed particle swarm optimization; simulated annealing

MSC: 68W99



Citation: Yang, A.; Zhang, Q.; Liu, Y.; Zhao, J. The Improvement of DV-Hop Model and Its Application in the Security Performance of Smart Campus. *Mathematics* **2022**, *10*, 2663. <https://doi.org/10.3390/math10152663>

Academic Editor: Akemi Galvez Tomida

Received: 27 June 2022

Accepted: 27 July 2022

Published: 28 July 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Through technologies such as radio frequency identification, the smart campus realizes the connection between objects and the Internet [1,2]. In the structure of the smart campus, the wireless sensor network plays an indispensable role in collecting information and transmitting information, and is the most basic structural unit [3,4]. With the development of wireless sensors, they face security issues. The security of wireless sensor networks limits the rapid and efficient development of the smart campus [5–9].

A large-scale wireless sensor network for multiple base stations (BS) and a key management protocol are designed in [10,11]. The protocol employs an end-to-end data security approach to securely relay data between nodes and base stations or between two nodes. A distributed key revocation scheme is further employed to effectively remove the damaged node and then form a multi-BS key management protocol. Wireless sensor networks are

often deployed in remote or harsh environments and are vulnerable to various types of security attacks. In [12,13], it proposed a new energy-saving clustering mechanism based on the artificial bee colony algorithm to extend the network life cycle. The performance of the proposed method is compared to protocols based on LEACH and particle swarm optimization, which are studied in several router applications. The experimental results show that clustering on the basis of the artificial bee colony algorithm can be successfully applied to WSN routing protocol. In [14,15], it proposes that a possible solution to reduce security risks is to use directional antennas instead of omnidirectional antennas or in combination with omnidirectional antennas. Therefore, it surveyed the latest state-of-the-art methods in the field, which provides broad perspective, new challenges and open research issues for the future use of directional antennas to reduce security risks.

In order to improve the accuracy of positioning, In [16,17], it proposed an improved DV-Hop algorithm. The proposed algorithm is derived from the DV-Hop algorithm and uses the weight of the anchor to improve positioning accuracy without the need for additional hardware. Simulation results show that the improved DV-Hop algorithm can provide more accurate position estimation than the DV-Hop algorithm. The DV-Hop algorithm limits its applicability to small or medium-sized sensor networks. In [18,19], it proposes a new localization-based protocol and demonstrates how to use the Voronoi diagram effectively to extend DV-Hop algorithm. At the same time, the DV-Hop positioning error is maintained or decreased. In the localization scheme, nodes can also be localized through their Voronoi unit. To evaluate the performance of the protocol, a wide range of simulation experiments were provided using ns-2. The results show that our proposed algorithm is better than DV-Hop in regard to their performance.

In this paper, in order to solve the problem that the particle swarm algorithm can easily fall into a local optimum, a perturbation mechanism is introduced in the basic particle swarm optimization algorithm, thus, the particle swarm algorithm can “fly” to a new position when it falls into local optimum. In the meantime, the acceptance probability P in the simulated annealing algorithm is introduced to judge whether the particle is “accepted” to the new position. The improved particle swarm optimization algorithm has a greater probability to find the global optimal solution. The proposed new algorithm optimizes the DV-Hop algorithm, thereby reducing the average positioning error during node positioning and enhancing the security performance of the smart campus.

2. Wireless Sensor Network Security in the Smart Campus

2.1. Composition and Structure of the Smart Campus System

Internet of Things technology enables the interconnection of items to items, items to people, and people to people. Through sensing devices such as radio frequency identification, infrared sensors, global positioning systems and laser scanners, according to the network protocol, any object can be used for information exchange and communication realize intelligent identification, positioning, tracking, monitoring and management [20]. The Internet of Things is a new type of system in which virtual networks interact with the real world. The core and foundation of the Internet of Things is still the Internet. The Internet of Things is an extended network based on the Internet. It is characterized by ubiquitous data perception, wireless-based information transmission, intelligent information processing, and the client can be extended to any item [21].

Among the architectures of the smart campus, one of the most representative smart campus architectures is the EPCGlobal “Internet of Things” architecture, which is composed of EPC encoding, EPC tags and readers, ONS servers, and EPCIS servers [10,22]. EPCGlobal “Internet of Things” can be divided into three levels, the bottom layer is the sensing layer for data sensing, the second layer is the network layer for data transmission, and the top layer is the content application layer. Figure 1 shows the architecture of the smart campus.

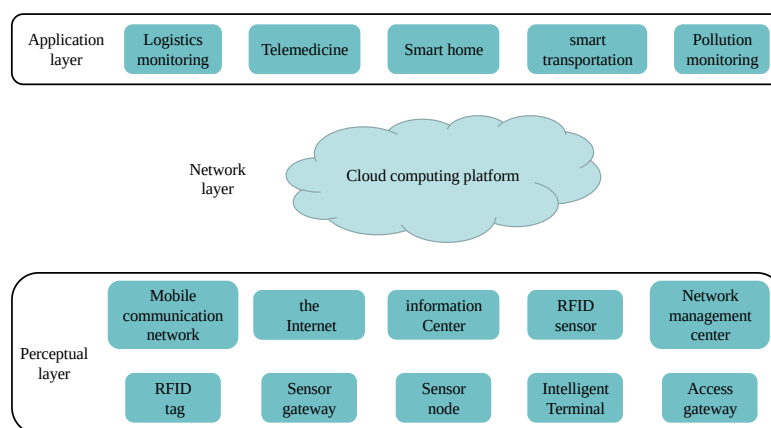


Figure 1. The Architecture diagram of smart campus.

The sensor is the “sensory organ” of the Internet of Things. It can sense the signals of heat, force, light, electricity, sound, and displacement, and provide the most original information for the network system’s processing, transmission, analysis, and feedback. As it can be seen from the above figure, the wireless sensor is the basic unit in the smart campus network. Meanwhile, the security of wireless sensor networks is the most fundamental security. Only by ensuring the security of the wireless sensor network, can the subsequent rapid transmission of information, information security encryption, and privacy protection be basically guaranteed.

2.2. Wireless Sensor Network Security Analysis

With the continuous development of technology, traditional sensors are gradually becoming miniaturized, intelligent, information and networking, and are experiencing a process of continuous enrichment from traditional sensors to embedded Web sensors. In the process of development, there are many security issues in wireless sensor networks [23–25].

- **Routing security issues.** In wireless sensor networks, a large number of sensor nodes are densely distributed in one area. Messages may need to pass through several nodes to reach the destination. The sensor network has a dynamic and multi-hop structure, and each node should have a routing function. Since each node is a potential routing node, it is more vulnerable to attack and makes the network insecure.
- **Protocol security issues.** In terms of security, there are two main methods: key management and secure multicast. Typically, the pre-configured key scheme computes the session key from pre-stored secret information. Due to node storage and energy limitations, pre-configured key management schemes must take saving storage space into consideration and reduce communication costs; wireless sensor networks may be placed in non-secure conditions. In order to prevent the supplier from injecting fake information into the network, it is necessary to implement secure multicast based on source authentication in the wireless sensor network.
- **Privacy and security issues.** Sensor network data privacy protection technology requires privacy protection technologies to prevent private information from being leaked and tampered with. At the same time, sensor network data management-related technologies are needed to complete tasks such as data aggregation, data query, and access control. Performance also needs to be optimized to reduce energy consumption, time delay, and data loss rate.

3. DV-Hop Algorithm

Wireless sensor networks are composed of sensors with sensing, computing, and communication capabilities and are widely used to monitor and track complex environments. Node location technology is one of the key technologies [26,27]. By optimizing node

positioning in wireless sensor networks, the goal of propagating more information when power is limited can be achieved.

At present, the localization algorithm can be mainly divided into an algorithm based on ranging and an algorithm without ranging [28]. The ranging-based algorithm improves the accuracy of positioning while adding additional hardware devices, which increases the energy cost for positioning. Positioning algorithms without ranging do not need to have additional hardware support. Although the positioning accuracy is low, the accuracy achieved is acceptable for most applications of sensor networks. If the positioning errors are less than 40% of the wireless communication radius of the sensor node, they would have little effect on the routing performance and target tracking accuracy. Typical positioning algorithms without ranging include the centroid algorithm [29], convex programming localization algorithm [30], and DV-Hop algorithm [31]. The positioning process of the DV-Hop algorithm consists of three steps, the flow chart of which is shown in Figure 2. The meaning of the relevant symbols is shown in Table 1.

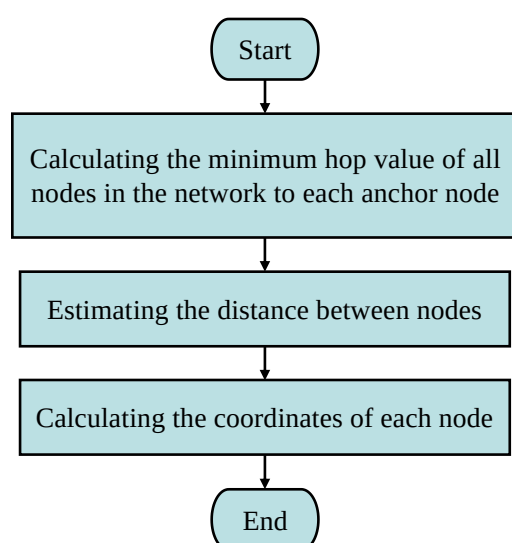


Figure 2. Flow chart of DV-Hop algorithm.

- Calculate the minimum hop value of all nodes in the network to each anchor node. The anchor node broadcasts a beacon to the neighbor node, including the coordinates and hop count of the anchor node, and the initial hop count is 0. After receiving the beacon, the neighbor node records the coordinate information of the anchor node, records and updates the minimum hop count to each anchor node, and then increments the hop count one by one to continue broadcasting to its neighbors. In this way, all nodes in the network record the minimum number of hops to each anchor node.
- Estimate the distance between nodes. Each anchor node calculates the average distance per hop according to the coordinate values of other anchor nodes recorded in the first step and the minimum hop count to these nodes, as shown in Equation (1). The anchor node calculates the average distance per hop and broadcasts the result to the network. The unknown node records only the first average distance per hop and forwards it to its neighbor node. The unknown node then multiplies the average hop distance value by the minimum hop number value to obtain the distance between the unknown node and the anchor node;

$$Hopsize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_j} \quad (1)$$

where $Hopsiz_e_i$ is the average bounce distance of the i -th anchor node, h is the hop value between the anchor node (x_i, y_i) and the anchor node (x_i, y_i) located.

- Calculate the coordinates of each node. Each unknown node calculates its own estimated coordinates according to the estimated distance from the unknown node to the anchor node obtained in the second stage. In general, the number of anchor nodes in the network would be more than three. In this paper, the partial least squares method is used to locate the nodes. Assuming that the position of n anchor nodes is $A_1(x_1, y_1), A_2(x_2, y_2), \dots, A_n(x_n, y_n)$, the position of the unknown node is (x, y) , and the estimated distance from the unknown node to the number of anchor nodes is $d_1, d_2, \dots, d_n$, respectively:

$$Ax = b \quad (2)$$

$$\text{where } X = \begin{bmatrix} x \\ y \end{bmatrix}, \quad A = -2 \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ \vdots & \vdots \\ (x_n - x_n) & (y_n - y_n) \end{bmatrix},$$

$$b = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}.$$

Partial least squares regression extracts components t_1 and u_1 in A and b . u_1 is $d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2$. t_1 is a linear combination of $(x_1 - x_n)$ and $(y_1 - y_n)$ in A . When extracting components, t_1 and u_1 should represent raw data as much as possible, and the correlation between t_1 and u_1 should be maximized.

Based on the above requirements, the steps of partial least squares regression are as follows [32]:

1. Vertex the original data A and b , and obtain the normalized independent variable matrix E_0 and the dependent variable matrix F_0 ;
2. Extract the first axes W_1 and c_1 and the corresponding first layer components t_1 and u_1 ;

$$\begin{aligned} t_1 &= E_0 w_1 \\ u_1 &= F_0 c_1 \end{aligned} \quad (3)$$

in the Equation (3), W_1 is a unitized eigenvector corresponding to the largest eigenvalue of the matrix $E_0' F_0 F_0' E_0$, and c_1 is a unitized eigenvector corresponding to the largest eigenvalue of the matrix $F_0' E_0 E_0' F_0$.

3. E_0 and F_0 respectively for the regression equation of t_1 ;

$$\begin{aligned} E_0 &= t_1 p_1' + E_1 \\ F_{0k} &= \alpha_{0k}^1 + \alpha_{1k}^1 t_1 + \dots + \alpha_{nk}^1 t_1^{n1} + F_{1k} \end{aligned} \quad (4)$$

where the regression coefficient vector is $p_1 = \frac{E_0' t_1}{\|t_1\|^2}$, and $p_1 = \frac{E_0' t_1}{\|t_1\|^2}$ is the k -th dependent variable. α_{ik}^1 represents the regression coefficient of the regression polynomial t_1^i term of the first component F_{0k} , n_{ik}^1 represents the number of times of the regression polynomial of the first component F_{0k} , and F_{0k} represents the residual after the F_{0k} regression.

$$F_1 = [F_{11}, F_{12}, \dots, F_{1n-1}] \quad (5)$$

E_1, F_1 are the residual matrices of two regression equations, respectively.

4. Test convergence. If the calculation accuracy requirement is not met, replace E_0 and F_0 with residual matrices E_1 and F_1 , then find the second axis and the second component,

and repeat the above steps until the requirements are met. If the calculation is terminated after the m -th component is calculated. That is:

$$\begin{aligned} E_0 &= \sum_{i=1}^m t_i p'_i \\ F_{0k} &= \sum_{i=0}^{n_k^1} \alpha_{ik}^1 t_1^i + \cdots + \sum_{j=0}^{n_k^m} \alpha_{ik}^m t_m^j \end{aligned} \quad (6)$$

5. Restore variables. Since $t_1, t_2, \dots, t_m$ can be expressed as a combination of $E_{01}, E_{02}, \dots, E_{0p}$, a regression equation form of $Y_k^* = F_{0k}$ for $x_j^* = E_{0j}$ can be obtained, namely:

$$y_k^* = \sum_{i=0}^{n_k^1} \alpha_{ik}^1 \left(\sum_{j=1}^p \beta_{ij} x_j^* \right)^i + \cdots + \sum_{i=0}^{n_k^m} \alpha_{ik}^m \left(\sum_{j=1}^p \beta_{ij} x_j^* \right)^i \quad (7)$$

in the Equation (7), β_{mj} represents the combination coefficient of x_j^* in the m -th component t_m after the linear combination of the normalized $x_1^*, x_2^*, \dots, x_p^*$. y^* and x^* were reduced according to a standardized reverse process.

In the partial least squares regression, since the extracted components are orthogonal to each other, the information carried by each component has no correlation, and the subsequently extracted information plays a complementary role in the previous information. The biggest advantage of partial least squares regression is that it can effectively overcome the multiple correlation between independent variables and the difficulty of observing sample points, and has the characteristics of a robust model and high precision. The position of the nodes in the wireless sensor network can be obtained by partial least squares regression, and the position information of each node is irrelevant.

Table 1. Symbol Description.

Symbol	Explanation
$Hops_{size_i}$	average by hop distance of the i -th anchor node
h	jump values between anchor nodes
d_i	estimated distance from the unknown node to each anchor node number
E_0	independent variable matrix
E_1	residual matrices
F_0	dependent variable matrix
F_1	residual matrices
W_1	eigenvectors
c_1	eigenvectors
p_1	regression coefficient vector
F_{0k}	the k -th dependent variable
α_{ik}^1	the regression coefficient of the regression polynomial t_1^i term of the first component F_{0k}
n_{ik}^1	the number of times of the regression polynomial of the first component F_{0k}
β_{mj}	the combination coefficient of x_j^*

4. Wireless Sensor Network Optimization Based on Simulated Annealing-Perturbed Particle Swarm Algorithm

Optimization of node positioning in wireless sensor networks is an NP-hard problem, and the current effective methods to solve such problems are heuristic intelligent algorithms. Particle swarm has the advantages of simple setup and high accuracy, but also has the advantage that it is easy to fall into local optimum. In order to solve the problem that the particle swarm algorithm is easy to fall into local optimum, a perturbation mechanism is introduced in the basic particle swarm algorithm. The particle swarm algorithm can “fly” to a new position when it falls into local optimum. At the same time, the acceptance probability P in the simulated annealing algorithm is introduced to determine whether a particle is accepted when it “flies” to a new position, based on which the particle swarm algorithm has a higher probability of finding the global optimal solution. The proposed new algorithm will optimize the DV-Hop algorithm to reduce the average localization error in the node localization process. In the following, it is proposed to start from the root cause of the particle swarm problem with a view to improving the particle swarm algorithm.

4.1. Perturbed Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm is derived from the study of predation behavior of birds. PSO is similar to a genetic algorithm and is an optimization tool for iterative classes. In the PSO algorithm, each particle has velocity and position, and the initial velocity and position are given randomly. The PSO algorithm is similar to the genetic algorithm and iteratively finds the optimal solution. It also evaluates the pros and cons of the solution through fitness, while finding the global optimal by following the current searched optimal value [33,34].

The PSO is initialized as a group of random particles (random solution), the i -th particle is represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, there is a d -dimension, the velocity of the particle i is $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, and then the optimal solution is found by iteration. In each iteration, the particle updates its position and velocity by tracking two “extreme values”. The first one is the optimal solution found for the particle itself. This solution is called the individual extremum P_{best} . The other extreme value is the optimal solution currently found for the entire population, which is the global extremum g_{best} . The iterative termination condition is generally selected as the maximum number of iterations.

When finding these two optimal values, particle i updates its d -dimensional velocity v_i and position x_i according to Equation (8)

$$\begin{aligned} v_{id}^{t+1} &= wv_{id}^t + c_1r_1(P_{best,id} - x_{id}^t) + c_2r_2(g_{best,d} - x_{id}^t) \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^t \end{aligned} \quad (8)$$

where $v_{id}(t)$ is the current speed of the d -th dimension of particle i . Usually, $|v_{id}| \leq v_{max}$. $x_{id}(t)$ is the current position $v_{id}(t+1)$ of the d -th dimension of the particle i , and $x_{id}(t+1)$ is the next moment value. $P_{best,d}$ and $g_{best,d}$ and as defined above, r_1, r_2 is a random number between $(0, 1)$, w is an inertia constant, and c_1, c_2 is an acceleration constant. Figure 3 is a flow chart of the steps of the particle swarm algorithm.

The particle swarm optimization algorithm could accomplish the optimization task well. The velocity update formula in Equation (8) consists of three parts: wv_i is the particle's original flight inertia, and c_1r_1 is the particle's best tendency to fly to itself. c_2r_2 gives the particles a tendency to fly toward the current global optimum. Here, the values of w, C_1 and C_2 and V_{max} have a great influence on the running results and efficiency of the algorithm.

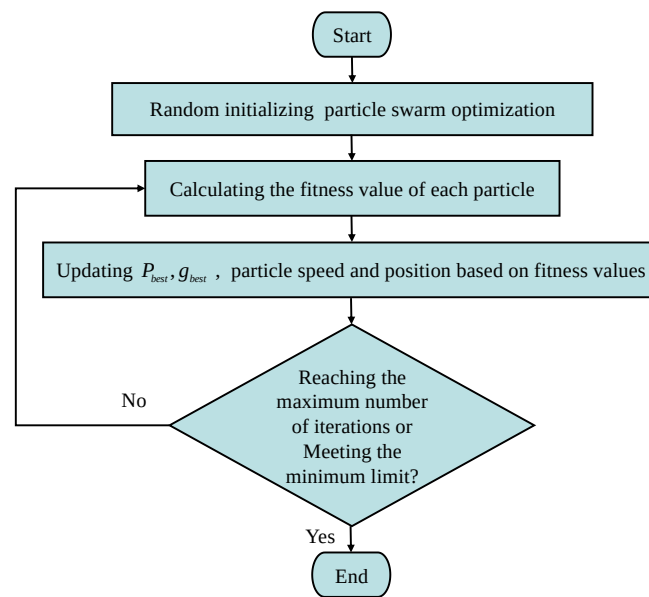


Figure 3. Flowchart of PSO algorithm.

4.2. Particle Swarm Optimization Algorithm Introducing Disturbance Mechanism

The convergence speed of PSO is very fast in the early stage of optimization, but the convergence speed becomes slow in the later stage of optimization, which leads to low convergence precision. This is mainly because the particle swarm is difficult to get rid of the local extremum. Many scholars have proposed a number of improved methods, such as adaptive PSO [35], chaotic PSO [36], and other optimization strategies. In this paper, the reason why PSO converges to local extremum is firstly analyzed, and then corresponding improvement strategies are proposed.

The experiment found that when the PSO is in a state of evolutionary stagnation, the particles in the particle group will appear as an “aggregate” phenomenon, and the particles will “scatter” when they break through the evolutionary stagnation. The following explains this phenomenon theoretically. Ref. [37] obtains the Equation (9) through strict mathematical derivation:

$$\lim_{t \rightarrow +\infty} x(t) = p^* = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2} p_{best} + \frac{c_2 r_1}{c_1 r_1 + c_2 r_2} g_{best} \quad (9)$$

Since r_1, r_2 are uniformly distributed, the average behaviour of the PSO system can be observed by its expected value, so Equation (9) can be rewritten as Equation (10):

$$\begin{aligned} \lim_{t \rightarrow +\infty} x(t) &= p^* \frac{c_1 p_{best} + c_2 g_{best}}{c_1 + c_2} \\ &= (1 - \alpha) p_{best} + \alpha g_{best} \end{aligned} \quad (10)$$

from Equation (10), with a basic PSO, the particles will accumulate above the extremum determined by their extremum and the global extremum of the population, if all particles do not find a position better than the global extremum in the approaching process. Then, the evolutionary process will be in a state of stagnation, and the particles will gradually gather.

This article adjusts $P_{best,d}$ and $g_{best,d}$ fly all particles to the new position p'^* . At this point, the particles will experience new search paths and domains, so the probability of finding a better solution is greater. In this paper, the evolutionary stagnation step is used as the trigger condition, and the individual extremum and the global extremum are simultaneously randomly perturbed. The extreme perturbation operator is:

$$p_{best} = r_3^{t_0 > T_0} p_{best}, g_{best} = r_4^{t_0 > T_0} g_{best} \quad (11)$$

where t_0 and t_g represent the individual extremum and global extremum evolution stagnation steps, respectively; T_0 and T_g represent the stagnation step thresholds for individual extremum and global extremum to be perturbed.

$$r_{3^{t_0 > T_0}} = \begin{cases} 1 & , t_0 \leq T_0 \\ X \sim E(0.5) & , t_0 > T_0 \end{cases}$$

$$r_{4^{t_0 > T_0}} = \begin{cases} 1 & , t_0 \leq T_0 \\ X \sim E(0.5) & , t_0 > T_0 \end{cases}$$

denotes an exponential random function with conditions. The exponential distribution has the characteristics of decreasing and no memory, and can be adjusted regularly with r_3 and r_4 . Therefore, Equation (8) can be rewritten as:

$$v_{id}^{t+1} = wv_{id}^t + c_1r_1(r_{3^{t_0 > T_0}}P_{best,d} - x_{id}^t) + c_2r_2(r_{4^{t_0 > T_0}}g_{best,d} - x_{id}^t) \quad (12)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t$$

We will simply refer to the particle swarm algorithm that introduces the perturbation mechanism as PPSO. It can be seen as the basic PSO algorithm is a special case of the PPSO algorithm. The particle swarm optimization algorithm that introduces the perturbation mechanism is expected to jump out of the local optimal value and find a better value.

4.3. Simulated Annealing-Perturbed Particle Swarm Optimization Algorithm

The simulated annealing algorithm is a stochastic optimization algorithm based on the Monte-Carlo iterative solution strategy. The starting point is based on the similarity between the annealing process of solid matter and the general optimization problem [38,39]. The principle of the simulated annealing algorithm is that the state changes with the temperature, and the state does not change as the temperature reaches the equilibrium state, and a local optimum solution may appear. It is necessary to anneal it at a certain annealing rate so that the local optimal solution can be jumped out. In the local optimal solution, probabilistic jumps out occur and converge to the global optimum. From the definition of $P = \exp(\Delta E/T)$ can be seen that high temperature can be accepted with the current state energy difference as a significant state, while at low temperatures, can only be accepted with the current state energy difference as the smaller new state is a significant state. When the temperature tends to 0, it is not accepting new states.

The simulated annealing algorithm includes the following steps: randomly selecting an initial model m_0 calculation within the range of parameter variation, and calculating a corresponding function value $E(m_0)$; perturbing the current model to generate a new model m , and calculating a new objective function value $\Delta E = E(m) - E(m_0)$; If $\Delta E \leq 0$, so the new model m is accepted, otherwise the new model m is accepted according to the probability $P = \exp(\Delta E/T)$, and T is the temperature. When the model is accepted, $m_0 = m, E(m_0) = E(m)$. We repeat a certain number of disturbances and acceptances at a temperature, slowly decrease the temperature and then repeat the above steps until the convergence meets the conditions. Figure 4 is a schematic diagram of the steps of a simulated annealing algorithm.

The simulated annealing algorithm is actually divided into two parts: a new model generated by random disturbance and calculating the change of the energy function and determining whether the new model is accepted. The increased model E may be accepted, so the minimum value can be discarded. Analysis of the simulated annealing algorithm shows that it mainly introduces the acceptance probability to judge whether the new model is accepted. As can be seen from Section 4.2, we introduced a perturbation mechanism to avoid the particle swarm algorithm from falling into local optimum. Here, we combine the acceptance probability in the simulated annealing algorithm. When the particle “fly” to another position, it is judged whether it is accepted by introducing the acceptance probability $P = \exp(\Delta E/T)$, in order to improve the proposed perturbed particle group.

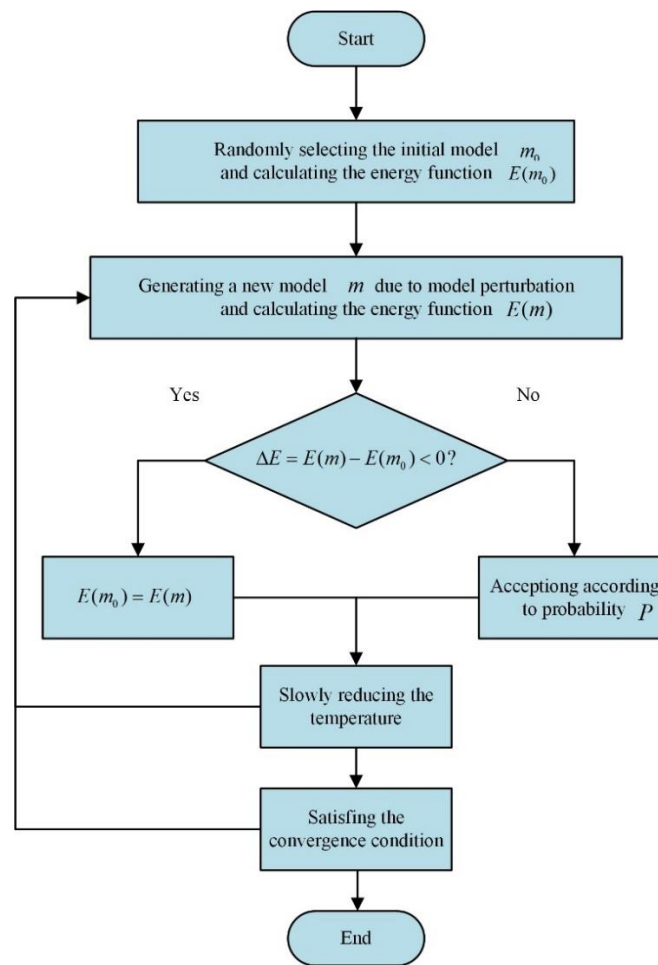


Figure 4. Flow chart of the simulated annealing algorithm.

The steps of the proposed simulated annealing-disturbed particle swarm algorithm are as follows:

1. randomly initializing the position of the particles in the particle group;
2. setting the initial temperatures $t_0 = f_{\max} - f_{\min}$, $f_{\max}$, $f_{\min}$ are the maximum and minimum fitness of the initial population;
3. setting the particle P_{best} to the current position, and the particle g_{best} setting the position of the optimal particle in the particle group;
4. Determine whether the algorithm convergence criterion is satisfied. If the convergence condition is met, turn to (13), otherwise perform the next step;
5. updating the position and velocity of the particles according to Equation (12);
6. If the fitness of the particle is better than the fitness P_{best} , P_{best} is the current position, otherwise the next step will be performed;
7. If the fitness of the particle is better than the fitness g_{best} , g_{best} is the current position, otherwise the next step is performed;
8. obtaining a new position by introducing a disturbance mechanism and calculating the fitness difference $\Delta\delta$ of the two;
9. If $\Delta\delta < 0$, the particles can enter the new position; otherwise, the next step is performed;
10. The particles are accepted according to the probability $P = \exp(\Delta E/T)$, and steps (6) and (7) are performed;
11. Temperature retreat operation: $t = 0.9 * t$;
12. Determine whether the algorithm satisfies the convergence condition. If the convergence condition is met, execute (13), otherwise turn to (5);
13. Output g_{best} , the algorithm ends.

By analyzing the perturbed particle swarm optimization algorithm that introduces the simulated annealing strategy, the introduced perturbation mechanism can make the particles “fly” to a new position. The acceptance probability in the simulated annealing strategy can be used to judge whether the new position is accepted so that the whole particle swarm algorithm avoids falling into local optimum and finds the global optimal solution. The proposed simulated annealing-disturbed particle swarm optimization algorithm is not a random “flying” to a new location, but a rational judgement.

5. Instance Verification and Result Analysis

The essence of the node location problem of wireless sensor networks is based on the optimization problem of different distance or path measurement values, which is an NP-hard problem [40]. At present, scholars believe that effective methods for solving NP-hard problems are heuristic intelligent algorithms, such as simulated annealing, genetic algorithms [33,41–45]. In this paper, the proposed simulated annealing-disturbed particle swarm optimization algorithm is used to solve the positioning optimization problem in wireless sensor networks.

It can be seen from the foregoing that we can get the estimated position $\hat{X}$ of the unknown node, and A^T is the position of the anchor node. Since the distance is estimated by partial least squares, the error always exists. The essence of the positioning problem of wireless sensor network nodes is to minimize the error. The error formula is as follows:

$$f_i(\hat{X}) = d_i - \left| \hat{X} - A_i^T \right| \quad (13)$$

For the convenience of calculation, we chose the circumscribed square of the communication range as the estimation of the communication range. Figure 5 is a schematic diagram of the communication range.

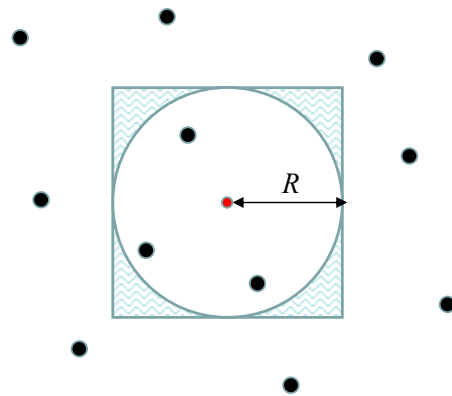


Figure 5. Schematic diagram of the communication range.

The proposed simulated annealing-disturbed particle swarm algorithm is simulated on the platform of MATLAB 2016. The particle updates the speed and position according to Equation (12), and the acceptance probability $P = \exp(\Delta E/T)$ determines whether the new position of “flying” is determined. In order to verify the feasibility of the proposed algorithm, we also tested the DV-Hop algorithm and the particle swarm optimization DV-Hop algorithm. In the simulation experiment, we set the communication radius of the node to $R = 10$, and deployed the unknown node in the square area of 100×100 . We set the parameter c_1 to 1.5, c_2 to 1.6, w to 0.4, evolutionary algebra to 150, and population size to 30. Equation (14) is a fitness function:

$$fitness(\hat{X}) = \sum_{i=1}^n \beta_i^2 f_i^2(\hat{X}) \quad (14)$$

among them, β_i^2 is proportional to the number of hops from the unknown node to the i -th anchor node, and the hop count can be obtained in the first step of the DV-Hop algorithm.

Equation (15) is a formula for evaluating the positioning error:

$$\text{AverageError} = \frac{\sum_{i=1}^n |\hat{X}_i - X_i|}{N * R} \times 100\% \quad (15)$$

among, N is the number of unknown nodes, X_i is the actual location of the unknown node, and R is the communication radius of the node.

In this paper, the simulation of the total number of fixed anchor nodes and fixed nodes is carried out, and the random distribution and uniform distribution of anchor nodes are considered. In the process of simulation, in order to avoid the generality, the positioning errors of the three algorithms of the simulation test are taken from the same distribution, and the average value of the five simulation results is the final simulation result. The execution time of the DV-Hop algorithm, PSO-DV-Hop algorithm, and the improved algorithm proposed in this paper are 91.8 s, 75.3 s and 73.6 s, respectively. It can be seen that the improved algorithm in this paper takes less time.

Figure 6 is a comparison of the average positioning errors of the three algorithms under uniform distribution and random distribution. Figure 7 is a graph showing the relationship between the proportion of anchor nodes and the average positioning error when the number of nodes is 300. Figure 8 is a comparison of the optimization ratio of the particle swarm algorithm and the algorithm of this paper.

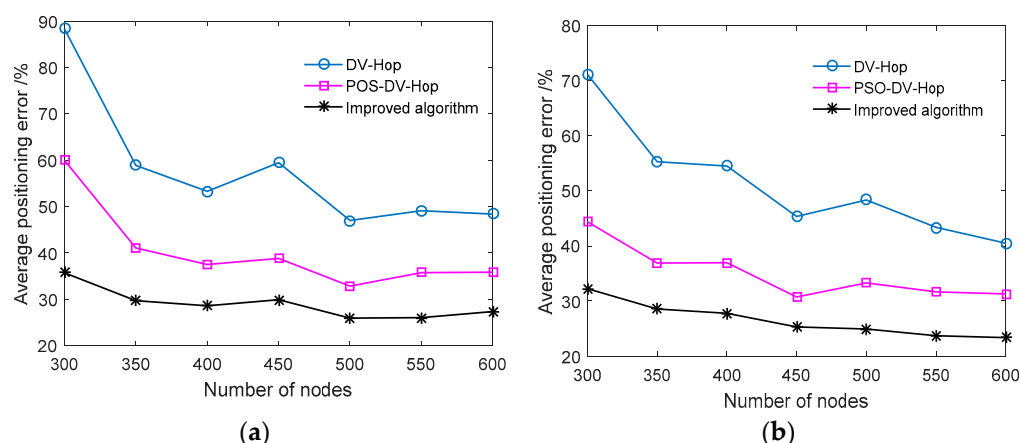


Figure 6. Comparison of average positioning errors of three algorithms for uniform distribution and random distribution: (a) average positioning error of uniform distribution of anchor nodes; (b) average positioning error under the random distribution of anchor nodes.

It can be seen from Figure 6 that the proposed algorithm has a lower average positioning error than the PSO-DV-Hop algorithm and the DV-Hop algorithm when the anchor nodes are uniformly distributed or randomly distributed. The advantage is more obvious than the other two algorithms. With the increase in the number of nodes, the average positioning error obtained by the algorithm proposed in this paper shows an overall decreasing trend. Compared with the PSO-DV-Hop algorithm and DV-Hop algorithm, the volatility is less and the average positioning error obtained is more stable. When the anchor nodes are uniformly distributed, the average positioning error of all three algorithms increases with the increase of the number of nodes when the number of nodes is between 400 and 450, especially the average positioning error obtained by the DV-HOP algorithm increases significantly. The average positioning error obtained by the PSO-DV-Hop algorithm and DV-Hop algorithm increases with the number of nodes when the anchor nodes are randomly distributed, between 450 and 500. However, the average positioning error of the proposed algorithm can show a stable decreasing trend with the increase in the number

of nodes. The algorithm proposed in this paper tends to be stable when the number of nodes is 500. When the number of nodes is fixed, the average positioning error of the three algorithms is smaller when the anchor nodes are randomly distributed compared to the average distribution. This also shows that the anchor nodes have better performance when they are randomly distributed.

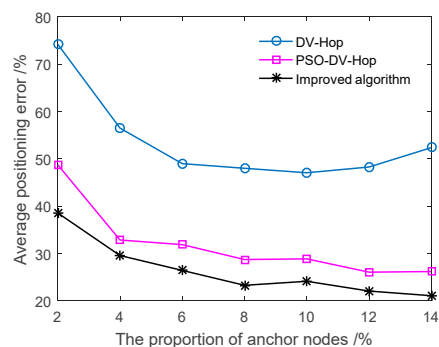


Figure 7. Relationship between the proportion of anchor nodes and the average positioning error.

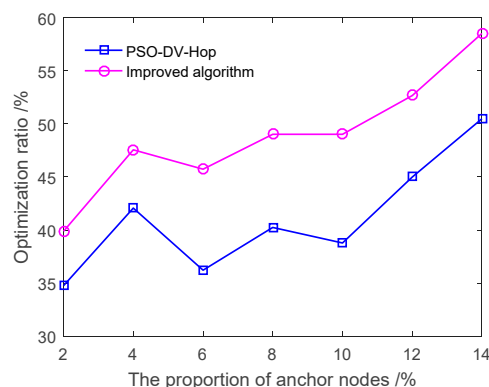


Figure 8. Comparison of the optimization ratio of the PSO algorithm and the algorithm of this paper.

It can be seen from Figure 7 that when the number of nodes reaches 300, the average positioning error is obtained by using the improved algorithm proposed in this paper and the PSO-DV-Hop algorithm has a gradual convergence trend as the proportion of anchor nodes increases. The DV-Hop algorithm has a tendency to decrease and then increase. The average localization error obtained by the three algorithms decreases significantly as the percentage of anchor nodes increases from 2% to 4%, with a reduction of more than 10% in all cases. At the same time, when the anchor node ratio is fixed, the average positioning error obtained by the algorithm in this paper is lower than that of the other two algorithms.

It can be seen from Figure 8 that with the increase in the proportion of anchor nodes, the optimization ratio obtained using the improved algorithm proposed in this paper shows an overall increasing trend. The optimization percentage obtained by both the PSO-DV-Hop algorithm and the proposed algorithm decreases when the percentage of anchor nodes is increased from 4% to 6% and 8% to 10%. However, the decrease of the optimization ratio obtained by the proposed algorithm is much smaller than that of the PSO-DV-Hop algorithm. It is obvious that the algorithm proposed in this paper is superior to the PSO-DV-Hop algorithm. The algorithm proposed in this paper is more effective.

When the anchor nodes are uniformly distributed and randomly distributed, the standard deviations of different algorithms are shown in Table 2.

Table 2. The standard deviations of different algorithms.

Situation \ Algorithm	DV-Hop Algorithm	PSO-DV-Hop Algorithm	Improved Algorithm
uniform distribution	173.67	75.06	10.50
random distribution	80.29	18.12	2.82

As can be seen, whether the anchor nodes are uniformly distributed or randomly distributed, the improved algorithm has a better standard deviation, which is enough to show that the improved algorithm has better reliability and accuracy compared to the other two algorithms.

6. Conclusions

In this paper, on the MATLAB 2016 platform, the improved algorithm proposed in this paper is simulated and tested in the communication range of 100×100 . At the same time, it is compared with the DV-Hop algorithm and PSO-DV-Hop algorithm. Simulation results show:

- When the anchor nodes are evenly distributed or randomly distributed, the proposed algorithm has a lower average positioning error than the PSO-DV-Hop algorithm and the DV-Hop algorithm when they have the same quantity of nodes. At the same time, with the increase in the number of nodes, the average positioning error has a tendency to gradually decrease, and the algorithm proposed in this paper tends to be stable when the number of nodes is 500.
- When the number of nodes is 300, with the increase in the proportion of anchor nodes, the average positioning error obtained by using the improved algorithm proposed in this paper has a gradual convergence trend, while the DV-Hop algorithm has a tendency to decrease and then increase.
- With the increase in the proportion of anchor nodes, the optimization ratio obtained using the improved algorithm proposed in this paper tends to increase gradually and is better than the PSO-DV-Hop algorithm.

The algorithm extracted in this paper can optimize the node location in the wireless sensor network without additional hardware equipment. When the storage capacity of the node is constant, the transmission information of the node could be improved, and the smart campus security problems could be solved.

Author Contributions: Methodology, A.Y., Q.Z., Y.L. and J.Z.; Writing—original draft, A.Y., Q.Z., Y.L. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [Tao Xue] grant number [ZJ2142Q] and the APC was funded by [ZJ2142Q].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data from UCI public dataset (<https://archive.ics.uci.edu/ml/index.php>, accessed on 27 June 2022).

Acknowledgments: Development of Intelligent Teaching Management Software Using Computer Vision Technology (ZJ2142Q).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, J.Q.; Yu, F.R.; Deng, G.; Luo, C.; Ming, Z.; Yan, Q. Industrial Internet: A Survey on the Enabling Technologies, Applications, and Challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1504–1526. [\[CrossRef\]](#)
2. Kumar, A.; Sharma, S.; Goyal, N.; Singh, A.; Cheng, X.; Singh, P. Secure and energy-efficient smart building architecture with emerging technology IoT. *Comput. Commun.* **2021**, *176*, 207–217. [\[CrossRef\]](#)

3. Ding, G.M.; Tan, Z.H.; Wu, J.S.; Zeng, J.; Zhang, L. Indoor fingerprinting localization and tracking system using particle swarm optimization and Kalman filter. *IEICE Trans. Commun.* **2015**, *E98B*, 502–514. [\[CrossRef\]](#)
4. Atzori, L.; Iera, A.; Morabito, G. SLoT: Giving a Social Structure to the Internet of Things. *IEEE Commun. Lett.* **2011**, *15*, 1193–1195. [\[CrossRef\]](#)
5. Sharma, S.; Sahu, A.; Verma, A.; Shukla, N. Wireless Sensor Network Security. In Proceedings of the International Conference on Computer Science and Information Technology, Kuala Lumpur, Malaysia, 12–14 June 2012; Springer: Berlin/Heidelberg, Germany, 2012.
6. Keerthika, M.; Shanmugapriya, D. Wireless Sensor Networks: Active and Passive attacks Vulnerabilities and Countermeasures. *Glob. Transit. Proc.* **2021**, *2*, 362–367. [\[CrossRef\]](#)
7. Tomić, I.; Mccann, J.A. A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols. *IEEE Internet Things J.* **2017**, *4*, 1910–1923. [\[CrossRef\]](#)
8. Tang, M.; Xin, Y.; Zhai, J. Efficient rate and power allocation in wireless sensor networks with big data constraint. *Optik* **2021**, *248*, 168138. [\[CrossRef\]](#)
9. Ma, A.; Mk, B. Information Security Measures in Homogeneous Wireless Sensor Networks. *Procedia Comput. Sci.* **2021**, *192*, 2699–2708.
10. Ferng, H.W.; Nurhakim, J.; Horng, S.J. Key management protocol with end-to-end data security and key revocation for a multi-BS wireless sensor network. *Wirel. Netw.* **2014**, *20*, 625–637. [\[CrossRef\]](#)
11. Seyhan, K.; Tu, N.N.; Akleyek, S.; Cengiz, K.; Islam, S.H. Bi-GISIS KE: Modified key exchange protocol with reusable keys for IoT security. *J. Inf. Secur. Appl.* **2021**, *58*, 102788. [\[CrossRef\]](#)
12. Karaboga, D.; Okdem, S.; Ozturk, C. Cluster based wireless sensor network routing using artificial bee colony algorithm. *Wirel. Netw.* **2012**, *18*, 847–860. [\[CrossRef\]](#)
13. Jawad, F.; Ozturk, C.; Wang, D.; Mahmood, M.; Al-Azzawi, O.; Al-Jemely, A. Sizing and layout optimization of truss structures with artificial bee colony algorithm. *Structures* **2021**, *30*, 546–559. [\[CrossRef\]](#)
14. Daniel-Ioan, C. Wireless Sensor Network Security Enhancement Using Directional Antennas: State of the Art and Research Challenges. *Sensors* **2016**, *16*, 488.
15. Yousefpoor, M.S.; Yousefpoor, E.; Barati, H.; Barati, A.; Movaghar, A.; Hosseinzadeh, M. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. *J. Netw. Comput. Appl.* **2021**, *190*, 103118. [\[CrossRef\]](#)
16. Li, J.; Zhang, J.; Liu, X. A Weighted DV-Hop Localization Scheme for Wireless Sensor Networks. In Proceedings of the 2009 International Conference on Scalable Computing and Communications; Eighth International Conference on Embedded Computing, Dalian, China, 25–27 September 2009; pp. 269–272.
17. Ouyang, A.; Lu, Y.; Liu, Y.; Wu, M.; Peng, X. An Improved Adaptive Genetic Algorithm based on DV-Hop for Locating Nodes in Wireless Sensor Networks. *Neurocomputing* **2021**, *458*, 500–510. [\[CrossRef\]](#)
18. Boukerche, A.; Oliveira, H.A.B.F.; Nakamura, E.F.; Loureiro, A.A. DV-Loc: A scalable localization protocol using Voronoi diagrams for wireless sensor networks. *IEEE Wirel. Commun.* **2009**, *16*, 50–55. [\[CrossRef\]](#)
19. Huang, Z.; Chen, D.Z.; Xu, J. Influence-based Voronoi diagrams of clusters. *Comput. Geom.* **2021**, *96*, 101746. [\[CrossRef\]](#)
20. Zhang, Y. Technology framework of the Internet of Things and its application. In Proceedings of the 2011 International Conference on Electrical and Control Engineering, Yichang, China, 16–18 September 2011; pp. 4109–4112.
21. Atzori, L.; Iera, A.; Morabito, G. Understanding the Internet of Things: Definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Netw.* **2017**, *56*, 122–140. [\[CrossRef\]](#)
22. Choi, E.Y.; Dong, H.L.; Lim, J.I. Anti-cloning protocol suitable to EPCglobal Class-1 Generation-2 RFID systems. *Comput. Stand. Interfaces* **2009**, *31*, 1124–1130. [\[CrossRef\]](#)
23. Shu, T.; Krunz, M.; Liu, S. Secure Data Collection in Wireless Sensor Networks Using Randomized Dispersive Routes. *IEEE Trans. Mob. Comput.* **2010**, *9*, 941–954.
24. Perrig, A. SPINS: Security protocols for sensor networks. *Wirel. Netw.* **2002**, *8*, 521–534. [\[CrossRef\]](#)
25. Rios, R.; Cuellar, J.; Lopez, J. Probabilistic receiver-location privacy protection in wireless sensor networks. *Inf. Sci.* **2015**, *321*, 205–223. [\[CrossRef\]](#)
26. Yan, W.; Shan, X.; Wei, J. The Mobile Nodes Location Technology in Wireless Sensor Network. *Chin. J. Sens. Actuators* **2011**, *24*, 1326–1330.
27. Nan, W.; Xue-Li, S. Research on WSN Nodes Location Technology in Coal Mine. In Proceedings of the 2009 International Forum on Computer Science-Technology and Applications, Chongqing, China, 25–27 December 2010; pp. 232–234.
28. Zhao, J.; Hua, L.I.; Zhao, X. The Comparative Research on the Location Technology of Wireless Sensor Networks. *Commun. Netw.* **2009**, *1*, 114–120. [\[CrossRef\]](#)
29. Deng, B.; Huang, G.; Zhang, L.; Liu, H. Improved centroid localization algorithm in WSNs. In Proceedings of the 2008 3rd International Conference on Intelligent System and Knowledge Engineering, Xiamen, China, 17–19 November 2008; pp. 1260–1264.
30. Doherty, L. Convex optimization methods for sensor node position estimation. In Proceedings of the IEEE Infocom, Anchorage, Alaska, 24–26 April 2001; pp. 1655–1663.
31. Venter, G.; Sobieszczanskisobieski, J. Particle Swarm Optimization. *Aiaa J.* **2002**, *41*, 129–132.
32. Geladi, P.; Kowalski, B.R. Partial least-squares regression: A tutorial. *Anal. Chim. Acta* **1986**, *185*, 1–17. [\[CrossRef\]](#)

33. Fu, H.; Li, Z.; Liu, Z.; Wang, Z. Research on Big Data Digging of Hot Topics about Recycled Water Use on Micro-Blog Based on Particle Swarm Optimization. *Sustainability* **2018**, *10*, 2488. [\[CrossRef\]](#)
34. Garcíagonzalo, E.; Fernándezmartínez, J.L. A Brief Historical Review of Particle Swarm Optimization (PSO). *J. Bioinform. Intell. Control.* **2012**, *1*, 3–16. [\[CrossRef\]](#)
35. Zhan, Z.H.; Zhang, J.; Li, Y.; Chung, H.S.H. Adaptive particle swarm optimization. *Prog. Nat. Sci. Mater. Int.* **2008**, *39*, 1362–1381.
36. Sun, Y.; Qi, G.; Wang, Z.; Van Wyk, B.Y.; Hamam, Y. Chaotic particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Washington, DC, USA, 6–9 July 1999; Volume 1, pp. 320–324.
37. Clerc, M.; Kennedy, J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [\[CrossRef\]](#)
38. Bertsimas, D.; Tsitsiklis, J. Simulated Annealing. *Stat. Sci.* **1993**, *8*, 10–15. [\[CrossRef\]](#)
39. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#)
40. Cai, L.; Chen, J. On fixed-parameter tractability and approximability of NP-hard optimization problems. In Proceedings of the 2nd Israel Symposium on Theory and Computing Systems, Natanya, Israel, 7–9 June 1993; pp. 118–126.
41. Khimasia, M.M.; Coveney, P.V. Protein Structure Prediction as a Hard Optimization Problem: The Genetic Algorithm Approach. *Mol. Simul.* **1997**, *19*, 205–226. [\[CrossRef\]](#)
42. Abdelhafiez, E.A.; Alturki, F.A. A new optimization algorithm for solving NP-hard problems. In Proceedings of the 2010 International Conference on Mechanical and Electrical Technology, Singapore, 10–12 September 2010; pp. 59–63.
43. Yang, A.M.; Yang, X.L.; Han, Y.; Guo, Y.K.; Liu, J.M.; Zhang, H.Q. Wireless channel optimization of Internet of things. *IEEE Access* **2018**, *6*, 54064–54074. [\[CrossRef\]](#)
44. Yang, A.M.; Li, Y.F.; Kong, F.B.; Wang, G.Q.; Chen, E.H. Security control redundancy allocation technology and security keys based on Internet of Things. *IEEE Access* **2018**, *6*, 50187–50196. [\[CrossRef\]](#)
45. Yang, A.M.; Zhang, C.Y.; Chen, Y.J.; Zhuansun, Y.X.; Liu, H.X. Security and Privacy of Smart Home Systems Based on the Internet of Things and Stereo Matching Algorithms. *IEEE Internet Things J.* **2020**, *7*, 2521–2530. [\[CrossRef\]](#)