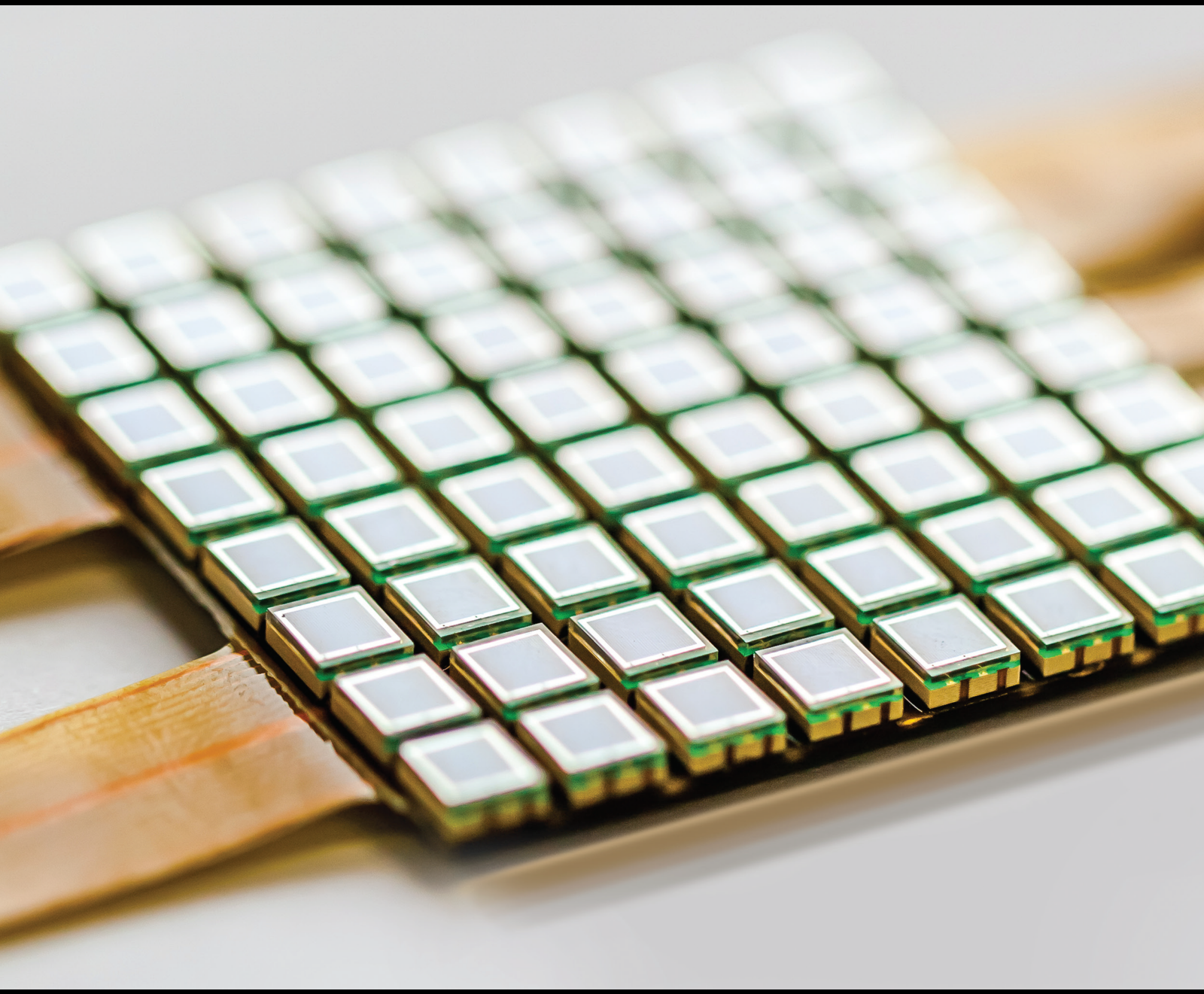
Journal of Sensors

Evaluation Techniques for Smart Sensor Networks

Lead Guest Editor: Wei-Chang Yeh
Guest Editors: Wei-Chiang Hong, Omprakash Kaiwartya, and Siddhartha Bhattacharyya



Evaluation Techniques for Smart Sensor Networks

Journal of Sensors

Evaluation Techniques for Smart Sensor Networks

Lead Guest Editor: Wei-Chang Yeh
Guest Editors: Wei-Chiang Hong, Omprakash Kaiwartya, and Siddhartha Bhattacharyya

Contents

Hindawi
Journal of Sensors
Volume 2021, Article ID 7833641, 12 pages
https://doi.org/10.1155/2021/7833641

Research Article

Solving Cold-Standby Reliability-Redundancy Allocation Problems with Particle-Based Simplified Swarm Optimization

Chia-Ling Huang,[1] Yunzhi Jiang ,[2] and Wei-Chang Yeh[3]

[1]*Department of International Logistics and Transportation Management, Kainan University, Taoyuan 33857, Taiwan*
[2]*School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*
[3]*Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan*

Correspondence should be addressed to Yunzhi Jiang; jiangyunzhi@foxmail.com

Received 22 April 2021; Revised 15 July 2021; Accepted 10 August 2021; Published 21 August 2021

Academic Editor: Wei-Chiang Hong

Particle swarm optimization (PSO) and simplified swarm optimization (SSO) are two of the state-of-the-art swarm intelligence technique that is widely utilized for optimization purposes. This paper describes a particle-based simplified swarm optimization (PSSO) procedure which combines the update mechanisms (UMs) of PSO and SSO to determine optimal system reliability for reliability-redundancy allocation problems (RRAPs) with cold-standby strategy while aimed at maximizing the system reliability. With comprehensive experimental test on the typical and famous four benchmarks of RRAP, PSSO is compared with other recently introduced algorithms in four different widely used systems, i.e., a series system, a series-parallel system, a complex (bridge) system, and an overspeed protection system for a gas turbine. Finally, the results of the experiments demonstrate that the PSSO can effectively solve the system of RRAP with cold-standby strategy and has good performance in the system reliability obtained although the best system reliability is not obtained in all four benchmarks.

1. Introduction

The reliability-redundancy allocation problem (RRAP) is the best known reliability design problem and is a classical optimization problem that seeks to maximize system reliability. To optimize system reliability for RRAP, the development of the system designs involves the selection of the reliability and the redundancy levels of the components. Hence, RRAP belongs to the category of mixed-integer programming problems because the components' reliabilities are denoted as continuous values that fall between zero and one, while the redundancy levels are integer values. RRAP formulations generally involve system constraints on allowable cost, weight, volume, etc.

Based on the system's required functions, the entire system is made up of a specific number of subsystems. The goal of the RRAP is to select the best combination of components and their reliabilities in each subsystem to maximize system reliability, R_s, given constraints such as cost, weight, and volume. In the RRAP literature, the objective is aimed at maximizing the system reliability subjected to several nonlinear constraints [1–4]. The mixed-integer nonlinear optimization programming model for RRAP is formulated as follows to maximize the system reliability by determining the number of components and the component reliabilities in each subsystem:

$$\begin{aligned} &\text{Maximize} && R_s = f(\mathbf{R}, \mathbf{N}), \\ &\text{Subject to} && g_j(\mathbf{R}, \mathbf{N}) \le u_j, \\ & && 1 \le j \le \text{the number of constraints}, \end{aligned} \tag{1}$$

where R_s is the system reliability; $\mathbf{R} = (r_1, r_2 \cdots, r_{\text{nsu}})$ and $\mathbf{N} = (n_1, n_2, \cdots, n_{\text{nsu}})$ are the component reliability vector and the redundancy allocation vector of the system, respectively, where r_i and n_i are, respectively, the reliability of

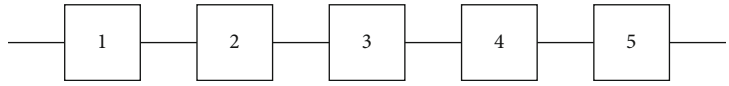


Figure 1: The series system.

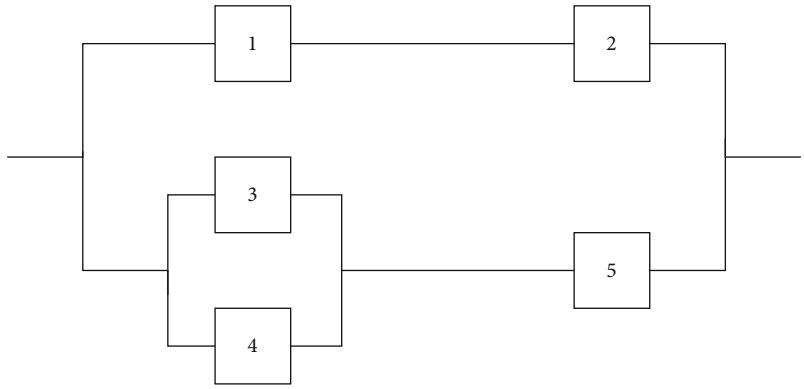


Figure 2: The series-parallel system.

each component and the number of components in subsystem i for $i = 1, 2, \cdots, n_{su}$; $f(\bullet)$ is the objective function for the system reliability; $g_j(\bullet)$ and u_j are the j^{th} constraint function and its resource limitation, respectively.

The main goal of reliability engineering is to increase the system reliability. Two different strategies, i.e., active and cold-standby, are usually used to meet this goal. All components simultaneously start to operate from time zero, for the active strategy, although only one is required at any particular time. The cold-standby strategy first developed and studied on redundancy allocation problem (RAP) by Coit in 2001 [5]; the redundant components are protected from stresses associated with system operation so that no component fails before its start.

There has been much research on different solution algorithms for RRAP. For example, particle swarm optimization (PSO) [3, 6], nondominated sorting genetic algorithm II (NSGA-II) [7–9], artificial bee colony algorithm (ABC) [10], genetic algorithms (GA) [1, 2, 4, 11], simplified swarm optimization (SSO) [12, 13], nest cuckoo optimization algorithm [14], a hybrid of PSO and SSO (PSSO) [15], and stochastic fractal search (SFS) [16] have been employed to study for RRAP.

Most previous research of RRAP in the literature has been devoted to the active strategy [1–4, 8, 10, 13, 15]. Several of these researches of RRAP using the cold-standby strategy can be distinguished which are aimed at solving the multiobjective [9] and focusing on the single objective of maximizing the system reliability [11, 12, 14, 16]. In addition, some researches of RRAP adopt the mixed strategy of active and cold-standby [6, 7]. In this work, the research of RRAP using the cold-standby strategy with single objective of system reliability that experimented on the four typical and famous benchmarks of RRAP including a series system (Figure 1), a series-parallel system (Figure 2), a complex (bridge) system (Figure 3), and an overspeed protection system for a gas turbine (Figure 4) as shown in Section 2.2 is studied. The research in [11, 12, 14, 16] studied RRAP using the cold-standby strategy with single objective of system reliability but only experimented on the first three famous benchmarks of RRAP including a series system, a series-parallel system, and a complex (bridge) system. Therefore, in this paper, the cold-standby strategy is used to increase system reliability in the RRAP formulation while focusing on the single objective of maximizing system reliability, and a solution methodology is presented to optimize system reliability for comprehensive experiments on all four famous benchmarks of RRAP.

Since the early 1990s, soft computing (SC) has been utilized to obtain optimal or good-quality solutions to difficult optimization problems. Swarm intelligence (SI) is a newly developed branch of SC that belongs in the category of population-based stochastic optimization. Particle swarm optimization (PSO) that was first developed by Kennedy and Eberhard in 1995 [17] and simplified swarm optimization (SSO) that was originally exploited by Yeh in 2009 [18] are two of the most well-known algorithms in SI. In recent years, we have seen an increasing interest both in PSO [3, 6, 15, 19–23] and in SSO [12, 13, 15, 24–28] for solving larger problems in science and technology.

The goal of this paper is to optimize the system reliability using RRAP with cold-standby strategy that belongs to the mixed-integer optimization programming model. Therefore, the merits of PSO and SSO, which are used to search for optima in real and discrete numbers, respectively, are adopted in this work. That is, a hybrid algorithm of PSO and SSO (PSSO) [15], which has only been used in RRAP with active strategy, is the first time used to optimize the system reliability using RRAP with cold-standby strategy. To demonstrate the efficiency of PSSO, a comprehensive comparative performance study with another recently introduced algorithm is presented for four different widely used systems. In summary, the novelty and contributions of this work are the RRAP using the cold-standby strategy with

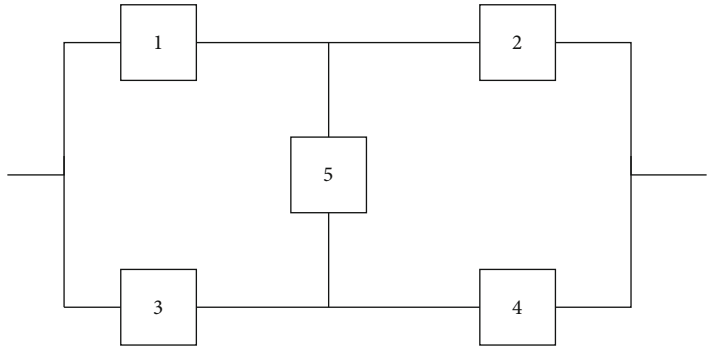


Figure 3: The complex (bridge) system.

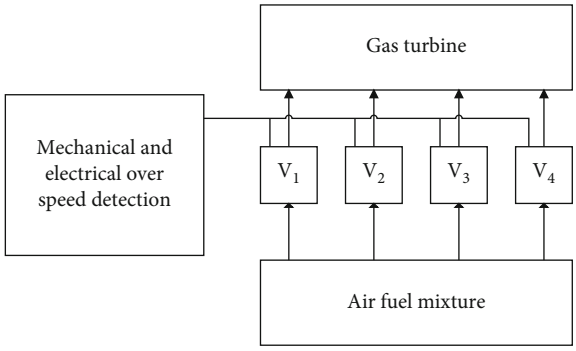


Figure 4: The overspeed protection system for a gas turbine.

single objective of system reliability that has comprehensive experiments on all the four famous benchmarks of RRAP.

This paper is organized as follows. Section 2 presents the mathematical formulation of the cold-standby redundancy strategy for RRAP and four systems. Section 3 provides respective descriptions of PSO and SSO and the orthogonal array test. The PSSO and related UM are discussed in Section 4. A comprehensive comparative study of the performances of PSSO optimizing the four systems is given in Section 5. Finally, the discussion and conclusion are given in Section 6.

2. The Cold-Standby Redundancy RRAP and Four Systems

2.1. The Cold-Standby Redundancy RRAP. Cold-standby redundancy is more difficult to implement than active redundancy because of the necessity to detect failures as they occur and activate the redundant component. If more than one component is used ($n_i > 1$), then there is one initially operating component and $n_i - 1$ components in cold standby waiting to be activated. The subsystem reliability for any distribution of component times-to-failure can be modeled as follows [5, 11].

$$R_i(t) = r_i(t) + \sum_{k=1}^{n_i-1} \int_0^t r_i(t-u) f_i^{(k)}(u)\,du. \quad (2)$$

A detection and switching mechanism is required to sense the occurrence of the component failure and to activate (switch to) a redundant component for cold-standby redundancy. However, the switch itself may fail. For the two imperfect operations, detection and switching, the subsystem reliability for any component time-to-failure distribution with imperfect failure detection and switching can be modeled as follows [5, 11].

Fact 1: continual detection and switching mechanism

$$R_i(t) = r_i(t) + \sum_{k=1}^{n_i-1} \int_0^t \rho_i(u)\, r_i(t-u) f_i^{(k)}(u)\,du. \quad (3)$$

Fact 2: detection and switching mechanism only at time of failure

$$R_i(t) = r_i(t) + \sum_{k=1}^{n_i-1} \int_0^t \rho_i^k\, r_i(t-u) f_i^{(k)}(u)\,du. \quad (4)$$

In this study, we investigate the continual detection and switching mechanism. It is difficult to determine a closed form of Equation (3). A convenient lower bound on subsystem reliability can be determined as follows because $\rho_i(u) \geq \rho_i(t)$ for all $u \leq t$.

$$R_i(t) \geq \tilde{R}_i(t) = r_i(t) + \rho_i(t) \sum_{k=1}^{n_i-1} \int_0^t r_i(t-u) f_i^{(k)}(u)\,du. \quad (5)$$

The limit of $R_i(t) - \tilde{R}_i(t)$ is zero as $\rho_i(t)$ approaches one. Hence, $\rho_i(t)$ is usually close to 1.0 [5].

If the probability distribution of a component's time-to-failure is exponential, then Equation (5) can be expressed by treating the probability of subsystem failure as a homogeneous Poisson process prior to the n_i^{th} failure. In this case, the reliability of the subsystem is the probability that there are strictly less than n_i failures, which is a Poisson distribution with parameter λ_i. Hence [5, 11],

$$\int_0^t r_i(t-u) f_i^{(k)}(u)\,du = \frac{e^{-\lambda_i t}(\lambda_i t)^k}{k!}. \quad (6)$$

A convenient lower bound on subsystem reliability is determined as follows:

$$\tilde{R}_i(t) = r_i(t) + \rho_i(t) \sum_{x=1}^{n_i-1} \frac{e^{-\lambda_i t}(\lambda_i t)^x}{x!}. \quad (7)$$

In the mathematical formulation of cold-standby redundancy for the RRAP, λ_i and n_i are the two decision variables and r_i is obtained on the basis of λ_i from Equation (6).

2.2. The Four Systems. This paper applies RRAP with cold-standby strategy to four systems: a series system (Figure 1), a series-parallel system (Figure 2), a complex (bridge) system (Figure 3), and an overspeed protection system for a gas turbine (Figure 4).

Due to their structures, the four systems have different objective functions to maximize their reliabilities but are subject to similar multiple nonlinear constraints. The respective RRAPs with cold-standby redundancy are formulated as follows.

System 1. The series system as in Figure 1 [11, 29]

$$\begin{aligned}
\max \quad & f(\mathbf{R},\mathbf{N}) = \prod_{i=1}^{N_{su}} R_i(n_i,t), \\
s.t \quad & g_1(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} w_i v_i^2 n_i^2 \le V, \\
& g_2(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} \alpha_i \left(\frac{-1000}{\ln r_i(t)}\right)^{\beta_i} \left(n_i + \exp\left(\frac{n_i}{4}\right)\right) \le C, \\
& g_3(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} w_i n_i \exp\left(\frac{n_i}{4}\right) \le W, \\
& 0 \le r_i(t) \le 1, r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer}, i = 1, \cdots, N_{su}.
\end{aligned} \quad (8)$$

System 2. The series-parallel system as in Figure 2 [11, 29]

$$\begin{aligned}
\max \quad & f(\mathbf{R},\mathbf{N}) = 1 - (1 - R_1(t)R_2(t)) \\
& \qquad \cdot \{1 - [1 - (1 - R_3(t))(1 - R_4(t))]R_5(t)\}, \\
s.t \quad & g_1(\mathbf{R},\mathbf{N}) \le V, \\
& g_2(\mathbf{R},\mathbf{N}) \le C, \\
& g_3(\mathbf{R},\mathbf{N}) \le W, \\
& 0 \le r_i(t) \le 1, r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer}, \ i = 1, \cdots, N_{su}.
\end{aligned} \quad (9)$$

System 3. The complex (bridge) system as in Figure 3 [11, 29]:

$$\begin{aligned}
\max \quad & f(\mathbf{R},\mathbf{N}) = R_1(t)R_2(t) + R_3(t)R_4(t) + R_1(t)R_4(t)R_5(t) \\
& \quad + R_2(t)R_3(t)R_5(t) - R_1(t)R_2(t)R_3(t)R_4(t) \\
& \quad - R_1(t)R_2(t)R_3(t)R_5(t) - R_1(t)R_2(t)R_4(t)R_5(t) \\
& \quad - R_1(t)R_3(t)R_4(t)R_5(t) - R_2(t)R_3(t)R_4(t)R_5(t) \\
& \quad + 2R_1(t)R_2(t)R_3(t)R_4(t)R_5(t), \\
s.t \quad & g_1(\mathbf{R},\mathbf{N}) \le V, \\
& g_2(\mathbf{R},\mathbf{N}) \le C, \\
& g_3(\mathbf{R},\mathbf{N}) \le W, \\
& 0 \le r_i(t) \le 1, r_i(t) \in \text{real number}, \\
& n_i \in \text{positive integer}, \ i = 1, \cdots, N_{su}.
\end{aligned} \quad (10)$$

System 4. The overspeed protection system [29]

An RRAP with a cold-standby redundancy formulation of an overspeed protection system with a time-related cost function [29] for a gas turbine is introduced for the first time. The model is formulated as follows.

$$\begin{aligned}
\max \quad & f(\mathbf{R},\mathbf{N}) = \prod_{i=1}^{N_{su}} [1\text{-}(1\text{-}r_i(t))^{n_i}], \\
s.t \quad & h_1(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} v_i n_i^2 \le V, \\
& h_2(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} \alpha_i \left(\frac{-1000}{\ln r_i(t)}\right)^{\beta_i} \left(n_i + \exp\left(\frac{n_i}{4}\right)\right) \le C, \\
& h_3(\mathbf{R},\mathbf{N}) = \sum_{i=1}^{N_{su}} w_i n_i \exp\left(\frac{n_i}{4}\right) \le W, \\
& 0.5 \le r_i(t) \le 1\text{-}10^{-6}, r_i(t) \in \text{real number}, \\
& 1 \le n_i \le 10, n_i \in \text{positive integer}, i = 1, \cdots, N_{su}.
\end{aligned} \quad (11)$$

3. Preliminaries

The PSO and SSO ought to be expounded at first because the PSSO is the hybrid of PSO and SSO. In addition, an orthogonal array test (OA) is introduced in this study to help improve solution quality and a penalty function is used to deal with constraints.

3.1. The PSO. PSO belongs to the family of swarm intelligence algorithms that was originally developed by Kennedy and Eberhard [17]. A population of random particles is initialized with random positions and velocities in the solution space; these are to be optimized by the fitness function to guide the direction of the solution. In each generation, pBest, denoted as P_i^{l-1}, is the local best solution among Y_i^0, Y_i^1, …,Y_i^{l-1}; each solution has its own pBest. P_{gBest}^{l-1} is the global

best, which is the best solution of all existing solutions among all pBests; there is only one gBest at a time. In the l^{th} generation, each solution Y_i^l moves towards pBest P_i^{l-1} and gBest P_{gBest}^{l-1} [19, 20] for $l = 0,1,2,\cdots,N_g$ and $i = 1, 2, \cdots, N_{\text{so}}$. The velocities and positions are updated according to the following equation after both P_i^{l-1} and P_{gBest}^{l-1} are found.

$$D_i^l = c_0 \cdot D_i^{l-1} + c_g \cdot \rho_1 \left(P_{\text{gBest}}^{l-1} - Y_i^{l-1}\right) + c_p \cdot \rho_2 \left(P_i^{l-1} - Y_i^{l-1}\right), \tag{12}$$

$$Y_i^l = Y_i^{l-1} + D_i^l, \tag{13}$$

where D_i^l and Y_i^l are the velocity and position of the i^{th} solution at the l^{th} generation, c_0 usually is equal 0.9999, $c_g\rho_1$ and $c_p\rho_2$ are the weights of the search directions, and 4 is the upper bound of $c_g + c_p$ [17].

3.2. The SSO. The SSO belongs to the swarm intelligence family and is a population-based dynamic optimization algorithm that was originally developed by Yeh [18]. It is also initialized with a population of random solutions inside the problem space and then searches for optimal solutions by updating subsequent generations. Let c_w, c_p, c_g, and c_r be the probabilities of the new variable value updated from the variable in the same position of the current solution; the sum of c_w, c_p, c_g, and c_r equals one. The fundamental concept of SSO is that to maintain population diversity and enhance the capacity to escape from a local optimum [18], each variable of any solution needs to be updated to a value related to its current value, its current pBest, the gBest, or a random feasible value. A random movement of SSO is based on the following model after c_w, c_p, and c_g are given:

$$x_{ij}^l = \begin{cases} x_{ij}^{l-1} & \text{if } \rho_{[0,1]} \in [0, C_w = c_w), \\ p_{ij}^{l-1} & \text{if } \rho_{[0,1]} \in \left[C_w, C_p = C_w + c_p\right), \\ g_j & \text{if } \rho_{[0,1]} \in \left[C_p, C_g = C_p + c_g\right), \\ x & \text{if } \rho_{[0,1]} \in \left[C_g, 1\right), \end{cases} \tag{14}$$

where $i = 1, 2, \cdots, \text{N}_{\text{so}}$, $j = 1, 2, \cdots, N_{\text{su}}$, $l = 0,1,2, \cdots, N_g$, and x is a random number between the lower and upper bounds of the j^{th} variable.

3.3. The Orthogonal Array Test (OA). This paper adopts the orthogonal array test (OA) to improve solutions because the OA is helpful to systematically and efficiently produce a potentially good approximation [30, 31]. Table 1 illustrates the class of the three-level OA where the numbers 1, 2, and 3 in each column indicate the levels of factors, and an equal number of 1s, 2s, and 3s is contained in each column. Columns 1-3 are the factors of the parameters c_w, c_p, and c_g in Equation (14), and column 4 is the factor of the parameter c_0 in Equation (12).

Table 1: The orthogonal array.

Tests	Columns			
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 2: The parameter values of SSO.

c_w	c_p	c_g	Reference
0.1	0.3	0.5	[39]
0.15	0.25	0.35	
0.1	0.3	0.5	[32]
0.1	0.3	0.3	
0.1	0.3	0.5	[34]
0.15	0.25	0.35	[35]
0.15	0.25	0.35	[30]
0.15	0.35	0.25	[33]
0.15	0.35	0.25	[36]
0.1	0.3	0.5	[37]
0.2	0.1	0.19	[38]

Table 3: The 3 most frequent parameter values of SSO.

c_w	Frequency	c_p	Frequency	c_g	Frequency
0.10	5	0.25	3	0.25	2
0.15	5	0.30	5	0.35	3
0.20	1	0.35	2	0.50	4

Table 4: The combinations of the parameter values.

	Factors			
	SSO			PSO
Tests	c_g	c_p	c_w	c_0
1	0.25	0.25	0.10	0.9999
2	0.25	0.30	0.15	Equation (15)
3	0.25	0.35	0.20	Equation (16)
4	0.35	0.25	0.15	Equation (16)
5	0.35	0.30	0.20	0.9999
6	0.35	0.35	0.10	Equation (15)
7	0.50	0.25	0.20	Equation (15)
8	0.50	0.30	0.10	Equation (16)
9	0.50	0.35	0.15	0.9999

The numbers of parameter values that were set according to the published papers of the SSO algorithm [30, 32–39] are arranged in Table 2. The top three parameter values that are most frequently set are $c_w = 0.1, 0.15, 0.2$, $c_p = 0.25, 0.3, 0.35$, and $c_g = 0.25, 0.35, 0.5$, as presented in Table 3.

The three parameter values that are most frequently set in PSO are 0.9999, linearly decreasing as in Equation (15) and exponentially decreasing as in Equation (16).

$$c_0 = 0.9999 - g * \frac{0.1}{N_g}, \tag{15}$$

$$c_0 = c_0 \cdot \sqrt[Ng]{\frac{.9}{c_0}}. \tag{16}$$

The nine combinations of the parameter values are presented in Table 4 from the information above.

3.4. The Penalty Function. A penalty function as shown in Equation (17) is used to deal with constraints. That is, the penalty function in Equation (17) is a penalty mechanism for system reliability if any constraint exceeds the upper limit of cost, weight, or volume.

$$R_{\text{penalty}} = \begin{cases} R_s & \text{if } X = (N, \Lambda) \text{is a feasible solution,} \\ R_s \left(\min \left\{ \left[\frac{V}{g_1(R,N)} \right], \left[\frac{C}{g_2(R,N)} \right], \left[\frac{W}{g_3(R,N)} \right], \right\} \right)^{\pi} & \text{otherwise,} \end{cases} \tag{17}$$

where R_{penalty} is the system reliability confirmed by the penalty function.

4. The PSSO

In this section, the PSSO is used together with an all-variable-UM (here termed n-UM and λ-UM) that retains the merits of PSO and SSO that are beneficial in searching for the optima in real and discrete numbers, respectively [15, 17–20, 32–39]. This study considers how the simultaneous application of those merits is conducive to computing the RRAP with cold-standby strategy, which is a mixed-integer programming model. Therefore, the key characteristics and merits of PSO and SSO are applied in this paper to optimize RRAP with cold-standby strategy.

The parameters λ_i of the Poisson distribution in Equation (7) and the numbers of all components need to be determined in cold-standby redundancy RRAP. Each number of components is an integer, and each parameter λ_i of a component is a real number. Hence, two different UMs, termed n-UM and λ-UM, are proposed to update the numbers of all components and the parameters λ_i of all components.

4.1. The n-UM. Applying the SSO, the proposed n-UM updates the number of components, i.e., n_{su} for $\text{su} = 1, 2, \cdots, N_{\text{su}}$, in each subsystem. Let $\text{so} = 1, 2, \cdots, N_{\text{so}}$, $\text{su} = 1, 2, \cdots, N_{\text{su}}$, $g = 1, 2, \cdots, N_g$, and n be a random number between the lower and upper bounds of the su$^{\text{th}}$ variable, and the mathematical model is as follows:

$$N_{\text{so,su}}^{g} = \begin{cases} \hat{n}_{\text{gBest},su} & \text{if } \rho_{[0,1]} \in [0, C_g = c_g), \\ \hat{n}_{\text{so,su}} & \text{if } \rho_{[0,1]} \in [C_g, C_p = C_g + c_p), \\ n_{\text{so,su}}^{g-1} & \text{if } \rho_{[0,1]} \in [C_p, C_w = C_p + c_w), \\ n & \text{if } \rho_{[0,1]} \in [C_w, 1]. \end{cases} \tag{18}$$

4.2. The λ-UM. Applying the velocity function of PSO, the proposed λ-UM updates the parameters λ_i of the Poisson distributions (Equation (7)) of the components in each subsystem. The mathematical model is as follows:

$$\Lambda_{\text{so}}^{g} = c_0 \cdot \Lambda_{\text{so}}^{g-1} + c_g \cdot \rho_1 \left(\hat{\Lambda}_{\text{gBest}}^{g-1} - \Lambda_{\text{so}}^{g-1} \right) + c_p \cdot \rho_1 \left(\hat{\Lambda}_{\text{so}}^{g-1} - \Lambda_{\text{so}}^{g-1} \right). \tag{19}$$

4.3. Pseudo-Code for PSSO. The pseudo-code of PSSO is as follows.

Step 0. Generate $X_{\text{so}}^0 = (N_{\text{so}}^0, \Lambda_{\text{so}}^0)$ randomly, calculate $F(X_{\text{so}}^0)$, and let $P_{\text{so}} = X_{\text{so}}^0$ and $F(P_{\text{so}}) = F(X_{\text{so}}^0)$ for $\text{so} = 1, 2, \cdots, N_{\text{so}}$.

Step 1. Find gBest such that $F(X_{\text{so}}^0) \le F(X_{\text{gBest}}^0)$ for $\text{so} = 1, 2, \cdots, N_{\text{so}}$.

Step 2. Let $g = 1$.

Step 3. Update N_{so}^g and Λ_{so}^g based on Equations (18) and (19).

Step 4. If $F(P_{\text{so}}) < F(X_{\text{so}}^g)$, let $P_{\text{so}} = X_{\text{so}}^g$ and $F(P_{\text{so}}) = F(X_{\text{so}}^g)$ and go to Step 5. Otherwise, go to Step 6.

Step 5. If $F(P_{\text{gBest}}) < F(P_{\text{so}})$, let gBest = so.

Step 6. If $\text{so} < N_{\text{so}}$, let so = so + 1 and go to Step 3.

Step 7. If $g < N_g$, let $g = g + 1$ and go to Step 2. Otherwise, halt.

Table 5: The parameter values used in Systems 1 and 3.

Subsystem i	$10^5\alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.330	1.5	1	7			
2	1.450	1.5	2	8			
3	0.541	1.5	3	8	110	175	200
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

Table 6: The parameter values used in System 2.

Subsystem i	$10^5\alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.500	1.5	2	3.5			
2	1.450	1.5	4	4.0			
3	0.541	1.5	5	4.0	180	175	100
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	4.5			

Table 7: The parameter values used in System 4.

Subsystem i	$10^5\alpha_i$	β_i	v_i	w_i	V	C	W
1	1	1.5	1	6			
2	2.3	1.5	2	6	250.0	400.0	500.0
3	0.3	1.5	3	8			
4	2.3	1.5	2	7			

Table 8: The reliability of System 1 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.96957732	0.96385617	0.91263728	0.00815955
2	0.96956482	0.96500015	0.95140225	0.00434131
3	0.99700404	0.92272200	0.82028328	0.04904667
4	0.99656129	0.91877856	0.76842776	0.05594605
5	0.96957924	0.95894940	0.87581632	0.01622886
6	0.96957612	0.96423395	0.93441124	0.00538507
7	0.96957233	0.94684495	0.81660930	0.02901255
8	0.99651026	0.92788871	0.81550324	0.05106236
9	0.96957926	0.88761690	0.68355049	0.07417057
Average	0.97861385	0.93954342	0.84207124	0.03259477

5. Experimental Results

While this paper aims at optimizing the system reliability, the studied RRAP with cold-standby strategy comprehensively applied to the typical and well-known four systems described in Section 2.2: a series system, a series-parallel system, a complex (bridge) system, and an overspeed protection system for a gas turbine is solved by PSSO [11, 29].

The PSSO implemented for RRAP with cold-standby strategy including the four systems was coded in the C^{++} programming language and run on an Intel Core i7 3.07 GHz PC with 6 GB memory. The experiments used

Table 9: The reliability of System 2 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99998828	0.99998567	0.99997026	0.00000451
2	0.99998827	0.99998617	0.99997358	0.00000360
3	0.99998826	0.99998677	0.99997358	0.00000332
4	0.99998826	0.99998517	0.99996740	0.00000534
5	0.99998828	0.99997617	0.99957038	0.00004333
6	0.99998827	0.99998245	0.99994913	0.00000724
7	0.99998827	0.99995877	0.99971734	0.00004601
8	0.99998826	0.99997295	0.99981548	0.00002908
9	0.99998828	0.99944028	0.99314011	0.00138911
Average	0.99998827	0.999919378	0.999119696	0.000170171

Table 10: The reliability of System 3 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99997538	0.99997142	0.99990226	0.00000963
2	0.99997535	0.99997288	0.99993734	0.00000603
3	0.99997532	0.99997305	0.99990998	0.00000703
4	0.99997532	0.99997176	0.99992335	0.00000656
5	0.99997538	0.99995309	0.99930499	0.00006959
6	0.99997536	0.99996981	0.99991011	0.00001123
7	0.99997538	0.99986639	0.99850974	0.00020563
8	0.99997535	0.99995631	0.99979033	0.00003090
9	0.99997538	0.99936835	0.98981897	0.00141941
Average	0.999975358	0.999889229	0.998556341	0.000196223

Table 11: The reliability of System 4 for the nine parameter combinations.

OA	Maximum	Mean	Minimum	Standard deviation
1	0.99679154	0.99678469	0.99674692	0.00001589
2	0.99679122	0.99678877	0.99676663	0.00000382
3	0.99679135	0.99678779	0.99678178	0.00000189
4	0.99679140	0.99678700	0.99674419	0.00000507
5	0.99679154	0.99677863	0.99674708	0.00002024
6	0.99679144	0.99678831	0.99674626	0.00000772
7	0.99679149	0.99677851	0.99669826	0.00002050
8	0.99679121	0.99678736	0.99674322	0.00000985
9	0.99679154	0.99320681	0.90823357	0.01352766
Average	0.99679141	0.99638754	0.98691199	0.00151251

1000 generations ($N_g = 1000$), the number of solutions was $N_{so} = 100$, the mission time $t = 1000$, and the convenient lower bound on subsystem reliability $\rho_i(t) = 0.99$.

Four systems are provided to evaluate the performance of PSSO for cold-standby RRAP, which is the mixed-integer nonlinear reliability design. The corresponding input

Table 12: Comparison of PSSO with previous work for System 1.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.96268903	0.95989227	0.96957758	0.96957924	0.99999	0.969579	0.99700404
N	(2,3,2,3,3)	(3,2,2,4,2)	(3, 2, 2, 3, 3)	(3, 2, 2, 3, 3)	(1, 3, 3, 2, 2)	(3, 2, 2, 3, 3)	(3,4,4,3,3)
λ_1	0.00015369	0.00030518	0.00026841	0.00026587	0.000596	0.000266	0.00003271
λ_2	0.00036900	0.00015259	0.00011931	0.00011924	0.000661	0.000119	0.00008385
λ_3	0.00006705	0.00009572	0.00008840	0.00008856	0.000546	0.000089	0.00002787
λ_4	0.00036014	0.00073242	0.00036600	0.00036728	0.000624	0.000367	0.00006836
λ_5	0.00026303	0.00009942	0.00025356	0.00025387	0.000559	0.000254	0.00008259
r_1	0.85754177	0.73699381	0.76459335	0.76653	0.742386	0.766498	0.96782041
r_2	0.69142591	0.85848344	0.88752892	0.88759	0.816868	0.887576	0.91957190
r_3	0.93514551	0.90871712	0.91539527	0.91525	0.713963	0.915293	0.97251610
r_4	0.69757716	0.48074328	0.69350544	0.69261	0.575015	0.692567	0.93392478
r_5	0.76871676	0.90536223	0.77603145	0.77579	0.738721	0.775825	0.92072882
MPI	91.97%	92.53%	90.15%	90.15%		90.15%	

Table 13: Comparison of PSSO with previous work for System 2.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.99998418	0.99996929	0.99998824	0.99998827	0.99999999	0.99998827	0.99998828
N	(3,3,1,2,3)	(3,3,1,2,3)	(3,3,2,1,3)	(3,3,1,2,3)	(3,2,1,3,3)	(3,3,1,2,3)	(3,3,1,2,3)
λ_1	0.00020286	0.00030518	0.00019255	0.00019156	0.00042564	0.00019186	0.00019217
λ_2	0.00018530	0.00027466	0.00017100	0.00016417	0.00017187	0.00016498	0.00016433
λ_3	0.00023161	0.00006104	0.00009632	0.00010691	0.00033426	0.00010705	0.00010766
λ_4	0.00007707	0.00005276	0.00010680	0.00009639	0.00033807	0.00009509	0.00009650
λ_5	0.00016042	0.00018311	0.00014449	0.00014839	0.00025940	0.00014847	0.00014768
r_1	0.81638904	0.73699381	0.82484672	0.82567	0.68245868	0.82542444	0.82516974
r_2	0.83085895	0.75983179	0.84281657	0.84860	0.86473794	0.84791146	0.84846299
r_3	0.79325626	0.94079016	0.90817308	0.89860	0.69972896	0.89848373	0.89793059
r_4	0.92582234	0.94860880	0.89869900	0.90811	0.73210357	0.90929419	0.90800596
r_5	0.85178644	0.83268033	0.86546301	0.86209	0.70098038	0.86202871	0.86270308
MPI	25.91%	61.83%	0.34%	0.09%		0.09%	

data and parameters are the same as in [11, 29] and are presented in the supplementary file "Data.docx" (available here) and Tables 5–7, respectively.

The experimental results solved by PSSO in terms of the statistical analysis for the maximum (the best), mean, minimum (the worst), and standard deviation of the related nine combinations, using the OA introduced in Section 3.3, are presented in Tables 8–11. The best solutions for the system reliability are 0.99700404, 0.99998828, 0.99997538, and 0.99679154 for the four systems, respectively.

Finally, Tables 12–14 illustrate the best performances of PSSO in comparison with previous results [11, 12, 14, 16], the PSO, and the SSO algorithms for the first three systems, respectively. The cold-standby strategy for RRAP is applied to the fourth system, i.e., overspeed protection of a gas turbine, for the first time, considering the PSSO applies the combined merits of PSO and SSO. Hence, the best performance of PSSO in comparison with the PSO and the SSO algorithms for the fourth system is illustrated in Table 15.

In Tables 12–15, the second row illustrates the solution to the system reliability. Other rows containing N, λ_i, and r_i indicate the number of components, the parameters of the Poisson distribution in Equation (7), and the reliability of components in each subsystem, respectively. Finally, the MPI rows give the improvements of the solutions found by the proposed solution over those of the best known previous solutions; the calculation equation is $(R_{s\_\mathrm{PSSO}} - R_{s\_\mathrm{other}})/(1 - R_{s\_\mathrm{other}})$, where R_{s_PSSO} indicates the system reliability obtained by PSSO and R_{s_other} indicates the system reliability obtained by a previous algorithm.

The results demonstrate that the PSSO performs better than the PSO and the SSO in terms of system reliability for the fourth system. The results of the first three systems in terms of system reliability obtained by ENCOA [14] are

Table 14: Comparison of PSSO with previous work for System 3.

	PSO	SSO	GA, 2014 [11]	New SSO, 2019 [12]	ENCOA, 2020 [14]	SFS, 2019 [16]	PSSO
R_s	0.99996602	0.99994009	0.99997413	0.99997537	0.99999995	0.99997538	0.99997538
N	(3,3,3,3,1)	(3,3,2,3,2)	(3,3,3,3,1)	(3,3,2,4,1)	(2,3,2,3,3)	(3,3,2,4,1)	(3,3,2,4,1)
λ_1	0.00026983	0.00033569	0.00021744	0.00019913	0.00048053	0.00019954	0.00020096
λ_2	0.00018939	0.00009709	0.00015412	0.00015582	0.00035308	0.00015658	0.00015568
λ_3	0.00016766	0.00006783	0.00014232	0.00006808	0.00058080	0.00006839	0.00006777
λ_4	0.00030117	0.00054932	0.00031802	0.00049070	0.00052360	0.00048775	0.00048910
λ_5	0.00017940	0.00045776	0.00026897	0.00027934	0.00028347	0.00027585	0.00027874
r_1	0.76351099	0.71484227	0.80457234	0.81944	0.70647573	0.81911028	0.81794334
r_2	0.82746317	0.90747281	0.85717305	0.85571	0.68327513	0.85506338	0.85583287
r_3	0.84564286	0.93442369	0.86734683	0.93418	0.88812138	0.93389858	0.93447620
r_4	0.73994868	0.57734434	0.72759162	0.61219	0.61431138	0.61400415	0.61317957
r_5	0.83577161	0.63269698	0.76416666	0.75628	0.70242105	0.75892956	0.75673635
MPI	27.54%	58.90%	4.83%	0.04%			

Table 15: Comparison of PSSO with PSO and SSO for System 4.

	PSO	SSO	PSSO
R_s	0.99670304	0.99677964	0.99679154
N	(3,3,3,3)	(3,3,3,3)	(3,3,3,3)
λ_1	0.00007028	0.00006779	0.00007003
λ_2	0.00009144	0.00009527	0.00009313
λ_3	0.00005319	0.00004520	0.00004502
λ_4	0.00009473	0.00009376	0.00009315
r_1	0.93213157	0.93446076	0.93236397
r_2	0.91261420	0.90913042	0.91107606
r_3	0.94820411	0.95580962	0.95597900
r_4	0.90962070	0.91050480	0.91105944
MPI	2.68%	0.36%	

the best but the results found by PSSO are the second best. The detailed comparisons are as follows.

(1) The system reliabilities R_s of 0.99999, 0.99999999, and 0.99999995 obtained by ENCOA [14] for the first three systems are better than PSSO, those of the previous work, the PSO, and the SSO

(2) The system reliability R_s of 0.99679154 obtained by PSSO for the fourth system is better than the PSO and the SSO

(3) However, the system reliabilities R_s of 0.99700404, and 0.99998828 obtained by PSSO for the first two systems are better than those of the previous work, the PSO, and the SSO except ENCOA [14], and the system reliability R_s of 0.99997538 obtained by PSSO for the third system is better than those of the previous work, the PSO, and the SSO except ENCOA [14] and SFS [16].

6. Conclusion and Future Work

A particle-based version of SSO called PSSO with a new UM to enhance the ability of traditional SSO is used to solve RRAP with cold-standby strategy. The RRAP cold-standby effectively maximizes the system reliability with the PSSO. Moreover, the UM is an important part of soft computing. This paper presents significant and novel modifications to SSO to optimize the cold-standby redundancy RRAP.

A comprehensive comparative study of the performances of the PSSO and previous work has been made. The system reliability obtained by the PSSO is better than the PSO and SSO for the fourth system. The system reliability obtained by the PSSO is the second best; those are second to ENCOA [14] for the first three systems. Roughly speaking, the PSSO based on UM has the ability to optimize the mixed-integer programming model and can be used to solve cold-standby redundancy RRAP efficiently. In future research, we will focus on strengthening SSO performance and will apply it to different optimization problems and solve practical engineering problems with larger-scale systems.

Acronyms

PSO: Particle swarm optimization
SSO: Simplified swarm optimization
PSSO: Particle-based simplified swarm optimization
RAP: Redundancy allocation problem
RRAP: Reliability-redundancy allocation problem
pBest: Local best
gBest: Global best
n-UM: Proposed update mechanism for the number variables of all components
λ-UM: Proposed update mechanism for the λ variables of all components
MPI: Maximum possible improvement.

Notations

n_{su}, n_{so}, n_g: The number of subsystems in the system, solutions, and generations, respectively

N: $\mathbf{N} = (n_1, n_2, \cdots, n_{su})$ is the redundancy allocation vector of the system, where n_i is the number of components in subsystem i for $i = 1, 2, \cdots, N_{su}$

R: $\mathbf{R} = (r_1, r_2, \cdots, r_{Nsu})$ is the component reliability vector of the system, where r_i is the reliability of each component in subsystem i for $i = 1, 2, \cdots, N_{su}$

$R_i(\bullet), q_i$: $R_i(\bullet) = 1 - q_i^{n_i}$ is the reliability of subsystem i, where $q_i = 1 - r_i$ is the failure probability of each component in subsystem i for $i = 1, 2, \cdots, N_{su}$

$\tilde{R}_i(\bullet)$: The convenient lower-bound reliability of subsystem i

$r_i(t)$: The reliability of each component in subsystem i at the mission time t

$f_i^{(k)}$: The pdf of subsystem i at the k^{th} failure arrival for $k = 0, 1, \cdots, n_{su} - 1$

$\rho_i(\bullet)$: The failure detection/switching reliability

λ_i: The parameter of the Poisson distribution in subsystem i

R_s: The system reliability

$g_j(\mathbf{R}, \mathbf{N})$: The j^{th} constraint function with respect to **R** and **N**

α_i, β_i: The physical feature of each component in subsystem i for $i = 1, 2, \cdots, N_{var}$

u_j: The resource limitation for the j^{th} constraint function

v_i, c_i, w_i: The volume, cost, and weight, respectively, of each component in subsystem i, $i = 1, 2, \cdots, N_{var}$

V, C, W: The upper limits on the volume, cost, and weight of the system, respectively

$f(\mathbf{R}, \mathbf{N})$: The fitness function with respect to **R** and **N**

N_{so}^g, $n_{so,su}^g$: $N_{so}^g = (n_{so,1}^g, n_{so,2}^g, \cdots, n_{so,Nsu}^g)$ is the redundancy allocation vector of the soth solution at the g^{th} generation, where $n_{so,su}^g$ is the suth variable for su = 1, 2, $\cdots$, N_{su}

Λ_{so}^g, $\lambda_{so,su}^g$: $\Lambda_{so}^g = (\lambda_{so,1}^g, \lambda_{so,2}^g, \cdots, \lambda_{so,Nsu}^g)$ is the λ vector of the component parameters of the soth solution at the g^{th} generation, where $\lambda_{so,su}^g$ is the suth variable for su = 1, 2, $\cdots$, N_{su}

X_{so}^g: $X_{so}^g = (N_{so}^g, \Lambda_{so}^g)$ is the soth solution at the g^{th} generation

$\bullet^{\wedge}$, $\bullet^{\wedge}_{gBest}$: The related pBest and gBest.

Data Availability

Data are available in the supplementary information file.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Conceptualization, software, investigation, resources, and writing—original draft preparation were contributed by all authors; methodology was contributed by Chia-Ling Huang; validation was contributed by Yunzhi Jiang; formal analysis was contributed by Yunzhi Jiang and Chia-Ling Huang; data curation was contributed by Chia-Ling Huang; supervision was contributed by Yunzhi Jiang and Chia-Ling Huang; project administration was contributed by Chia-Ling Huang; funding acquisition was contributed by Yunzhi Jiang. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China and Ministry of Science and Technology, Taiwan, R.O.C., under grant 61702118, MOST 105-2221-E-424-003, and MOST 109-2221-E-424-002.

Supplementary Materials

The corresponding input data and parameters are the same as in [11, 29] and are presented in the supplementary file "Data.docx" and Tables 5–7, respectively. *(Supplementary Materials)*

References

[1] L. Sahoo, "Reliability redundancy allocation problems under fuzziness using genetic algorithm and dual-connection numbers," in *Nature-Inspired Computing Paradigms in Systems*, pp. 111–123, Elsevier, 2021.

[2] H. A. Khorshidi, I. Gunawan, and M. Y. Ibrahim, "A value-driven approach for optimizing reliability-redundancy allocation problem in multi-state weighted k-out-of-n system," *Journal of Manufacturing Systems.*, vol. 40, no. 1, pp. 54–62, 2016.

[3] E. Zhang and Q. Chen, "Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization," *Reliability Engineering & System Safety.*, vol. 145, pp. 83–92, 2016.

[4] H. Kim and P. Kim, "Reliability-redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm," *Reliability Engineering & System Safety.*, vol. 159, pp. 153–160, 2017.

[5] D. W. Coit, "Cold-standby redundancy optimization for non-repairable systems," *IIE Transactions.*, vol. 33, no. 6, pp. 471–478, 2001.

[6] Z. Ouyang, Y. Liu, S. J. Ruan, and T. Jiang, "An improved particle swarm optimization algorithm for reliability-redundancy allocation problem with mixed redundancy strategy and heterogeneous components," *Reliability Engineering & System Safety.*, vol. 181, pp. 62–74, 2019.

[7] W. Wang, M. Lin, Y. Fu, X. Luo, and H. Chen, "Multi-objective optimization of reliability-redundancy allocation problem for multi-type production systems considering redundancy strategies," *Reliability Engineering & System Safety.*, vol. 193, p. 106681, 2020.

[8] P. K. Muhuri, Z. Ashraf, and Q. M. D. Lohani, "Multiobjective reliability redundancy allocation problem with interval type-2 fuzzy uncertainty," *IEEE Transactions on Fuzzy Systems.*, vol. 26, no. 3, pp. 1339–1355, 2018.

[9] M. Abouei Ardakan and M. T. Rezvan, "Multi-objective optimization of reliability-redundancy allocation problem with cold-standby strategy using NSGA-II," *Reliability Engineering & System Safety.*, vol. 172, pp. 225–238, 2018.

[10] W. C. Yeh and T. J. Hsieh, "Solving reliability redundancy allocation problems using an artificial bee colony algorithm," *Computers & Operations Research.*, vol. 38, no. 11, pp. 1465–1473, 2011.

[11] M. Abouei Ardakan and A. Zeinal Hamadani, "Reliability-redundancy allocation problem with cold-standby redundancy strategy," *Simulation Modelling Practice and Theory.*, vol. 42, pp. 107–118, 2014.

[12] W. C. Yeh, "Solving cold-standby reliability redundancy allocation problems using a new swarm intelligence algorithm," *Applied Soft Computing.*, vol. 83, p. 105582, 2019.

[13] W. C. Yeh, Y. Z. Su, X. Z. Gao, C. F. Hu, J. Wang, and C. L. Huang, "Simplified swarm optimization for bi-objection active reliability redundancy allocation problems," *Applied Soft Computing Journal.*, vol. 106, p. 107321, 2021.

[14] M. ArezkiMellal and E. Zio, "System reliability-redundancy optimization with cold-standby strategy by an enhanced nest cuckoo optimization algorithm," *Reliability Engineering & System Safety.*, vol. 201, p. 106973, 2020.

[15] C. L. Huang, "A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems," *Reliability Engineering & System Safety.*, vol. 142, pp. 221–230, 2015.

[16] M. N. Juybari, M. A. Ardakan, and H. Davari-Ardakani, "A penalty-guided fractal search algorithm for reliability–redundancy allocation problems with cold-standby strategy," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 233, no. 5, pp. 775–790, 2019.

[17] J. Kennedy and R. C. Eberhard, "Particle swarm optimization," in *proceedings of IEEE international conference on neural networks*, pp. 1942–1948, Publishing, Piscataway, NJ, USA, 1995.

[18] W. C. Yeh, "A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems," *Expert Systems with Applications.*, vol. 36, no. 5, pp. 9192–9200, 2009.

[19] G. R. You, Y. R. Shiue, W. C. Yeh, X. L. Chen, and C. M. Chen, "A weighted ensemble learning algorithm based on diversity using a novel particle swarm optimization approach," *Algorithms.*, vol. 13, no. 10, p. 255, 2020.

[20] L. Li, L. Chang, T. Gu, W. Sheng, and W. Wang, "On the norm of dominant difference for many-objective particle swarm optimization," *IEEE Transactions on Cybernetics.*, vol. 51, no. 4, pp. 2055–2067, 2021.

[21] X. F. Song, Y. Zhang, D. W. Gong, and X. Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[22] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation.*, vol. 25, no. 4, pp. 794–808, 2021.

[23] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Transactions on Cybernetics.*, vol. 51, no. 2, pp. 874–888, 2021.

[24] W. C. Yeh, "A new harmonic continuous simplified swarm optimization," *Applied Soft Computing.*, vol. 85, p. 105544, 2019.

[25] W. C. Yeh and J. S. Lin, "New parallel swarm algorithm for smart sensor systems redundancy allocation problems in the Internet of Things," *Journal of Supercomputing.*, vol. 74, no. 9, pp. 4358–4384, 2018.

[26] X. Zhang, W. C. Yeh, Y. Jiang, Y. Huang, Y. Xiao, and L. Li, "A case study of control and improved simplified swarm optimization for economic dispatch of a stand-alone modular microgrid," *Energies*, vol. 11, no. 4, p. 793, 2018.

[27] W. C. Yeh, "A novel boundary swarm optimization method for reliability redundancy allocation problems," *Reliability Engineering & System Safety.*, vol. 192, p. 106060, 2019.

[28] W. C. Yeh, C. L. Huang, P. Lin, Z. Chen, Y. Jiang, and B. Sun, "Simplex simplified swarm optimisation for the efficient optimisation of parameter identification for solar cell models," *IET Renewable Power Generation.*, vol. 12, no. 1, pp. 45–51, 2018.

[29] T. C. Chen, "IAs based approach for reliability redundancy allocation problems," *Applied Mathematics and Computation.*, vol. 182, no. 2, pp. 1556–1567, 2006.

[30] W. C. Yeh, "Novel swarm optimization for mining classification rules on thyroid gland data," *Information Sciences.*, vol. 197, pp. 65–76, 2012.

[31] S. J. Ho, S. Y. Ho, and L. S. Shu, "OSA: orthogonal simulated annealing algorithm and its application to designing mixed<tex>$rm H_2 /rm H_infty$</tex>optimal controllers," *Man and Cybernetics, Part A: Systems and Humans.*, vol. 34, no. 5, pp. 588–600, 2004.

[32] W. C. Yeh, "Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans.*, vol. 42, no. 1, pp. 250–261, 2012.

[33] W. C. Yeh, "New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series," *IEEE Transactions on Neural Networks and Learning Systems.*, vol. 24, no. 4, pp. 661–665, 2013.

[34] C. Bae, W. C. Yeh, N. Wahid, Y. Y. Chung, and Y. Liu, "A new simplified swarm optimization (SSO) using exchange local search scheme, International Journal of Innovative Computing," *Information and Control*, vol. 8, no. 6, pp. 4391–4406, 2012.

[35] W. C. Yeh, C. M. Lin, and S. C. Wei, "Disassembly sequencing problems with stochastic processing time using simplified swarm optimization, International Journal of Innovation," *Management and Technology.*, vol. 3, no. 3, pp. 226–231, 2012.

[36] W. C. Yeh, Y. M. Yeh, P. C. Chang, Y. C. Ke, and V. Chung, "Forecasting wind power in the Mai Liao Wind Farm based on the multi-layer perceptron artificial neural network model with improved simplified swarm optimization," *Electrical Power and Energy Systems.*, vol. 55, pp. 741–748, 2014.

[37] C. Bae, N. Wahid, Y. Y. Chung, W. C. Yeh, N. W. Bergmann, and Z. Chen, "Effective audio classification algorithm using swarm-based optimization, International Journal of Innovative

Computing," *Information and Control*, vol. 7, no. 1, pp. 1–10, 2011.

[38] W. C. Yeh, "Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components," *Knowledge-Based Systems.*, vol. 64, pp. 1–12, 2014.

[39] C. Bae, W. C. Yeh, Y. Y. Chung, and S. L. Liu, "Feature selection with intelligent dynamic swarm and rough set," *Expert Systems with Applications.*, vol. 37, no. 10, pp. 7026–7032, 2010.

Hindawi
Journal of Sensors
Volume 2021, Article ID 5051863, 13 pages
https://doi.org/10.1155/2021/5051863

Research Article

Adaptive Chaotic Ant Colony Optimization for Energy Optimization in Smart Sensor Networks

Wenxian Jia,[1] **Menghan Liu**,[1] **and Jie Zhou**[1,2]

[1]*College of Information Science and Technology, Shihezi University, Shihezi 832000, China*
[2]*Xinjiang Tianfu Information Technology Co., Ltd., China*

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 14 April 2021; Revised 8 June 2021; Accepted 18 June 2021; Published 5 July 2021

Academic Editor: Omprakash Kaiwartya

Smart sensor network has the characteristics of low cost, low power consumption, real time, strong adaptability, etc., and it has a wide range of application prospects in the agricultural field. However, the smart sensor node is limited by its own energy; it also faces many bottlenecks in agricultural applications. Therefore, balancing the energy consumption of nodes and extending the life of the network are important considerations in the design of efficient routing for smart sensor networks. Aiming at the problem of energy constraints, this paper proposes an intelligent sensor network clustering algorithm based on adaptive chaotic ant colony optimization (ACACO). ACACO introduces logical chaotic mapping to interfere with the pheromone on the initial path and uses the adaptive strategy to improve the transition probability formula. After selecting the best next hop node, the advancing ants are released to update the local pheromone, and the current pheromone content is adjusted by the chaos factor. When the ants determine the path, they release subsequent ants to update the global pheromone. The simulation results show that ACACO has obvious advantages over genetic algorithm (GA) and particle swarm optimization (PSO).

1. Introduction

Smart sensor networks use smart sensors to collect environmental information in the monitored area, which have changed the way of interaction between humans and nature, which greatly expanded human perception. And smart sensor networks are widely used, such as realizing the monitoring of forest factors such as temperature, humidity, and light in forest environmental protection. In terms of security, the monitoring of late arrivals is realized by deploying sensor nodes in public places [1], implementing campaign tactical reconnaissance against enemy targets in the military field to monitor the enemy's real-time dynamics, and, in terms of wildlife protection, the deployment of sensor nodes to monitor the activities of wild animal groups, and so on.

Many key areas of smart sensor networks are worth studying, such as energy management [2, 3], data privacy protection, node location monitoring, and network routing settings. Although the application prospect is very impressive, many problems are still exposed. For example, the cluster head node of the traditional network is arbitrarily selected, without considering factors such as transmission distance. In addition, smart sensor nodes are generally deployed in unmanned areas with harsh conditions, and the access of nodes to the network increases the difficulty of network maintenance. In addition, power replacement under centralized management is impractical and difficult to achieve. Therefore, the irreplaceability of node power makes the energy consumption problem particularly important compared to other key technologies of smart sensor networks. Without affecting performance, designing an effective energy consumption control strategy has become a core issue in smart sensor networks.

It is worth mentioning that the cluster head node of the traditional network is arbitrarily selected, without considering factors such as transmission distance, which will cause excessive power consumption of public nodes [4–6]. In addition, some large-scale and vulgar mobile smart sensor networks have complex structures and changeable topologies. At the same time, the communication distance between sensor nodes is also limited. Therefore, how to select cluster head nodes for clustering in smart sensor networks and

design an effective energy consumption control strategy while ensuring the completion of the detection task has become the core issue in smart sensor networks [7–9].

Taking energy optimization as the starting point, this paper studies and designs a clustering algorithm based on ACACO to reduce the energy consumption of smart sensor single-round communication as much as possible. To verify the effectiveness of the algorithm, we compare it with two heuristic artificial intelligence algorithms, which are GA [10, 11] and PSO [12]. The contributions of this research are as follows:

(1) We propose a new smart sensor clustering model with randomly distributed nodes. We also define the energy consumption formula for network transceivers and calculate the total energy consumption of the network

(2) Different from the traditional ant colony optimization (ACO) [13, 14] to solve the cluster optimization problem, this paper proposes a new intelligent ACACO. It uses chaotic mapping to perturb the pheromone update and then uses an adaptive path selection strategy. This improves the usability of the algorithm and avoids falling into local optima. What is more important, ACACO balances the energy consumption of nodes and prolongs the life of the network, thereby solving the problem of energy constraints, enabling effective use of resources and reducing industrial costs

The structure of the paper can be expressed as follows. In Section 2, the related work is discussed. Then, in Section 3, the sensor clustering model and the evaluation of the energy consumption of smart sensor networks are introduced. In Section 4, ACACO based on ant colony optimization is proposed for minimizing the energy consumption of smart sensor networks. Section 5 introduces and discusses the performance of the proposed model and algorithm through simulation experiments. Finally, the conclusion part is given in Section 6.

2. Related Work

Compared with other traditional networks, smart sensor networks not only meet the high-quality service requirements of high throughput or low transmission delay but also pay attention to energy utilization and extend the lifespan of the network. In practical applications, smart sensors are usually deployed randomly at one time, and power replacement under centralized management becomes impractical. However, smart sensor nodes are powered by limited memory and batteries, and smart sensor networks consume a lot of energy in practical applications. Therefore, the energy efficiency of smart sensor nodes basically determines the life cycle of smart sensor networks, which is crucial to the overall network life. In order to extend the service life, smart sensor networks can save a lot of energy for the network through clustering. The correct choice of cluster head is one of the solutions to this problem. In the work of selecting cluster heads, the commonly used methods are as follows:

(1) Use improved artificial intelligence algorithms when optimizing cluster head selection

(2) Improve the objective function to balance the energy consumption of nodes

In paper [15], a subtractive clustering algorithm is proposed. The solution relies on subtractive clustering to generate cluster head nodes in densely populated areas. The algorithm solves the problem of the ownership of noncluster head nodes so that the consumption of the entire network is evenly distributed and reduces the energy consumption of a part of the network. However, the algorithm converges slowly and its run time is long.

Paper [16] proposes an improved PSO-based fuzzy clustering algorithm. It designs a new objective function by optimizing the movement of particles and specifies a suitable cluster head. That solution overcomes the problems of hot spots and energy holes, but in actual operation, when the number of iterations increases, it is easy to cause premature convergence.

In the paper [17], the chaotic monkey algorithm was used to establish a problem model for the low-energy clustering problem. Simulation experiments show that there is indeed higher energy efficiency in large-scale wireless sensor networks, but it falls into local convergence in smart sensor networks.

In addition to optimizing artificial intelligence algorithms, there are also ways to extend the life of the network by optimizing energy budgets. For example, in data collection or surveillance, unmanned aerial vehicles (UAV) are used to create a more flexible data collection platform, and then, the optimal cluster head selection strategy is proposed. As shown in paper [18], it installs sensors for each UAV and then uses the average remaining energy of the sensor nodes, channel conditions, and Euclidean distance to select cluster head nodes. Although the lifespan of the network has increased compared with the traditional solution, the algorithm complexity is too high to accept.

3. System Model

First of all, the following assumptions are given for the network model.

Assumption 1. The monitoring environment of smart sensor networks is a regular shape, and the sensor nodes are randomly and discretely distributed in the monitoring area.

Assumption 2. All sensor nodes have the same initial energy, and it consumes information when sending, fusing and transmitting. With the same power, the data communication capability is the same, the energy of the base station is not limited, and it is always in a normal working state. Other nodes have limited energy, and they are judged as dead nodes after the energy is zero.

Assumption 3. According to the distance between nodes, the transmission power of each node can be flexibly selected, and the communication between nodes is not restricted.

3.1. Topological Structure Model. Based on the above-mentioned hypothetical network model, the types of smart sensor nodes can be divided into cluster head nodes, gateway nodes, and perception nodes. Among them, the perception node merges the data it monitors and sends it to the cluster head node in a single hop. After that, the cluster head node receives the data from the perception node, then performs data fusion, and finally sends it to its corresponding gateway node. Then, the gateway node aggregates the data and gives it to the cluster head node. The user does further analysis and processing. Then, the user sends the monitoring task to the gateway node and distributes the monitoring target in the network. In the initial state, all nodes with the same state evolve into nodes with the above-mentioned different functions based on the routing protocol and form the topology as shown in Figure 1.

3.2. Energy Consumption Model. This part adopts the energy consumption model in LEACH protocol, that is, different models are adopted according to different distances. To reduce the energy consumption of the network while being restricted by the communication distance of nodes, smart sensor networks must develop efficient clustering schemes for reasonable clustering. The energy consumption of smart sensor networks is mainly composed of communication energy consumption, perception energy consumption, and microprocessing energy consumption. Research shows communication energy consumption such as sending and receiving accounts for more than half of the energy consumption of smart sensor networks. At the same time, the perceived energy consumption and microprocessing energy consumption are relatively fixed, and it is not easy to optimize and reduce them. This part mainly focuses on how to reduce the communication energy consumption of smart sensor networks through reasonable clustering.

The energy consumption formula for sending bit data is shown in

$$\text{cost}_s(k, d) = \begin{cases} E_{\text{elec}}k + \varepsilon_{\text{fs}}kd^2, & d < d_0, \\ E_{\text{elec}}k + \varepsilon_{\text{amp}}kd^4, & d > d_0, \end{cases} \quad (1)$$

where $\text{cost}_s(k, d)$ is the energy consumed by the sending node to send k bits of data to the receiving node with a distance of d. E_{elec} is the electronics energy parameter, ε_{amp} is the power amplification parameter in the multipath fading channel model, and d is the transmission distance between the sending node and the perception node. Among them, ε_{fs} is the power amplification parameter in the free space propagation model.

When the distance between the sending node and the perception node is less than d_0, the free space propagation model (power loss is proportional to d^2) is used. Otherwise, the multipath fading channel model is used (power loss is proportional to d^4).

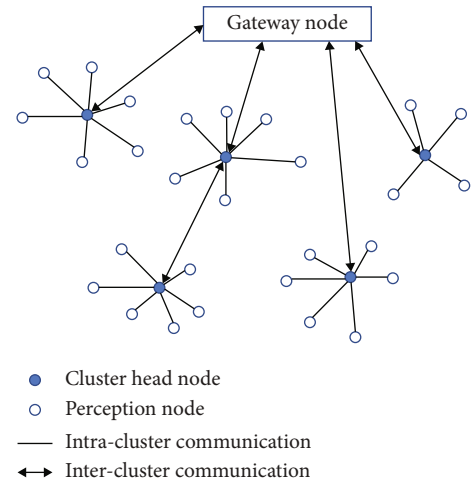


Figure 1: ACACO cluster structure.

The energy consumption of the perception node to receive k bits data can be obtained by

$$\text{cost}_r(k) = E_{\text{elec}}k, \quad (2)$$

where $\text{cost}_r(k)$ is the energy consumed by the perception node to receive k bits data.

4. ACACO for Energy Consumption in Smart Sensor Networks

The traditional clustering routing protocol adopts a random selection method in the election of cluster head nodes, and the possibility of obtaining the optimal clustering is low, which leads to uneven distribution of nodes and high energy consumption for communication within the cluster. To solve this problem, the cluster head node set is selected from the candidate cluster head node set to complete the initialization of the cluster head node, and ACACO is used to optimize the cluster head node set.

4.1. Population Initialization. We use binary individual coding to cluster. First, we number the Q sensor nodes in the area except the gateway node with natural numbers. The binary code of an individual can be represented by a vector. The number "1" means that the sensor at the corresponding position on the vector is a cluster head node, and the number "0" means that the sensor node at the corresponding position on the vector is a perception node. For example, if there are a total of 8 nodes in smart sensor networks and the individual code is "00101011", it means that nodes 3, 5, 7, and 8 are cluster head nodes, and the remaining nodes are perception nodes. The perception node only clusters with the nearest cluster head node, so no coding is required.

In smart sensor networks with Q sensors, as the number of individuals in the population is N and the cluster heads

number is M, the population coding and constraint conditions can be expressed as the matrix form shown in

$$F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,Q-1} & f_{1,Q} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,Q-1} & f_{2,Q} \\ \vdots & \vdots & f_{n,q} & \vdots & \vdots \\ f_{N-1,1} & f_{N-1,2} & \cdots & f_{N-1,Q-1} & f_{N-1,Q} \\ f_{N,1} & f_{N,2} & \cdots & \mathrm{f}_{N,Q-1} & f_{N,Q} \end{bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_n \\ \vdots \\ F_N \end{bmatrix}, \tag{3}$$

$$\sum_{q=1}^{Q} f_{n,q} = M, \ (n\epsilon\{1, 2, \cdots, N\}). \tag{4}$$

In formula (3), $f_{n,q} = 1$ means that the q^{th} sensor in the n^{th} individual is a cluster head node. If it is 0, it is a perception node. Formula (4) constrains the cluster head node number in each individual in smart sensor networks to a fixed value M.

4.2. Fitness Value. In the clustering scheme of smart sensor networks for the purpose of optimizing the energy consumption of a single round of transmission, the ACACO objective function can be expressed in the form shown in

$$\mathrm{fit}(F_n) = \mathrm{cost}_s(k, d) + \mathrm{cost}_r(k). \tag{5}$$

Formula (5) indicates that the communication energy consumption of a single round is the sum of the perception energy consumption and the sending energy consumption in formulas (1) and (2).

4.3. Initial Pheromone Improvement. To ensure that the ACACO can search for multiple transmission paths in the initial stage, this chapter designs and improves the initial state path pheromone. ACACO introduces a chaotic mapping mechanism and replaces the traditional ACO with a chaotic operator to optimize the solution. The reciprocal distance is used to represent the initial pheromone concentration, thereby jumping out of the limitation of the suboptimal path solution. The initial pheromone is obtained by using logistic mapping of the typical nonlinear chaotic sequence in chaos theory, which is defined as

$$\tau_{ij}(t+1) = \mu\tau_{ij}(t)\left(1 - \tau_{ij}(t)\right) \quad \tau_{ij} \in (0, 1), \tag{6}$$

where $\tau_{ij}(t)$ represents the pheromone concentration on the initial path from cluster node i to perception node j in the $(t+1)^{\text{th}}$ iteration and μ is the logistic parameter.

4.4. Transition Probability Formula. In ACACO, a binary route of an ant can be generated according to the pheromone intensity and visibility to update the current optimal solution. The ant's movement rules and transition probability can be shown in

$$P_{ij}^{k}(t) = \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{j\text{eallowed}_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}. \tag{7}$$

Among them, $\eta_{ij}^{\beta}(t)$ represents the path enlightenment information from node i to node j. α is the pheromone track intensity, and β is the path visibility. allowed_k represents the set of perception nodes that ant k has not visited and is stored in the taboo table.

$\eta_{ij}^{\beta}(t)$ is calculated as formula (8). When the energy consumption from the current cluster head node i to the perception node j is less than the original energy consumption, the path enlightenment information is the distance. In this model, the nodes selected by the ants are the cluster heads in smart sensor networks. In order to evenly distribute the cluster heads in the monitoring area to reduce energy consumption, instead of clustering them together, we tentatively determine that the larger the distance between the cluster heads, the better. As the number of iterations increases, the optimal solution will be selected according to the termination condition.

$$\eta_{ij}^{\beta}(t) = \begin{cases} d_{ij}, \text{ if } \mathrm{fit}'(j) < \mathrm{fit}(j), \\ 0, \text{ otherwise}, \end{cases} \tag{8}$$

where $\mathrm{fit}(j)$ is the original fitness value and $\mathrm{fit}'(j)$ is the current fitness value.

4.5. Pheromone Update Strategy. Ants release pheromone during the path search. To prevent the accumulation of too much pheromone and cause the algorithm to stagnate, a pheromone volatilization mechanism is introduced. The pheromone update operation is performed after a complete access path. The specific calculation formula is shown as

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \quad \rho \in (0, 1), \tag{9}$$

where $(1-\rho)\tau_{ij}(t)$ represents the pheromone volatilization process and ρ is the pheromone volatilization rate. $\Delta\tau_{ij}(t)$ is the pheromone increase mechanism and the calculation method is as follows:

$$\Delta\tau_{ij}(t) = \begin{cases} \sum_{k=1}^{m} \dfrac{Q}{L_k} & \text{, if the } k^{\text{th}} \text{ ant passes through the path } (i, j) \text{ in this cycle\#}, \\ 0, \text{ otherwise} \end{cases} \tag{10}$$

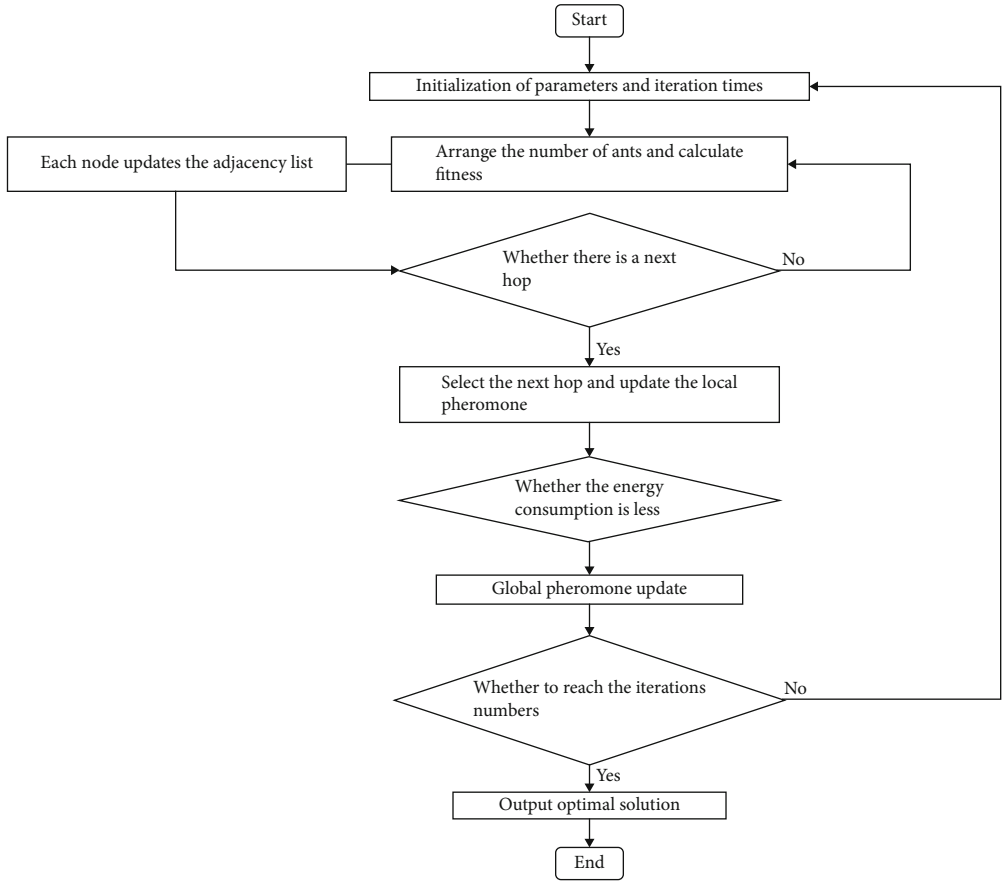


Figure 2: Steps of ACACO.

Table 1: Parameters of the simulation.

Parameter	Parameter symbol	Value
Electronics energy parameter	E_{elec}	50 nJ/bit
Power amplification parameter in the multipath fading channel model	ε_{fs}	10 PJ/(bit × m^2)
Power amplification parameter in the free space propagation model	ε_{amp}	0.013 PJ/(bit × m^4)

where Q is the pheromone constant and L_k is the total length of road by the k^{th} ant in this cycle.

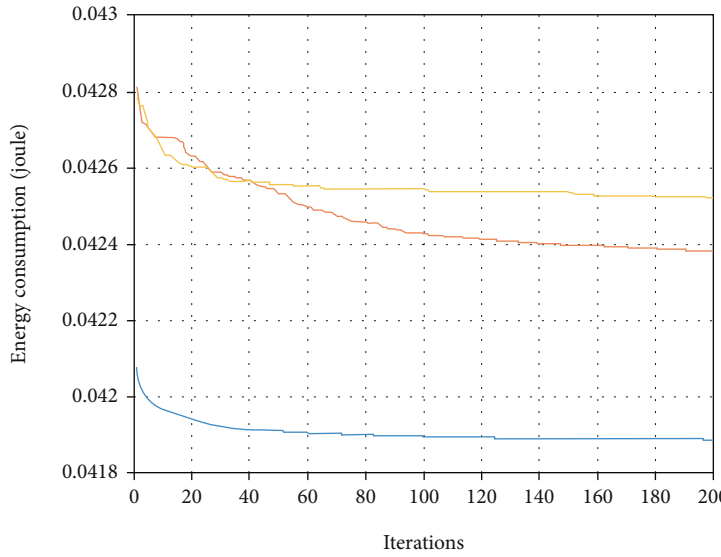
4.6. Adaptive Selection Strategy of Ant Enlightenment Information. After all the ants in the ant colony have moved, the enlightenment information of the ants needs to be updated according to the optimal solution of the previous iteration. In order to collect node and path information in the search process and give positive feedback to the final complete path information, ACACO designs an elite selection strategy.

Set to the t generation, $a(t)$ in the population is the best individual. Assume that $a(t+1)$ is a new generation population. If there is no better individual than $a(t)$ in $a(t+1)$, add $a(t)$ to $a(t+1)$. $a(t)$ will be the n^{th} individual of $a(t+1)$. Here, n is the size of the population. In order to maintain a certain population size, if elite individuals are added to the new population, the individual with the smallest fitness value in the new population needs to be removed.

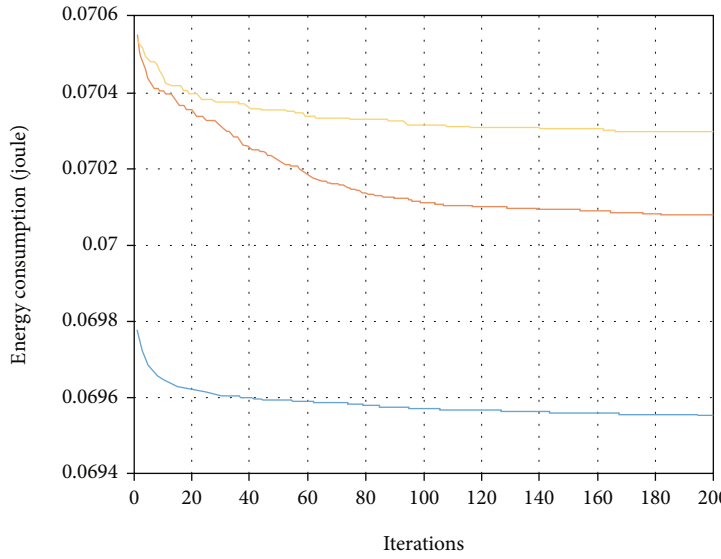
Table 2: Parameters of the ACACO.

	α	β	ρ	Q	μ
ACACO	2	3	0.9	20	0.5

It can be seen from formula (8) that the distance between nodes in this cycle of ants has been selected by elites. Nodes with small path distances are eliminated directly, leaving large distances and increasing the proportion of large path distances. Therefore, the possibility of selecting a node with a large distance to become the cluster head increases, thereby reducing the energy consumption between the cluster head node and the sensing node. Therefore, the elite selection node



(a)



(b)

Figure 3: Continued.

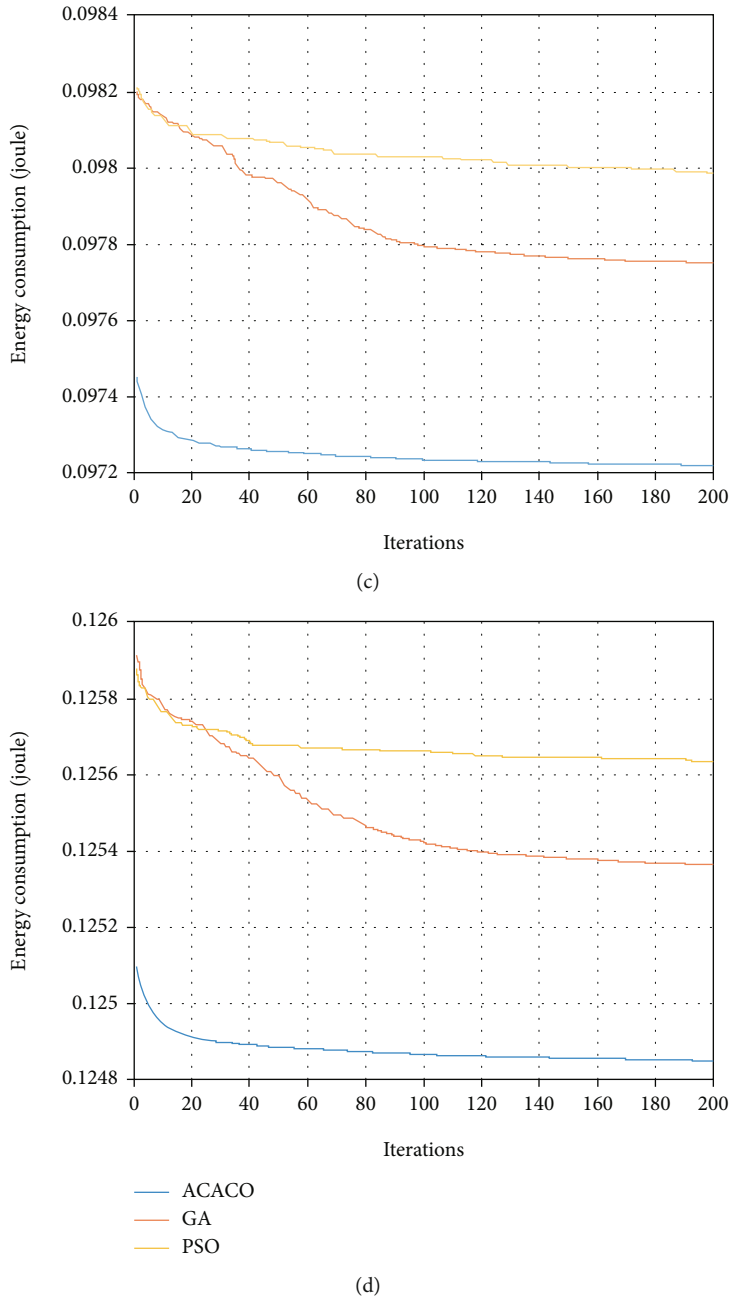


Figure 3: The energy consumption of the three algorithms under different numbers of nodes: (a) 150 node number, (b) 250 node number, (c) 350 node number, and (d) 450 node numbers.

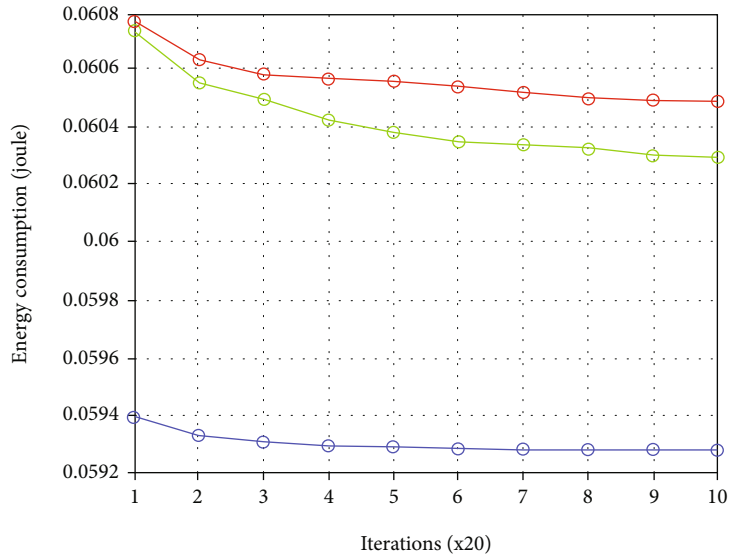
strategy can effectively improve the fitness of weakened ants, thereby improving the overall fitness of the ant colony.

4.7. Termination Condition. The termination condition is the criterion by which ACACO decides whether to continue operation or stop. In the process of repeating the iterative loop, until the solution of the predetermined iterations number is reached. When the target value reaches a certain threshold, ACACO will terminate according to the iteration number. After the maximum iteration number, the process is terminated, and the individual with the lowest network energy consumption in the community is the final solution.

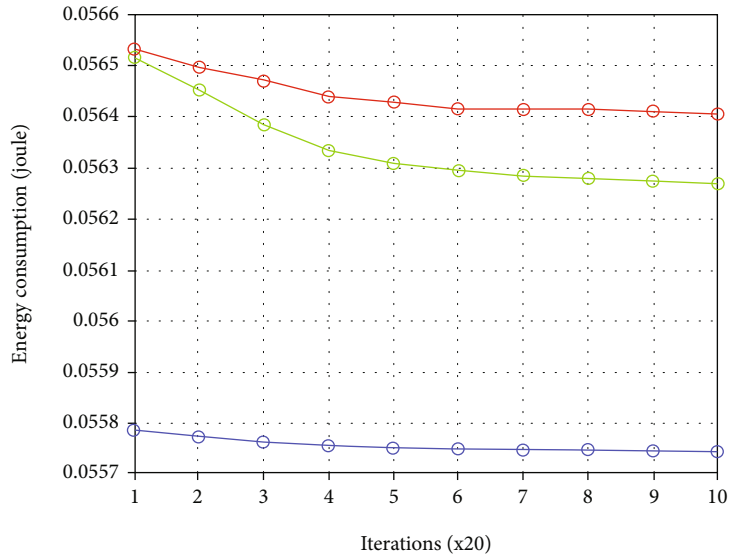
4.8. The Steps of ACACO. The specific steps of ACACO are as follows.

Step 1. The initial ant colony is generated by logistic chaotic mapping, and the parameter initialization includes parameters such as initialization pheromone and various heuristic factors; at the same time, update the number of iterations.

Step 2. Deploy m ants on the original cluster head node and calculate fitness.



(a)



(b)

Figure 4: Continued.

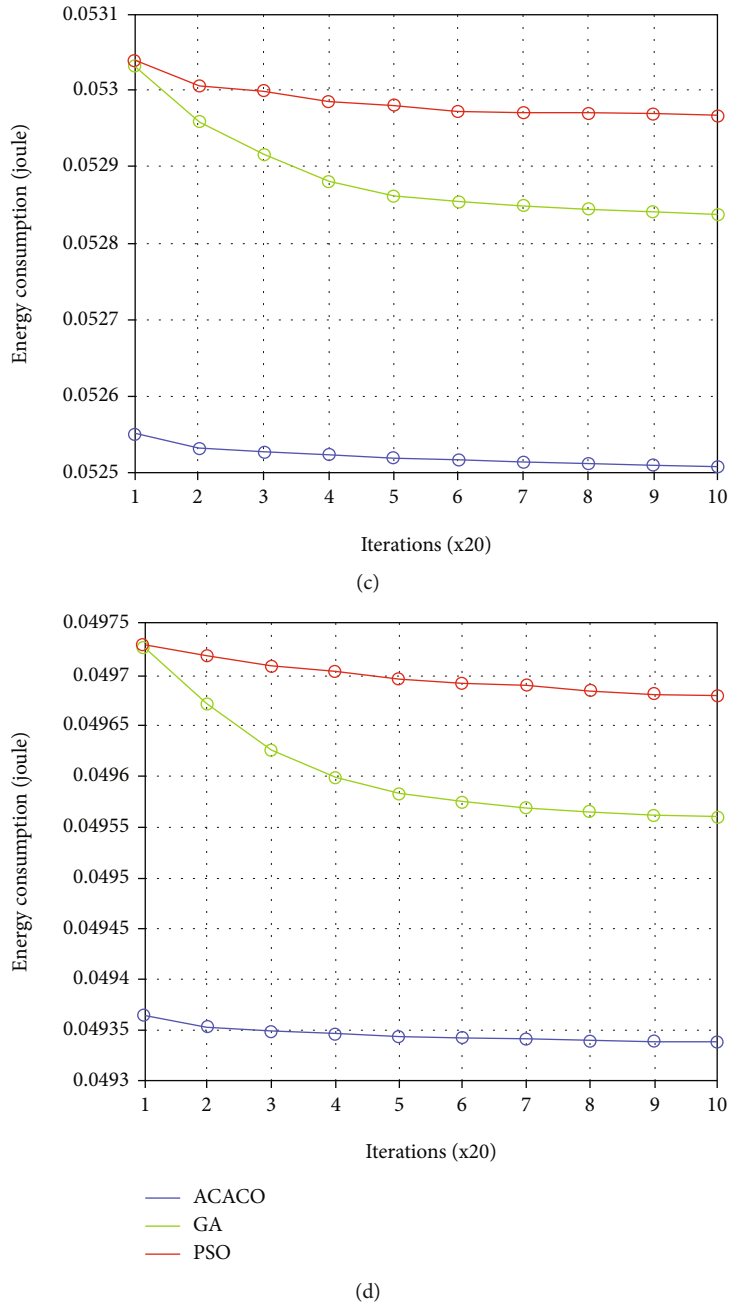


Figure 4: The energy consumption of the three algorithms under different cluster head probabilities. (a) The cluster head probability is 5%. (b) The cluster head probability is 10%. (c) The cluster head probability is 15%. (d) The cluster head probability is 20%.

Step 3. Measure the specific position of each ant and update the adjacency list.

Step 4. Check whether there is a next-hop node to be selected in the adjacency list; if it does not exist, expand the search radius and update the adjacency list; if it exists, go to Step 5.

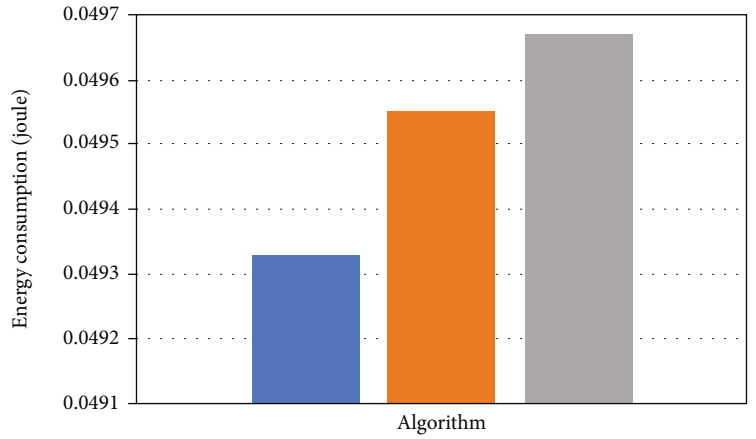
Step 5. According to the transition probability formula, select the next hop node.

Step 6. Record and update node and path information, and perform local pheromone update operations.

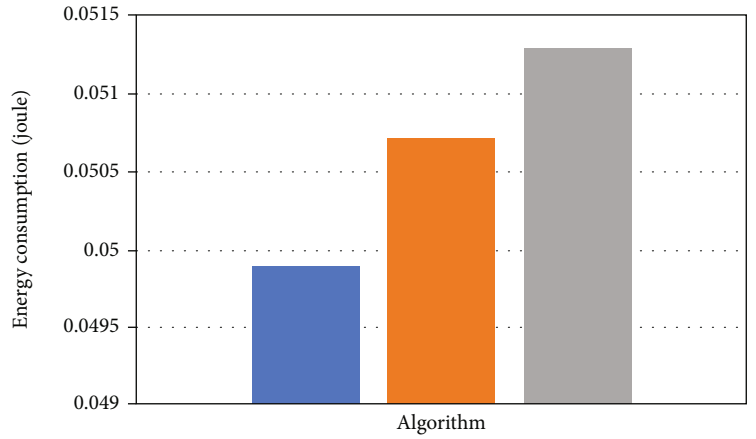
Step 7. Determine whether the energy consumption of m ants is less than the previous one; if yes, execute the elite selection strategy and update the global pheromone; if not, release the ants again.

Step 8. Determine whether the number of iterations of the algorithm is met; if the conditions are not met, continue to iteratively execute Steps 3–7, and terminate the algorithm if the conditions are met and output the result.

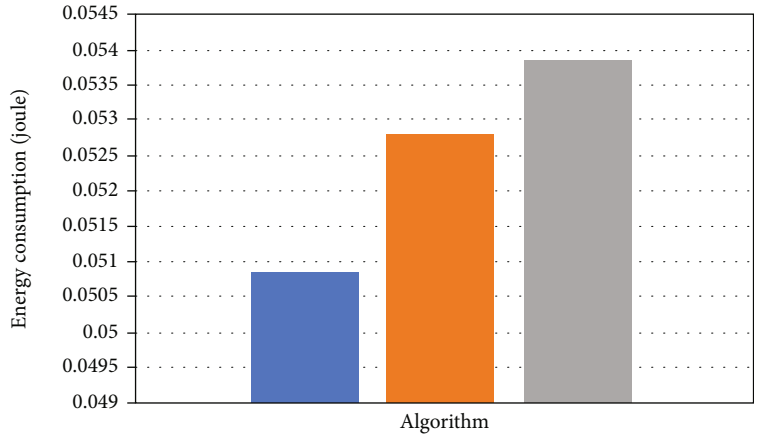
The concrete simulation flowchart is shown in Figure 2.



(a)

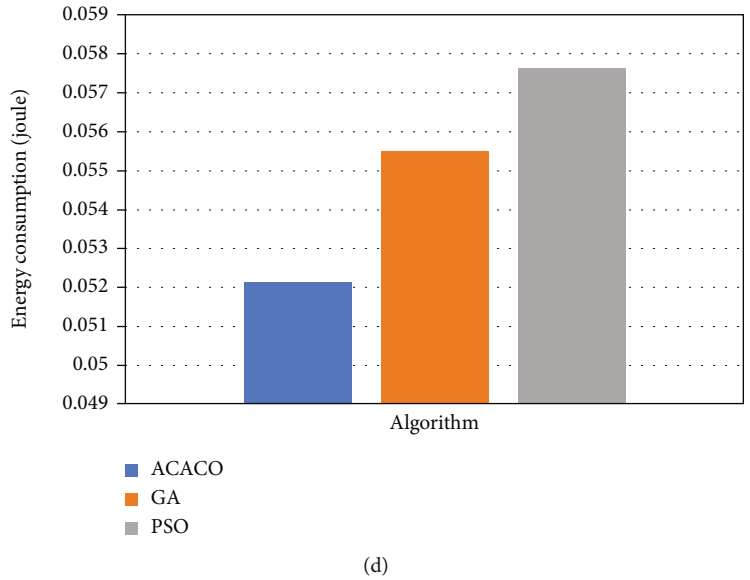


(b)



(c)

Figure 5: Continued.



(d)

FIGURE 5: The energy consumption of the three algorithms under different monitoring areas. (a) The monitoring area is 100 × 100 m. (b) The monitoring area is 200 × 200 m. (c) The monitoring area is 300 × 300 m. (d) The monitoring area is 400 × 400 m.

5. Results and Discussion

This paper uses MATLAB R2018a environment to carry out simulation experiments of three algorithms. To ensure the accuracy of the experimental data, the simulation experiment is run 50 times and the average value is taken as the experimental result. In this section, we compare how different node numbers, cluster head ratios, and monitoring area affect the energy consumption of smart sensor networks.

We uniformly define the parameters for calculating sensor energy consumption in smart sensor networks, to compare these three algorithms under the same experimental conditions. The iteration number is set to 200 generations, the population size is 40, and the coordinates of the nodes are randomly generated in this area. The specific parameters of ACACO are in Table 1.

The key to affecting the behavior and performance of genetic algorithms is the possibility of crossover and mutation. Therefore, in this simulation calculation, as a comparison algorithm, the crossover probability of GA is set to 0.9, and the mutation probability is 0.05. Besides, in PSO, the maximum speed determines the maximum moving distance of the particles in a cycle, which is set to 4. Both cognitive and social parameters are set to 2, namely, $c_1 = c_2$. And the specific simulation parameters of ACACO are set in Table 2. In actual data transmission, packet drop is inevitable, so the packet drop rate from the sensing node to the cluster head is set to 1% in the simulation. Correspondingly, the packet drop rate from the cluster head to the gateway is also 1%. Then, the overall packet drop rate is 98.01%.

Figure 3 shows the energy optimization in smart sensor networks based on ACACO, GA, and PSO when the sensor cluster head ratio is 0.1. In order to determine the energy consumption of the algorithm under different numbers of sensors, in the four subgraphs, the number of sensors is set to 150, 250, 350, and 450, respectively. And the monitoring area is 100 × 100 m.

It can be seen from the simulation results in Figures 3(a)–3(d) that the energy consumption value based on PSO decreases slowly with the increase of the number of iterations of the algorithm. Compared with the ACACO which dynamically adjusts the pheromone, the fixed parameters of GA are more difficult to introduce new genes, which make it difficult to reduce energy consumption. The energy consumption reduction based on PSO is relatively stable, but due to the fixed coding method, it is easy to fall into a stagnant state in the later stage of the algorithm operation. The ants in ACACO use chaotic mapping to dynamically adjust the pheromone concentration and introduce the elite selection strategy to select the next node, avoiding the phenomenon of premature convergence and evolutionary stagnation caused by the deterministic state.

Figure 4 shows the variation of the average communication energy consumption of smart sensor nodes with the ratio of sensor cluster heads when smart sensor nodes move within the monitoring area. The proportions of cluster heads for Figures 4(a)–4(d) are 5%, 10%, 15%, and 20%, respectively. The number of sensors is 200. And the monitoring area is 100 × 100 m.

It can also be seen from Figure 4(a) that when the proportion of cluster heads is 5%, the downward trend of the energy consumption of the three algorithms is not very obvious. However, the advantages of reducing communication energy consumption based on ACACO's simulated evolutionary calculation are obvious compared with the other two algorithms. When the proportion of cluster heads is 10%, 15%, and 20%, as the proportion of sensor cluster heads increases, the energy consumption of GA drops rapidly, and the evolution speed of PSO is slow.

Figure 4(b) shows that when the cluster head ratio is 10%, the PSO-based clustering method has relatively stable

performance during operation, and the energy consumption is not significantly reduced. Based on the GA-based solution for cluster head selection, the energy consumption has dropped significantly. However, compared with the fixed parameters of GA and PSO, ACACO has introduced a global update strategy. The best path obtained after the end of each generation cycle is rewarded for the pheromone content on the path with a positive feedback mechanism. This way means that at the beginning of each generation cycle, the pheromone content has been dynamically changed, the probability of selecting the best path is greatly increased, and the probability of retaining the previous generation of low-power solutions is also increased. Therefore, after the iterative cycle, the network using ACACO has the better performance in terms of energy consumption. In Figures 4(c) and 4(d), the single-round energy consumption decline trends of PSO and ACACO are similar, but ACACO converges faster.

Figure 5 better shows the energy efficiency of these three algorithms in different monitoring areas. In Figures 5(a)–5(d), we set the size of the monitoring area of smart sensor networks to 100×100 m, 200×200 m, 300×300 m, and 400×400 m in turn. The number of sensors is 200, and the cluster head ratio is 0.2.

The histogram in Figure 5 has the same trend. It can be seen that the clustering method based on ACACO calculation requires less average communication energy consumption of nodes, which can effectively improve energy utilization efficiency.

6. Conclusions

We propose a clustering algorithm based on ACACO, which uses heuristic simulation evolution calculation method to dynamically select the number and location of cluster heads to reduce the energy consumption of smart sensor networks. In the iterative process, the algorithm dynamically changes the algorithm parameters through chaotic mapping, avoiding premature convergence, and at the same time, the convergence speed is faster. We also dynamically update the global pheromone content and adopt an adaptive strategy to select the best individual. The simulation results show that compared with the other two schemes, the proposed clustering scheme based on ACACO calculation can effectively reduce the single-round communication energy consumption of smart sensor networks. This paper only considers a single case where there is only one base station in the scenario. When there is a many-to-many relationship between the optional base station and the cluster head node, how to effectively allocate transmission tasks to optimize the network lifespan needs to be considered from multiple perspectives in the future.

Data Availability

The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This paper was funded by the Corps Innovative Talents Plan, grant number 2020CB001; the Project of Youth and Middle-Aged Scientific and Technological Innovation Leading Talents Program of the Corps, grant number 2018CB006; the China Postdoctoral Science Foundation, grant number 220531; the Funding Project for High-Level Talents Research in Shihezi University, grant number RCZK2018C38; and the Project of Shihezi University, grant number ZZZC201915B.

References

[1] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and A. H. Gandomi, "Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network," *IEEE Sensors Journal*, p. 1, 2021.

[2] S. K. Aanchal, O. Kaiwartya, and A. H. Abdullah, "Green computing for wireless sensor networks: optimization and Huffman coding approach," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 592–609, 2017.

[3] S. Kumar, O. Kaiwartya, M. Rathee, N. Kumar, and J. Lloret, "Toward energy-oriented optimization for green communication in sensor enabled IoT environments," *IEEE Systems Journal*, vol. 14, no. 4, pp. 4663–4673, 2020.

[4] F. Lin, W. Dai, W. Li, Z. Xu, and L. Yuan, "A framework of priority-aware packet transmission scheduling in cluster-based industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5596–5606, 2020.

[5] Y. Cao and Z. Wang, "Combinatorial optimization-based clustering algorithm for wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2020, 13 pages, 2020.

[6] A. Pathak, "A proficient bee colony-clustering protocol to prolong lifetime of wireless sensor networks," *Journal of Computer Networks and Communications*, vol. 2020, Article ID 1236187, 9 pages, 2020.

[7] L. Chen, W. Liu, D. Gong, and Y. Chen, "Clustering and routing optimization algorithm for heterogeneous wireless sensor networks," in *International wireless communications and Mobile computing (IWCMC)*, pp. 407–411, Limassol, Cyprus, 2020.

[8] H. L. Wang, G. B. Zhou, L. Bhatia, Z. C. Zhu, W. Li, and J. A. McCann, "Energy-neutral and QoS-aware protocol in wireless sensor networks for health monitoring of hoisting systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5543–5553, 2020.

[9] F. Sanhaji, H. Satori, and K. Satori, "Cluster head selection based on neural networks in wireless sensor networks," in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–5, Fez, Morocco, 2019.

[10] P. K. Kashyap and S. Kumar, "Genetic-fuzzy based load balanced protocol for WSNs," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 2, pp. 1168–1183, 2019.

[11] S. H. Sackey, J. Chen, A. J. Henry, and X. Zhang, "A clustering approach based on genetic algorithm for wireless sensor network localization," in *2019 15th international conference on computational intelligence and security (CIS)*, pp. 54–58, Macao, China, 2019.

[12] S. Kumar and S. Mehfuz, "A PSO based malicious node detection and energy efficient clustering in wireless sensor network," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 859–863, Noida, India, 2019.

[13] P. S. Lakshmi and M. G. Jibukumar, "Energy efficient MAC protocol for lifetime maximization in wireless sensor networks," in *2018 IEEE 13th international conference on industrial and information systems (ICIIS)*, pp. 191–195, Rupnagar, India, 2018.

[14] M. Abdelhafidh, M. Fourati, L. C. Fourati, A. B. Mnaouer, and M. Zid, "Linear WSN lifetime maximization for pipeline monitoring using hybrid K-means ACO clustering algorithm," in *2018 Wireless Days (WD)*, pp. 178–180, Dubai, United Arab Emirates, 2018.

[15] L. Chen, W. Liu, D. Gong, and Y. Chen, "Cluster-based routing algorithm for WSN based on subtractive clustering," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 403–406, Limassol, Cyprus, 2020.

[16] G. Kadiravan and P. Sujatha, "PSO based clustering approach for mobile wireless sensor network," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–5, Pondicherry, India, 2019.

[17] Y. Lu, J. Zhou, and M. Xu, "A biologically inspired low energy clustering method for large scale wireless sensor networks," in *2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pp. 20–23, Fuzhou, China, 2019.

[18] S. K. Haider, M. A. Jamshed, A. Jiang, H. Pervaiz, and Q. Ni, "UAV-assisted cluster-head selection mechanism for wireless sensor network applications," in *2019 UK/ China Emerging Technologies (UCET)*, pp. 1-2, Glasgow, UK, 2019.

Hindawi
Journal of Sensors
Volume 2021, Article ID 5575802, 16 pages
https://doi.org/10.1155/2021/5575802

Research Article

Traffic and Energy Aware Optimization for Congestion Control in Next Generation Wireless Sensor Networks

Saneh Lata Yadav,[1] R. L. Ujjwal,[1] Sushil Kumar,[2] Omprakash Kaiwartya,[3] Manoj Kumar,[2] and Pankaj Kumar Kashyap[2]

[1]*School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, New Delhi 110078, India*
[2]*School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India*
[3]*Department of Computer Science, Nottingham Trent University, Nottingham NG11 8NS, UK*

Correspondence should be addressed to Omprakash Kaiwartya; omprakash.kaiwartya@ntu.ac.uk

Received 20 March 2021; Revised 30 May 2021; Accepted 9 June 2021; Published 30 June 2021

Academic Editor: Xavier Vilanova

Congestion in wireless sensor networks (WSNs) is an unavoidable issue in today's scenario, where data traffic increased to its aggregated capacity of the channel. The consequence of this turns in to overflowing of the buffer at each receiving sensor nodes which ultimately drops the packets, reduces the packet delivery ratio, and degrades throughput of the network, since retransmission of every unacknowledged packet is not an optimized solution in terms of energy for resource-restricted sensor nodes. Routing is one of the most preferred approaches for minimizing the energy consumption of nodes and enhancing the throughput in WSNs, since the routing problem has been proved to be an NP-hard and it has been realized that a heuristic-based approach provides better performance than their traditional counterparts. To tackle all the mentioned issues, this paper proposes an efficient congestion avoidance approach using Huffman coding algorithm and ant colony optimization (ECA-HA) to improve the network performance. This approach is a combination of traffic-oriented and resource-oriented optimization. Specially, ant colony optimization has been employed to find multiple congestion-free alternate paths. The forward ant constructs multiple congestion-free paths from source to sink node, and backward ant ensures about the successful creation of paths moving from sink to source node, considering energy of the link, packet loss rate, and congestion level. Huffman coding considers the packet loss rate on different alternate paths discovered by ant colony optimization for selection of an optimal path. Finally, the simulation result presents that the proposed approach outperforms the state of the art approaches in terms of average energy consumption, delay, and throughput and packet delivery ratio.

1. Introduction

The advancement in low-cost, small, and tiny sensor nodes makes a significant role where the sensor nodes have very attractive characteristics of sensing the environmental conditions and process the received signals. The sensor nodes can be deployed in all accessible and inaccessible areas for sensing the data across various applications like battlefield, building inspection, target field imaging greenhouse, and monitoring disaster area [1]. The deployment of sensor nodes is application-dependent, so it can be random or deterministic [2]. The sensor nodes of the wireless sensor network (WSN) sense the data or event, gather the data under defined infrastructure, and process the received signals. After a lot of advancement in wireless sensor networks, it is lacking with few specifications like limited memory, inadequate computation, limited bandwidth, and battery-powered nodes [3, 4]. Due to the short communication range of sensor nodes, the intermediate nodes collaborate in forwarding the data packets. There are various applications of WSN where sensor nodes are deployed in an infrastructure-less network. The sensor nodes sense an event and report to the nearest base station for respective action. To obtain the quality of service such as congestion-free end-to-end data delivery and delay-free data transmission, an effective and efficient network is required to design to

tackle congestion and energy issues of WSN [5, 6]. When a node receives data more than its capacity to process, congestion may occur which leads to the retransmission of unacknowledged packets. But this frequent retransmission of packets may deplete the energy of the nodes. The nodes in WSN are battery-powered, so energy efficient decisions are desired solution to maximize the lifetime of the network [7]. Load balancing, duty cycling, and data aggregation are the various traditional approaches that have been studied and implemented on WSN. But due to the rapid increase in sensor nodes, the traditional approach faces many issues. Nowadays, trajectory-based data forwarding, mobile sink, and energy supply based on node selection approaches are used for residual energy consumption [8, 9].

WSN is bounded with limited transmission and processing capabilities; therefore, the energy of sensor nodes and successful data transmission are the two important and mandatory requirements. There are four popular applications of WSN such as event-oriented, query-oriented, continuous, and hybrid applications. The application performs on few predefined situations, resulting in an unforeseeable traffic rate. The applications where one sensor node transmits a query and another sensor node has to respond to it are known as query-oriented applications. On the other side, continuous applications perform periodically or in few time slots. Lastly, the hybrid applications are the fusion of the above three applications [6]. The transport layer deals with the congestion problem, which is accountable for end-to-end network connection. The objective of the transport layer is to provide fair bandwidth allocation, control the data flow rate with reliable connection links, and retransmit lost packets in an energy-efficient fashion. Few existing strategies for controlling congestion are a slow start, congestion avoidance, fast retransmission, and fast resumption, but these are not able to outperform in today's network scenario [10]. Therefore, an effective and efficient CC mechanism under few constraints, such as resource, performance, and scalability, is in high demand. Congestion is noticed on two locations, namely, node or link. When the node's buffer is fully occupied owing to the fast arrival rate of incoming packets, node-level congestion happens. When data packets collapse on network/radio links owing to the fast transmission of data packets, link-level congestion is noticed. The performance is influenced by both types of congestion. Congestion should be detected at its early stage, then be notified to the nearby nodes to the source node, and eventually, control it [11]. Traffic control and resource control are the two most popular approaches for this domain. When the data flow rate is controlled, traffic controlled approach is considered, while the resource control approach looks after the fair allocation of resources to divert the data on a less congested path [12]. Both the approaches contemplate the node's priority, which plays the foremost role in dealing with congestion. To address the abovementioned challenge, we present an energy-efficient, robust, and heuristic-based approach to boost the lifetime of the network, provided congestion-free. Therefore, major research contributions are as follows:

(1) An efficient congestion avoidance approach using Huffman coding enabled ant colony optimization (ECA-HA) in wireless sensor networks is proposed considering the important constraints of WSN such as energy, packet loss rate, and congestion level

(2) Ant colony optimization has been applied to search congestion-free alternate paths whenever there is a hike in congestion on the current path. The forward ant constructs multiple congestion-free paths from source to sink node, and backward ant ensures about the successful creation of paths moving from sink to source node, considering energy of the link, packet loss rate, and congestion level

(3) Packet loss probability is computed for each path using maxima entropy principle. Packet entropy helps in evaluating the uncertainty of congestion degrees on alternate paths considering packet loss rate

(4) Huffman coding is used to select an optimal path to direct the load. The optimal path is the best path among multiple paths identified by ACO in terms of energy, congestion level, and packet loss rate. Huffman coding considers the packet loss probability on different alternate paths discovered by ant colony optimization for selection of an optimal path

(5) Finally, performance of the proposed (ECA-HA) approach is compared with the state of the art approaches

The rest of the paper is organized as follows: in Section 2, literature review related to congestion avoidance and control is discussed. The efficient congestion avoidance approach is elaborated in Section 3 that includes problem statement and congestion indicator model followed with Huffman coding enabled ant colony optimization concepts. The simulation and results of proposed mechanism (ECA-HA) are explained in Section 4. Section 5 elaborates the conclusion of the proposed mechanism.

2. Literature Review

This section explains the literature survey of different research works concerned with congestion avoidance and congestion control algorithm.

2.1. Nonnature-Inspired Techniques. This proposed work is aimed at avoiding congestion-like situations in WSN which is a very challenging issue. It directly affects the throughput of the network. It broadly occurs when the sensor nodes in WSN send data to a link more than its capacity to process. So, when these nodes accept data packets with a higher rate than their capacity, then the data packets may get dropped in between the path or face delay. If the packets acknowledged are not received, the packet is considered as dropped or lost. Such an unacknowledged packet requires retransmission. But this retransmission of lost/drop packet has a limit. Retransmission of the lost packet again and again may degrade the network performance. So, to tackle such situations, various algorithms have already been proposed. These

algorithms are not suitable for all WSN applications. CODA (congestion detection and avoidance) algorithm is proposed by Wan et al. The mechanism works on three broad steps: congestion detection, congestion notification, and congestion control. Congestion detection is done using buffer occupancy and channel status. This is a mechanism to detect a faulty situation before their occurrence. After detecting congestion, a notification can be done by the backpressure method. It applies hop by hop backpressure notification. This notification can be implicit or explicit. Both notifications have its advantages and disadvantage. To notify, it sends a beacon message implicitly or explicitly to the backward node, which increases overhead on an already congested link. For controlling purpose, AIMD (additive increase multiplicative decrease) is used, which additively increases and decreases the data rate [13]. CCF (congestion control and fairness) is proposed by Brahma et al. for many to one routing to control congestion and a transparent end-to-end packet delivery. Fairness is evaluated based on the ratio of the number of packets transmitted and the number of packets received [14]. Kasyap and Kumar proposed Trickle which is a self-regulating algorithm for code propagation and maintenance. It broadcast updated data information to disseminate any sort of data [15]. PCCP (priority-based congestion control) algorithm is proposed by Wang et al., which prioritizes the nodes for critical situation. This method uses a parameter congestion degree by evaluating the interarrival time and service time of packets. In any critical situation, prioritized packet may process first then other packets, which is aimed at maintaining the overall performance of the network [16]. CAF (congestion avoidance and fairness) proposed by Ahmad and Turgut calculates the characteristic ratio of the number of downstream nodes and upstream nodes. It monitors buffer occupancy of downstream nodes to avoid congestion. It fairly balances the load by monitoring buffer occupancy levels [17]. DAlPaS (dynamic alternate path selection algorithm) proposed by Sergiou and Vassiliou measures congestion by the increased capacity of the link and opts alternate congestion-free path to flow the data rate. Various optimization techniques were also introduced for congestion avoidance and control [18].

2.2. Nature-Inspired Techniques. All the above approaches lead to excessive communication load and battery power depletion. These traditional approaches are not suitable for complex problems and do not provide the optimal solution. These drawbacks cannot be recovered easily in the communication networks. These are not favorable for a dynamic and unpredictable environment of WSN applications. So, it is a need of time to introduce self-adaptive approach. Self-adaptation is the strength of nature-inspired techniques. Nature-inspired techniques are very attractive to compute issues that rose in WSN. Nature-inspired techniques are performing the complex tasks in a seemly manner with limiting resources and capability. These techniques take advantage of various disciplines for computing complex tasks with significant features of emerging fields. These algorithms are developed by drawing inspiration from nature, social, and local behavior leading to emergent global behavior of reducing congestion. These take knowledge from various branches of science like chemistry, mathematics, physics, biology, and engineering which helps in developing computation tools for complex problems. Nature-inspired approaches have opted when a problem is nonlinear and complex, with the huge number of potential solutions and objectives.

Based on produced solutions, optimization problems are divided into two categories such as deterministic and nondeterministic (stochastic) algorithms. Deterministic algorithms are conventional and classical algorithms that are based on mathematical programming like linear or nonlinear programming, quadratic programming, gradient-based, or gradient-free methods. Nondeterministic algorithms exhibit some randomness and produce variant results for different problems. Stochastic algorithms explore different search spaces to obtain global optimum escaping local optima. It is more capable of handling NP-hard problems. Optimization techniques like ACS (adaptive cuckoo search), bird flock behavior, artificial honey bee, fireflies and particle swarm intelligence, honey pot, and gravitational search algorithm, and ant colony optimization are some nature-inspired approaches. These all nature-inspired optimization techniques help in providing the optimum solution to control congestion to maintain performance and throughput of WSN. The adaptive cuckoo search uses cuckoo behavior for an optimal rate adjustment to minimize congestion. It adjusts the share rate within the limits of the packet service rate. It uses a fitness function for share rate, provided the congestion of minimum drop at the congested link. This method maximizes the system performance [19–21]. Flock-CC is based on bird behavior proposed by Antoniou et al. This approach adopts a swarm intelligence paradigm where packets are guided to form a flock and flow towards the sink like a global attractor. It dynamically balances the loads within available network resources. The method provides graceful performance in terms of packet delivery ratio, packet loss rate, and delay. This approach is very scalable and can be adopted in the complex networks [22]. REFIACC proposed by Kafi et al. is reliable, efficient, fair, and interference-aware congestion control protocol which schedules the communication to prevent interference in inter and intrapath hot spots. Maximum utilization of available bandwidth is done which outperforms the overall performance [23]. PSO (particle swarm intelligence) is inspired by fish schooling and bird swarm intelligence. Each particle flows in some direction by learning from its own experience or experience of its companion. All individual particle moves cooperatively and competitively in search space synchronously to obtain the best solution [24, 25]. Every particle focuses to move on the best neighbor node on the path. The particle moves from one location or node to another best possible node with a regulating velocity. It discovers p best and g best solution. Finally, fitness function value for each node is obtained. Further, GSA is used in collaboration with PSO, where best performer nodes are attracted towards sink nodes. Heuristic and metaheuristic approaches are mainly two derivative-free stochastic optimization algorithms. Metaheuristic approaches provide better results for complex problems as compared to the heuristic approach. So, the work goes

towards a metaheuristic approach because it provides an optimal solution after considering various concerned parameters. It discovers results using the trial and error phenomenon. It includes a group of search agents that explore the most feasible output depending on randomization and few specific rules. These rules are nature-inspired [26, 27]. Lee and Teng proposed an improved version for LEACH clustering protocol to reduce packet loss rate as well as to prolong the network lifetime using fuzzy inference systems. The approach is named as enhanced hierarchical clustering approach. The nodes with higher residual energy, slower moving rate, and longer pause time would be selected as cluster head [28]. To conserve energy and packet loss in WSN, El Alami and Nagid work on routing techniques considering the mobility of sink as the significant challenge for packet loss rate. The technique named routing-Gi proves itself as more efficient than the existing techniques in terms of lifetime of network, energy efficiency, and packet delivery rate [29]. El Alami and Nagid further adopt an enhanced clustering hierarchy approach to maximize the network lifetime, where the nodes with higher energy will gather data to transmit it to the base station. This approach resolves the issue of redundant data collection by nodes also. It works in a sleeping-waking mechanism that conserves the energy consumption. This approach is utilized in homogeneous and heterogeneous networks [30]. Sangeetha et al. introduced a heuristic approach for searching an energy efficient optimal path in WSN. The author focused on adjusting node degree and topology periodically to save battery power. After this, the data flow is balanced using fuzzy logic to avoid congestion. If congestion has occurred, it finds an alternate best path using learning real-time A star heuristic algorithm [31]. Logambigai et al. continue to work on heuristic approaches, and another energy efficient grid-based clustering approach was introduced with intelligent fuzzy rules. In this method, the routing is performed using grid coordinator with fuzzy rules considering minimum intermediate nodes in routing process. This approach is good in terms of lifetime of network and energy [32]. One more improved form of congestion aware routing mechanism using fuzzy rule sets is proposed by Sangeetha et al. This is a traffic prudent method which identifies more reliable path and handles excessive traffic using fuzzy rule prediction. It works in two segments; one is to identify path through positioning nonlocalized nodes, and another is to identify routes to mitigate congestion free path. The congestion estimate is done through ECFM algorithm [33]. One more clustered gravitational and fuzzy-based energy efficient approach was proposed by Selvi et al. to address the limitations and challenges of existing routing system. It utilizes a heuristic gravitational clustering approach to provide an optimal solution for effective routing and efficient clustering. The most appropriate route with cluster head nodes is chosen after applying fuzzy rule sets. The approach increases the lifetime of the network and reduces the consumption of energy [34]. ACO is a nature-inspired metaheuristic mechanism carries few inherent features that manifest excellent scaling characteristics. Metaheuristic techniques outperform to mitigate congestion and significant for large-scale networks as well. The objective of ACO is to discover congestion-free alternate paths. ACO prospects alternate paths by generating and forwarding artificial ants on search space. Like real ants' budge on search space in search of food, they budge one after another perceiving the existence of pheromone. The probabilistic movement from one node to another allows the ants to prospect new and safe paths. The density of pheromone attracts other ants to budge on the best possible path. Also, heuristic values of different paths incorporate a vital role in adopting the most appropriate path [35–39]. The heuristic value determines local information about the path which is cooperating in evaluating congestion degree. The data rate on different routes is uncertain or random. So information entropy or packet entropy is measured here along with buffer entropy, to calculate the packet loss rate. The packet loss rate helps measure congestion degree and move towards the best solution. The entropy function is used to evaluate the probability of congested paths via packet loss probability on different paths. This indicates the uncertainty of the flow of data, which can be the cause of congestion. Entropy is a concept used to measure the disorder, uncertainty, or randomness of a system. Information theory introduced entropy. The concept of entropy was first introduced for statistical thermodynamics. But now, its applications are communication networks, biological research, and many more. In communication networks, it is widely used to measure the abnormality degree of an event by monitoring abnormal events which helps improve performance metrics of the network like throughput and energy efficiency. Entropy is mainly studied in three categories such as Shannon entropy, Renyi entropy, and Tsallis entropy. Most of the studies discussed only Shannon and Tsallis entropies. Shannon entropy is discovered by C. E. Shannon, and it is extensive, while Tsallis entropy is nonextensive. Shannon entropy is a quantitative measure of uncertainty in a data set. Tsallis entropy explores problems with multifractal structure for long-range dependence. It focuses on the effectiveness of entropy in controlling congestion. Shannon's work is extended by Jaynes named maximum entropy principle which has the inherent property to optimize entropy measure when incomplete information is provided in moment constraint form [40–44]. Shannon expressions and their relevance in the proposed technique are well described in the congestion indicator model section. Table 1 includes the notations used in this paper.

3. Proposed Efficient Congestion Avoidance Approach:(ECA-HA)

In this section, the problem statement and congestion indicator model are presented to elaborate the congestion issue and behavior of buffer occupancy. Alternate congestion-free path selection using ACO and Huffman coding is proposed.

3.1. Problem Statement. For monitoring purposes, N sensor nodes are deployed in a network size A. The WSN is represented by a connecting graph $G = \{V, E\}$ where $V = \{v_1, v_2, v_3, v_4, v_5, v_6 \cdots \cdots .. v_n\}$ a set of vertices and $E = \{e_1, e_2, e_3, e_4, e_5, e_6 \cdots \cdots e_n\}$ a set of connecting edges between sensor nodes. V_n node is considered as sink node and has an adequate energy resource as shown in Figure 1. All the

Table 1: Notations and their description.

Notation	Description
EA	Average residual energy
E_i	Initial energy
E_T	Energy consumed in transmitting
E_R	Energy consumed in receiving
D_i	Delay
E_{amp}	Amplifying energy
d	Distance from source to base station
$E(\text{link})$	Energy consumed on link
$h(\text{link})$	Intermediate node count on link or hop count
N_L	Lifetime of link
N_{lifetime}	Lifetime of complete network
D_{pkt}	Data packet
λ_A	Data arrival rate
λ_D	Data service rate/departure rate
BO_V	Virtual buffer occupancy
BO_{Max}	Maximum buffer size
L_C	Congestion level
thr_{max}	Maximum threshold of buffer
TTL	Time to live
$P_{ij}^{(t)}$	Probability of selecting next hop
A_{ij}	Pheromone density
B_{ij}	Heuristic value
R_{ij}	Updated pheromone value
Mk	Memory to store visited nodes
W_{ij}	Load to distance ratio
E_V	Evaporation factor
Li	Load on i node
d_{ij}	Distance from i node to j node
M_P	Message generated per unit time
$P_L(N)$	Packet loss probability

nodes are static once deployed in the network. The initial energy and communication range of all the nodes are equal. The nodes within each other's communication range are assumed to be adjacent nodes. E_i is the initial energy of all sensor nodes where $i = 1, 2, 3 \cdots n$. E_A is the average residual energy. $(E_A - E_i) = 0$, D_i is the delay, and $h(\text{link})$ is the hop count. When V_i a source node transmits data packets to V_n (sink node), the energy level is decreased in each transmission and receiving of the data packet. E_T and E_R are the energy consumed in transmitting and receiving k bits of information from V_i to V_n.

$$E_T(V_i, V_n) = k\left(E_d + E_{\text{amp}} * d^2\right), \tag{1}$$

$$E_R(V_i, V_n) = k(E_d). \tag{2}$$

E_T and E_R are the energy consumed in transmitting and receiving one bit. E_{amp} is the amplifying energy, and E_d is the energy dissipation per bit in transmitting and receiving. Distance between V_i and V_n. is d. Total energy consumed on the link is the sum of energy consumed in transmitting and receiving.

$$E(\text{link}) = \sum_{1}^{n-1}(E_T + E_R). \tag{3}$$

This can be simplified as energy consumed in transmission and receiving of k bit data, multiplied by several intermediate nodes including the source node. Suppose v_1 as source node and $v_2, v_3, \text{v}_4 \cdots .v_{n-1}$ are intermediate nodes before reaching V_n. The node V_1 to V_{n-1} be intermediate nodes between V_i and V_n.

$$(v_1, v_2, v_3, v_4 \cdots .v_{n-1}) \in h(\text{link}), \text{where } h(\text{link}) \text{ is the hop count of the link.} \tag{4}$$

We define link lifetime (N_L) using the minimum residual energy as

$$N_L = \frac{E_{\min}(\text{link})}{(E_T + E_R)}. \tag{5}$$

When one node is depleted its all energy, it will not participate in future network communication. Using equation (5), lifetime of complete network can be expressed as

$$N_{\text{lifetime}} = \min_{\text{link}\in n}(N_L). \tag{6}$$

Equation (6) can be expressed as

$$N_{\text{lifetime}} = \min_{\text{link}\in n}\left(\frac{E_{\min}(\text{link})}{(E_T + E_R)}\right). \tag{7}$$

We assume that the nodes in WSN are deployed in Gaussian distribution fashion. Gaussian distribution for a random variable x can be defined as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}\, e^{-1(x-\mu)/2\sigma}. \tag{8}$$

Mean or expected value of the data packet is denoted by μ, and standard deviation is σ. It is experienced that if sensor nodes are distributed in Gaussian fashion, the probability of congestion detection is higher as compared to any other distribution strategy. D_{pkt} is the data packets, travelled between v_i and v_n. When arrival rate of source node exceed the service rate of sink node, the data packets are required to wait in the buffer of the sink node. In WSN, each sensor node has a limited buffer space in which waiting data packets can be hold till their service time. When the data

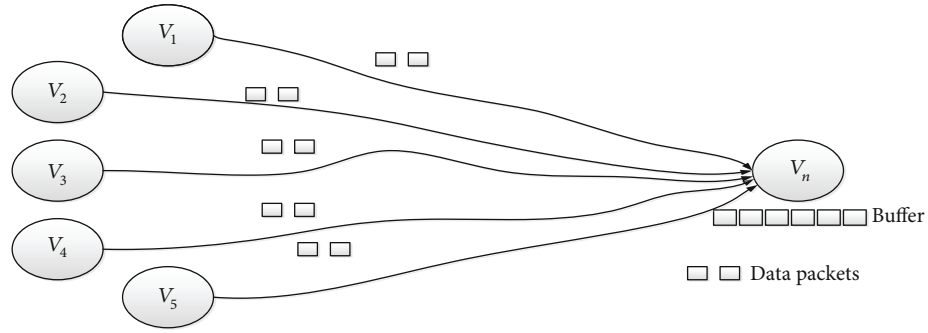


Figure 1: Wireless sensor network communication.

packets exceed the limit of buffer, congestion occurs and the data packets may get dropped. This indicates the congestion on the path. Data arrival rate is the average time elapsed from arrival to successfully process from queue. Let λ_A be the arrival rate and λ_D be the service rate or departure rate of D_{pkt}. The following conditions should be met to declare the network congestion-free or congested.

$$\lambda_A > \lambda_D \text{OR } \lambda_A = \frac{1}{\lambda_D}, \text{congestion occurs,} \tag{9}$$

$$\lambda_A < \lambda_D, \text{congestion-free transmission.} \tag{10}$$

Interarrival time of data is $1/\lambda_A$, and data departure rate is $1/\lambda_D$. Also, congestion can be predicted by calculating message generated per unit time (M_P) from a particular node plus message arrival rate. If this is greater than message departure rate of that particular node then it is also a clear indication of congestion in near future. It can be simplified as

$$\frac{1}{\lambda_A + M_P} > \lambda_D. \tag{11}$$

This needs retransmission of dropped data packets. Frequent retransmission of D_{pkt} may lead to queue delay, more energy consumption, and increase congestion. Congestion deteriorates network lifetime and quality of service. The queue occupancy level can be the indicator of congestion. In worst case, if congestion on path occurs, it must be controlled. So to control congestion, the flow of data is redirected to an alternate congestion-free path. In such case, metaheuristic approach is expected to be adopted to redirect the flow on an alternate congestion-free path. Metaheuristic technique promises fast, effective, and efficient relief from congestion. The proposed approach ECA-HA is aimed at controlling congestion by identifying optimal alternate route for data flow and by considering different parameters.

3.2. Congestion Indicator Model. This section elaborates the behavior of buffer occupancy that eventually affects congestion level followed with notification regarding congestion on path. Virtual buffer occupancy (BO_v) by incoming packets and BO_{Max} is the actual maximum buffer size of a node. If BO_v level exceed to BO_{Max} or buffer occupancy of incoming packet is 95% or more, then it is an indication of congestion. 5% buffer space is left for the packets on fly to prevent them from drop. Here, thr_{max} is the upper threshold of the buffer. Once the congestion indication is favorable, it is mandatory to evaluate congestion level

(a) Buffer occupancy status

BO_{Max} and BO_v are buffer occupancy maximum and virtual buffer occupancy.

If

$$\sum_{V}^{n} (\text{BO}_v > \text{BO}_{\text{Max}}.\text{thr}_{\text{max}}) \text{ OR } \text{BO}_v > 0.95 * \text{BO}_{\text{Max}}, \tag{12}$$

where $v = 1, 2, 3, \cdots \cdots .n$, congestion occurs.

(b) Level of congestion (L_C)

If

$$\frac{\text{BO}_v}{\text{BO}_{\text{Max}}} >= 1, L_C = \frac{\text{BO}_v}{\text{BO}_{\text{Max}}} - 1. \tag{13}$$

Else

$$L_C = 0. \tag{14}$$

Congestion on intermediate node can be computed as

$$L_{C_n} = \sum_{i=1}^{n} (V_i * L_c). \tag{15}$$

L_{C_n} is the level of congestion at node n. $V_i = V_1, V_2, V_3 \cdots \cdots .V_n$ are nodes of network. The bandwidth of network also decreases with increasing level of congestion. As the incoming packets increases to its thr_{max} level, congestion level also increases.

(c) *Notification for Congestion*. When a congestion indicator provides evidence for congestion occurrence, it must implicitly or explicitly notify backward nodes. So to reduce extra overhead and an implicit notification, beacon packet is sent to the source node.

(d) *Congestion Control*. To control drop rate, the data flow must be redirected to some alternate route. This alternate route must be congestion-free. So to choose alternate congestion-free route, ant colony optimization technique is used. ACO helps in identifying multiple alternate congestion-free paths. Huffman coding helps in selecting an optimal path from multiple alternate paths. The proposed approach is preemptive which monitors the occurrence of an event before its occurrence and avoids its initiation.

3.3. Alternate Congestion-Free Path Selection. This section explains the steps involved to obtain multiple congestion-free alternate paths. Each sensor node maintains a routing table that contains the following information as source ID, sink ID, F-ANT ID (forward ant), distance between source and sink in terms of number of intermediate hops, residual energy of node, and packet loss rate and congestion level of node and pheromone value. Ant colony optimization method constructs multiple alternate paths between sources to sink. F-ANT (forward ant) and B-ANT (backward ant) are two types of ants used in constructing paths. F-ANT moves from source to sink and gathers information in the forward direction. It will initialize a pheromone value in this direction. Then, the pheromone value is updated in the reverse direction from the sink to the source. B-ANT ensures source node regarding energy consumption, packet loss rate, congestion level, and hop counts. In this way, multiple alternate congestion-free paths are constructed by F-ANT and confirm about a successful path creation by B-ANT to the source node.

3.3.1. Forward Ant (F-ANT). The number of F-ANT is generated to construct congestion-free multiple alternate paths between the sources to the sink shown in Algorithm 1. When ant moves from one location to another, it drops pheromone on that path. The density of pheromone is higher on a most favorable path. F-ANT carries information like source ID, sink ID, F-ANT ID, hop count (h_i), energy consumption $E(\text{link})$, congestion level L_C, minimum energy $E_{\min}$, and time to live TTL. Every node in network maintains pheromone table and routing table.

In F-ANT, $P_{ij}^{(t)}$ is the probability of selecting next hop while moving from the source to the sink.

$$P_{ij}^{(t)} = \frac{\left(A_{ij}\right)^x * \left(B_{ij}\right)^y}{\sum_{i,j=1}^{n}\left(A_{ij}\right)^x * \left(B_{ij}\right)^y}. \tag{16}$$

Therefore, $x, y > = 0$, and all j belongs to nodes of the network.

A_{ij} is the pheromone density of link (i, j), and B_{ij} is the heuristic value of link or the attractive coefficient at

1. If (TTL! = 0)
2. If (F-ANT is at sink node)
3. Initialize B-ANT
4. Else
5. If Random(y) < 0.001 then
6. Initialize F-ANT
7. Go to next hop randomly
8. Else
9. Nodes as current and next node are updated here
10. If node already visited
11. Go to other next neighbor node
12. Update heuristic value by step 11
13. END

Algorithm 1: Pseudo-code for F-ANT.

time "t.". M^k is the memory to store information about path (visited nodes) which is used to compute path length in terms of hop count. Here, x and y are controlling parameters for pheromone and heuristic values. The pheromone value is updated with updating visiting nodes and expressed in equation (15).

$$A_{ij}(t, t+1) = (1 - E_v)A_{ij}(t) + E_v R_{ij}(t, t+1), \tag{17}$$

$$R_{ij}(t, t+1) = \frac{a * L_i * L_j}{\left(d_{ij}\right)^2}, \tag{18}$$

$$W_{ij} = \frac{(L_i)^p}{\left(d_{ij}\right)^q}. \tag{19}$$

E_v is the evaporation factor, and it is decremented with time. L_i and L_j are loads on nodes i and j, and "a" is the coefficient. R_{ij} stores load information from source to destination. Here, W_{ij} denotes ratio of load to distance where d_{ij} is the distance between i to j, and p and q are weights or controlling parameters. B_{ij} is the local heuristic value for different paths which incorporates in measuring congestion degree and works as an important parameter to calculate packet loss probability on different paths.

3.3.2. Estimation of Packet Loss Probability. This section explains the applicability of the packet loss probability in the congestion avoidance that is calculated from packet entropy.

As the data rate on different routes is uncertain which can be the cause of congestion in WSN, the packet loss rate is also uncertain here. The uncertainty of information is measured through entropy which was introduced under information theory concept. The information theory was first introduced for statistical thermodynamics. In communication networks, entropy is widely used to measure the abnormality degree of an event which ultimately helps to improve performance metrics of the network such as throughput and energy efficiency. Here, Shannon entropy [40] is used which is a quantitative measure of uncertainty in a data set. Shannon's work is extended by Jaynes named maximum entropy principle

which has the inherent property to optimize entropy measure when incomplete information is provided in moment constraint form.

Every node in a WSN is designed with a finite buffer space to store the incoming data packets. To evaluate the packet loss rate, the buffer entropy is calculated. The maximum entropy principle is used when there is absence of information regarding mean arrival and service rates. This yields Lagrange's loss formula for providing the express for the state probability distribution of loss system. This maximum entropy principle framework is used to study queue behavior for packet loss rate which considers arrival rate and service rate under normalization constraints, moment constraint, and utilization constraint.

While data travelled from one node to another, the value of entropy changes and rate of data arrival in buffer is also different which changes buffer entropy too. The heuristic data observes data packet arrival rate in buffer. Assume a single server queue with finite buffer of N size. Probability distribution of arrival rate of packets is P_{λ_i} or it is a distribution of the queue with size λ_i. Entropy is defined here as the average of information that is received. Therefore, information quantity (IQ) of the data packets is shown in equation (20).

$$\mathrm{IQ} = -\log_2 \frac{1}{P_\lambda}. \tag{20}$$

Shannon entropy [40] is used here, that is, a quantitative measure of uncertainty. The uncertainty arises in probabilistic as well as in deterministic phenomenon where the outcome is about possibility of some specific outcome. Maximum entropy finds the maximum packet loss probability which helps in selecting an optimum path. The path having minimum packet loss probability will be chosen as an optimal path. Packet entropy is average of information quantity. $H(P)$ is a function defined by Shannon to measure uncertainty where n is finite number of data packets transmitted between sources to sink [41].

$$H(P) = -\sum_{i=1}^{N} P_{\lambda_i} \left(\log_2 P_{\lambda_i}\right). \tag{21}$$

The cases are as follows:

(1) $H(P) = 0$, if the probability of arrival packets is either 0 or 1. This is also known as minimum entropy

(2) $H(P) = 1$, if the probability of arrival packets is ½. This is also known as maximum entropy

Moment constraints to deduce maximum uncertainty for finite buffer queuing system are defined in equation (22).

$$\sum_{i=1}^{N} \left(P_{\lambda_i}\right) {\lambda_i}^k = A_k, \text{ where } k = 1, 2, 3, 4 \cdots. \tag{22}$$

Utilization U defines queue utilization function for nonempty queue is defined in equation (23)

$$\sum_{i=1}^{N} \left(h(\lambda_i)\left(P_{\lambda_i}\right)\right) = U = 1 - \left(P_{\lambda_0}\right). \tag{23}$$

The queuing system also includes the empty system as problem of state space, where (P_{λ_0}) is a probability of zero job in system and $h(.)$ is a function defined as 0 when n is equal to 0 and 1. When n is not equal to 0 where natural probability constraint or normalization constraint is shown in equation (24),

$$\sum_{i=0}^{n} \left(P_{\lambda_i}\right) = 1. \tag{24}$$

Maximizing Shannon entropy is subject to moment constraint in equations (22), (23), and (24). Lagrange multiplier is used to determine maxima or minima of a function. It is a weighted sum of objective and constraint function. α, β, and γ are Lagrange multiplier associated with moment constraint, utilization, and standard probability constraint.

$$\begin{aligned} &L\left(P_{\lambda_i}, \alpha, \beta, \gamma\right) \\ &\quad = H(\mathrm{P}) - \alpha\left(1 - \sum_{i=0}^{n} \left(P_{\lambda_i}\right)\right) \\ &\qquad + \beta\left(U - \sum_{i=1}^{n} \left(h(\lambda_i)\left(P_{\lambda_i}\right) + \gamma(A_k - \sum_{i=1}^{n} \left(P_{\lambda_i}\right) {\lambda_i}^k\right), \end{aligned} \tag{25}$$

$$\begin{aligned} &L\left(P_{\lambda_i}, \alpha, \beta, \gamma\right) \\ &\quad = -\sum_{i=1}^{N} P_{\lambda_i}\left(\log_2 P_{\lambda_i}\right) - \alpha\left(1 - \sum_{i=0}^{n} \left(P_{\lambda_i}\right)\right) \\ &\qquad + \beta\left(U - \sum_{i=1}^{n} \left(h(\lambda_i)\left(P_{\lambda_i}\right) + \gamma\,(A_k - \sum_{i=1}^{n} \left(P_{\lambda_i}\right) {\lambda_i}^k\right). \end{aligned} \tag{26}$$

Differentiate the Lagrange function with respect to P_{λ_i}, and maximize Lagrange function, that is, $\partial L / \partial P_{\lambda_i} = 0$, as given in [43, 44], and the value of P_{λ_i} is

$$P_{\lambda_i} = e^{\alpha - \beta.h(\lambda_i) - \gamma.{\lambda_i}^k}. \tag{27}$$

Now, by substituting the value of P_{λ_i} in differential constraint functions (22) and (23),

$${\lambda_i}^k \sum_{k=1}^{m} e^{\alpha - \beta.h(\lambda_i) - \gamma.{\lambda_i}^k} = A_k, \tag{28}$$

$$h(\lambda_i) \sum_{k=1}^{m} e^{\alpha - \beta.h(\lambda_i) - \gamma.{\lambda_i}^k} = U, \tag{29}$$

1. While (B-ANT not at source node)
2. Move to reverse direction to select next link between i and j
3. Pheromone value is updated
4. Update $P_L(N)$, $E(\text{link})$, $h(\text{link})$, L_{C_i}
5. End while

Algorithm 2: Pseudo-code for F-ANT.

$$\sum_{k=1}^{m} e^{\alpha-\beta.h(\lambda_i)-\gamma.\lambda_i^k} = 1. \tag{30}$$

Now, for $k = 1 \cdots .m$, solve equation (30) to obtain value of e^{α} as shown in equation (31).

$$e^{\alpha} = e^{\beta.h(\lambda_i)+\gamma.\lambda_i^k}. \tag{31}$$

The packet loss probability $P_L(N)$ [43] for fully occupied finite buffer using maximum entropy probability distribution of system size is computed as

$$P_L(N) = P_{\lambda_i} = \frac{\left(1 - P_{\lambda_0}\right)\lambda_i^{-\gamma}}{\zeta(\gamma)}. \tag{32}$$

Here, $\zeta(\gamma)$ is Riemann zeta function widely used in number theory when distribution is related to prime numbers. As $i \geq 1$ is an infinite series shown as in equation (33),

$$\zeta(\gamma) = \sum_{i=1}^{N} \lambda_i^{-\gamma}, \gamma > 1. \tag{33}$$

3.3.3. Backward Ant (B-ANT). The B-ANT follows backward path from sink to source node identified by F-ANT. The pheromone value is updated with every move in communication link shown in Algorithm 2. It also contains source ID, sink ID, B-ANT ID, energy consumption, hop counts, and congestion level. $A_{ij}(t, t+1)$ is updated pheromone value of link. The computation function of path for packet loss rate, total link energy consumption, hop counts of current path, and congestion level of link are $f_1(t)$, $f_2(t)$, $f_3(t)$, and $f_4(t)$, respectively, along with e, l, m, and n weight parameters. Here, $a1$ and $a2$ are used as controlling parameters to monitor pheromone updating. The ant moves to the next node in backward direction and updates the pheromone value. It also recalculates and updates the mentioned parameters, in order to make it capable for optimal decision.

$$A_{ij}(t, t+1) = a1 * F(t+1), \tag{34}$$

$$F(t+1) = a2\ \frac{F_1(t)^e}{F_2(t)^l * F_3(t)^m * F_4(t)^n}, \tag{35}$$

$$f_1(t) = \min\left(P_L(N)\right), \tag{36}$$

$$f_2(t) = \sum E(\text{link}), \tag{37}$$

$$f_3 = \sum_{i,j \in h(\text{link})} (i, j), \tag{38}$$

$$f_4 = \sum_{i=1}^{n} L_{C_i}. \tag{39}$$

3.4. Select Optimal Path Using Huffman Coding. After ACO identified multiple alternate congestion-free path, Huffman coding selects optimal path. For this, a tree of N nodes is generated with nodes having low congestion level near to root node and high congestion level near to leaf node. The root node act like sink node. The objective is to choose a path where congestion is less or no congestion near sink node. Such path is considered as an optimal one. Well, all the parameters play important role in selecting optimal path. But congestion is our major concern, so Huffman coding opts an optimal path having minimum congestion level.

Multiple paths can be represented by Q_i, and $h(\text{link})$ is the path length with W_i a weight. So,

$$Q_i = \sum_{i=1}^{N} W_i h_i(\text{link}). \tag{40}$$

Congestion level-oriented optimal path evaluation function g_i can be represented as

$$g_i = \left\{ \frac{L_{C_{\text{Avg}}}}{L_{C_i}} * P_L(N) \right\}^{\tau} + \left(L_{C_{\min}}\right)^{\omega}, \tag{41}$$

$$\text{Opt}(\text{path}) = \min_{i \in N}(g_i). \tag{42}$$

Here, opt(path) is optimal path chooses after Huffman coding. $P_L(N)$ is the packet loss rate and $L_{C_{\min}}$ is the minimum congestion on link. τ and ω are the controlling parameters for hop count and congestion level.

3.5. Huffman Coding for Congestion Control. This section defines the working of Huffman coding for selection of optimal path among multiple alternate paths identified by ACO. It focuses on packet loss rate on different paths while selecting optimal path. The working of Huffman coding (HC) is elaborated in Figure 2, where a, b, c, d, e, and f are the different nodes of a network with different packet loss rate 6, 10, 13, 14, 17, and 46 as shown in Figure 2(a). Huffman tree is also constructed to make better understanding of how to opt an optimal path in WSN during congestion. This Huffman coding approach is aimed at arranging the nodes in an optimal way of their usage. Create a min heap of six nodes, and each node represents root of a tree with single node.

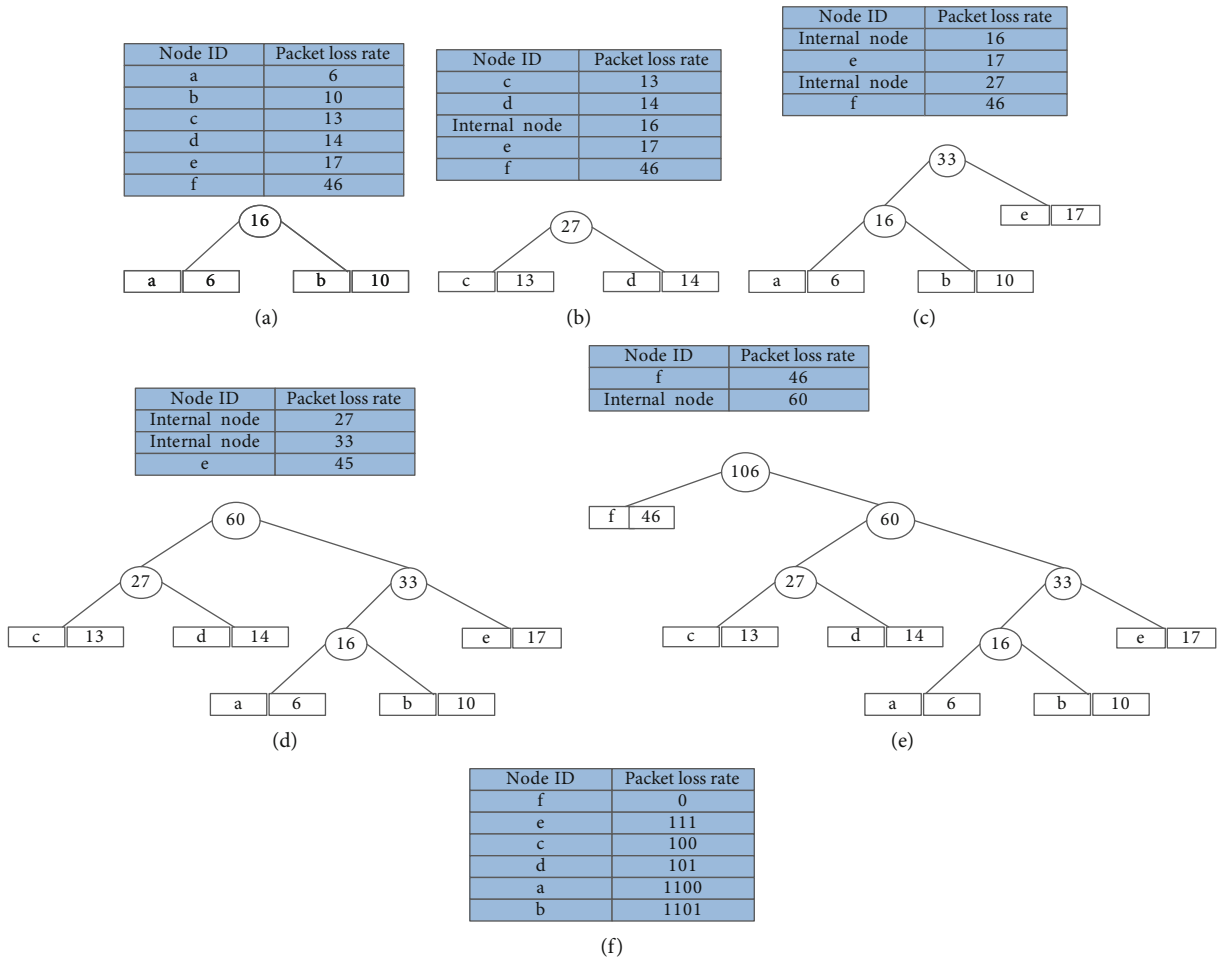


Figure 2: Huffman coding for optimal path selection.

Extract two nodes with minimum packet loss rate from min heap and add an internal node with packet loss rate $6 + 10 = 16$, shown in Figure 2(a). Now min heap has five nodes, where four nodes are roots of tree with three elements shown in Figure 2(b). Extract two nodes with minimum packet loss rate from heap. Add a new internal node with packet loss rate $13 + 14 = 27$. Now, min heap has four nodes, where two nodes are roots of tree with single element each and two heap nodes are roots of tree with more than one node shown in Figure 2(c). Extract two minimum nodes with packet loss rate again. Add internal node with packet loss rate $16 + 17 = 33$. Now, min heap has three nodes. Extract two nodes with minimum packet loss rate. Add a new internal node with packet loss rate $27 + 33 = 60$ shown in Figure 2(d). Now, min heap has two nodes. Extract two nodes with minimum packet loss rate. Add a new internal node with packet loss rate $46 + 60 = 106$ shown in Figure 2(e). Now, min heap has only one node. Since the heap has only one node, the algorithm ends here. Start traversing tree from root node. For all left child, write 0, and for all right child, write 1.

3.6. ECA-HA Flow Chart. The complete operations of proposed technique are presented by a flow chart in Figure 3. The three major portions of this flow chart clearly define the operations of proposed technique. During communication, if λ_A is greater than λ_D then check the congestion level by computing buffer occupancy level, and if virtual buffer occupancy is greater than maximum buffer limit then it is a link layer congestion and it is a time to reroute the data on an alternate route; otherwise, the network is safe and communication goes smooth. Then, ACO operation performs here, and ACO identifies multiple alternate congestion-free paths. It identifies packet loss probability by calculating packet entropy which helps in identifying an optimal path. Huffman coding selects an optimal path among multiple congestion-free alternate paths. The pseudo-code for ACO clearly defined all the steps involved.

3.7. Time Complexity. The time complexity of the proposed ECA-HA algorithm depends upon the complexity of Huffman coding and ant colony optimization. For the "n" number of sensor nodes, the Huffman coding takes $O(n \log n)$ to control the congestion in the path of the network [10], whereas ant colony optimization selects the best route under duration $O(1/E_v n^2 \log n)$, with E_v defines the evaporation rate of pheromone [35]. Thus, overall time complexity of the proposed ECA-HA is the summation of $O(n \log n) + O(1/\rho n^2 \log n = O(1/\rho n^2 \log n))$.

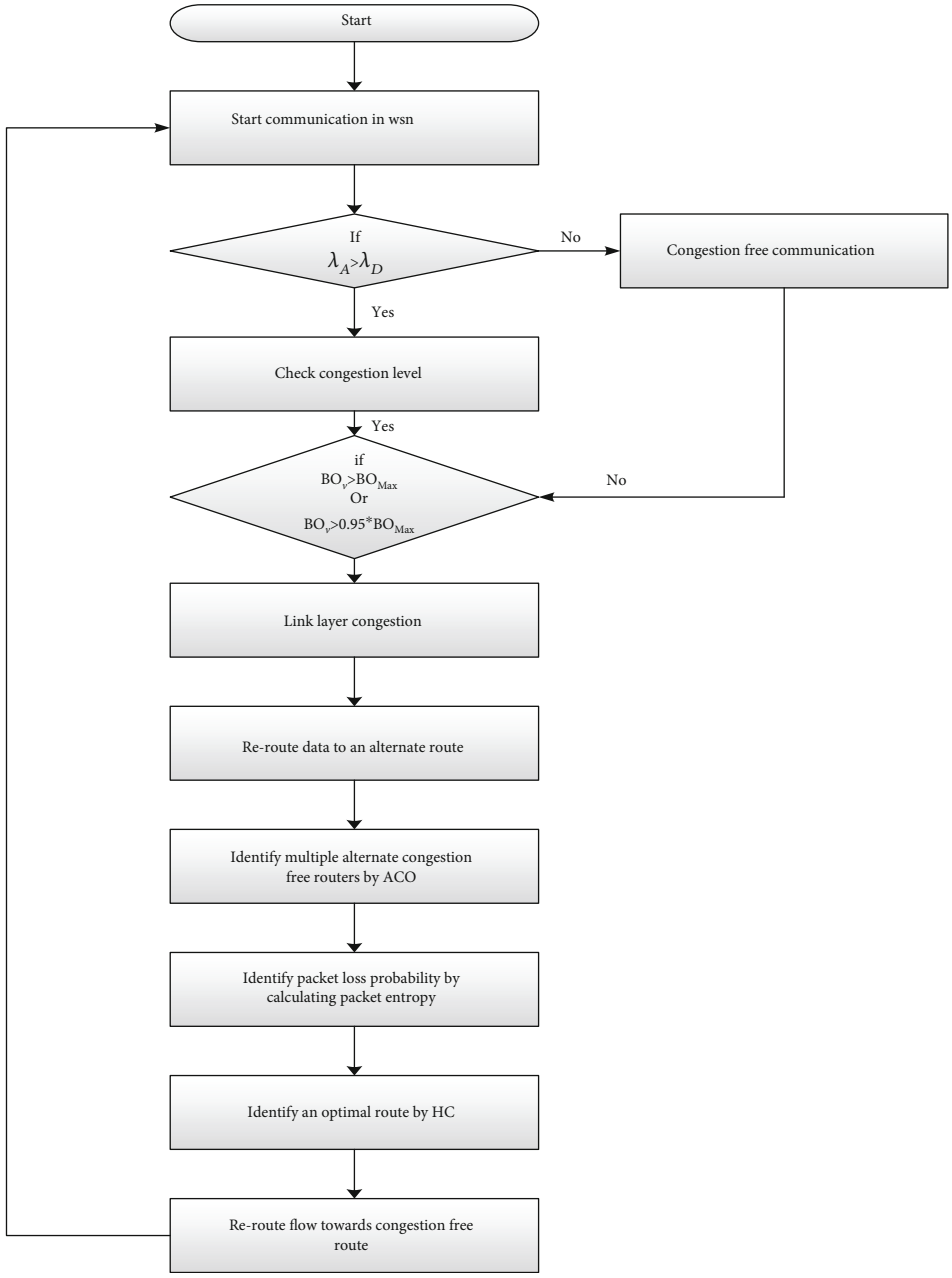


Figure 3: Flow chart of ECA-HA

4. Simulation and Result Discussion

In this section, we performed the extensive simulation using simulator MATLAB2017b of the proposed algorithm. The sensor nodes are in range from 20 to 100 randomly distributed in the 2D $(100 \times 100\, m^2)$ space of the wireless network. Base station is placed inside (40, 72) in the network which has capability of data gathering and query processing. The initial energy of node is $E_i = 0.5$ joules. The pheromone released by ant is evaporation once in every 4 seconds, whereas the amount of search ant is set to be 6 and their maximum lifetime is 20. The maximum number of iterations or round is 1000. The remaining list of parameters used for simulations is shown in Table 2. The proposed algorithm is compared with the nature-inspired algorithm ACSRO [19] and traditional congestion avoidance algorithm CODA [13]. The performance of the proposed algorithm is evaluated against the average throughput, average hop-by-hop delay, and packet delivery ratio and node death percentage. These parameters define the rate of successful delivery of data over the total bandwidth, time delay in forwarding the packet from current hop to next forward hop, and ratio of the number of packets successfully received by receiver to the number of packets send from source. Further, comparative analysis of the ECA-HA with state-of-art algorithm is also evaluated.

4.1. Average Energy Consumption over Rounds. A comparison of energy consumption of the proposed ECA-HA algorithm

Table 2: Simulation parameters.

Parameter	Value
Network area	$100 \times 100\text{m}^2$
Number of nodes (N)	20-100
E_{amp}	0.0013 pJ/bit/m^4
E_{elec}	50 nJ/bit
E_i	0.5 J
D_{pkt}	500 bits
d_{ij}	[5, 20]
Control packet size	100 bits
Data transmission rate (λ_D)	512 Kb/s
Data arrival rate (λ_A)	700 Kb/s
Communication range	20 m
Max packet in buffer (BO_{Max})	40
Congestion threshold of buffer (thr_{max})	33
Pheromone density of link (A_{ij})	20
Evaporation factor (E_v)	0.65
Antenna type	Omni antenna

with state-of-art algorithms over number of rounds is shown in Figure 4. It can be observed from the results, for the initial phase of rounds at 400, the energy consumption is 0.12 joules, 0.2 joules, and 0.51 joules for ECA-HA, ACSRO [19] (nature-inspired algorithm), and CODA, respectively. And further increase in the rounds up to 1000 rounds the ECA-HA only consumes 1.4 joules, and ACSRO consumes 1.6 joules and worst performance shown by CODA that consumes around 1.9 joules. This is due to the fact that the proposed algorithm chose the next hop routing path based upon higher residual energy and less congested routing density based on Huffman coding and ant colony optimization, whereas ACSRO uses adaptive cuckoo search rate adjustment for the selection of next hop that performs better than CODA algorithm. It is also noted down that the worst performance is shown by without nature-inspired algorithm, because it does not have any selection policy for the next hop routing.

4.2. Average Residual Energy of Nodes over Rounds. A comparison of average residual energy between ECA-HA and the state-of-art algorithm is presented in Figure 5. It is clearly observed from the result as the number of rounds increases, residual energy decreases for all the three algorithms, whereas ECA-HA and ACSRO having residual energy 0.4 and 0.2 joules, respectively, for 1000 rounds. Further, it is noticeable that CODA algorithm exhausts all the available energy and in turn zero residual energy. This is because of proposed algorithm ECA-HA selects the next hop for the routing purpose based on ant colony optimization, and priority is given to nodes having higher residual energy using Huffman coding, whereas ACSRO selects the path for packet delivery based upon residual energy and minimum hop distance but does not include the congestion density on the next available node rather the proposed ECA-HA considered all the mentioned parameters. It is also worthy to note down

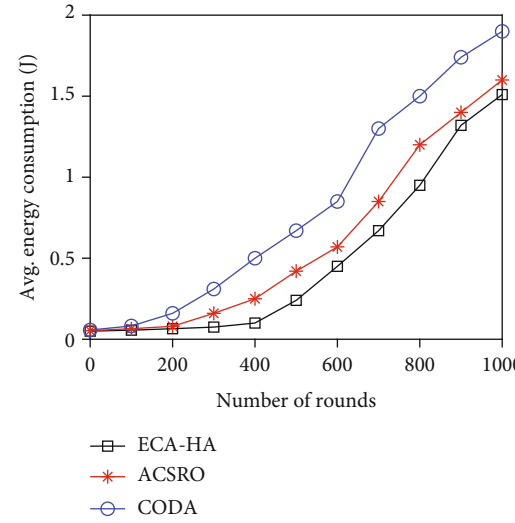


Figure 4: Average energy consumption.

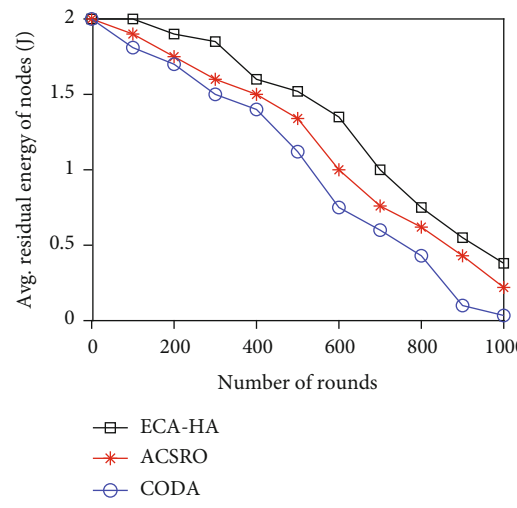


Figure 5: Average residual energy.

that CODA algorithm shows poorest performance due to selection of the next route depends on minimum distance and does not consider residual energy of the next hop that in turns increases the overall energy consumption, and lifetime of the network is decreased.

4.3. Average Throughput over Source Data Rate. A comparison of average throughput of the proposed algorithm using 20 and 100 numbers of nodes in the network is shown in Figures 6(a) and 6(b), respectively. It is clearly observed from the result of Figure 6(a) that proposed algorithm ECA-HA improves throughput about 18% comparing to ACSRO and 21% comparing to CODA algorithm, respectively, whereas average throughput in Figure 6(b) shows that all three state-of-art algorithms in the beginning (source data rate = 10) provide nearly the same 50 packets per second delivery rate. With further increase in the source data rate, our proposed algorithm ECA-HA increases the average throughput rapidly and reaches about 100 packets per second comparing

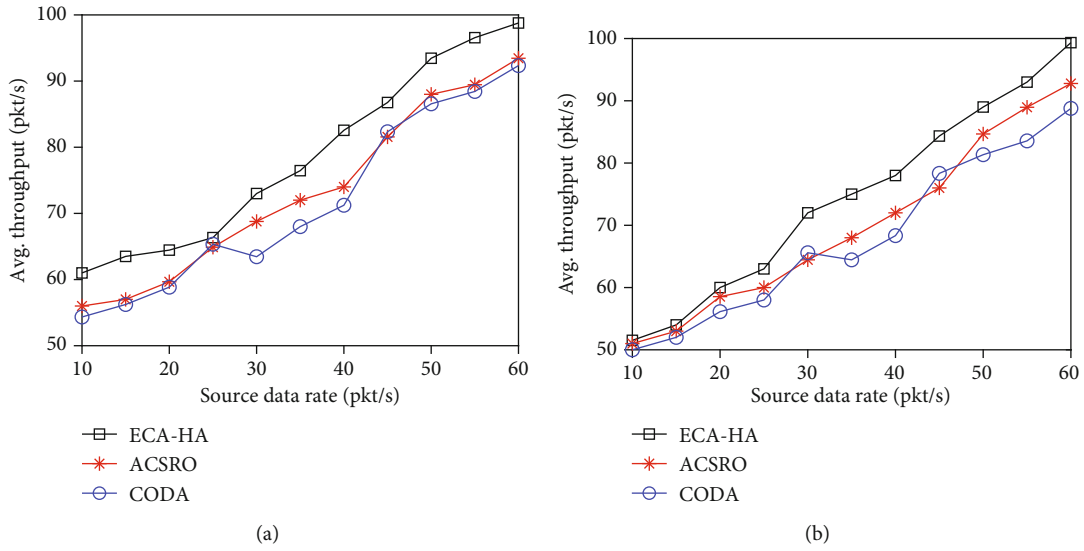


Figure 6: (a) Avg. throughput per 20 nodes. (b) Avg. throughput per 100 nodes.

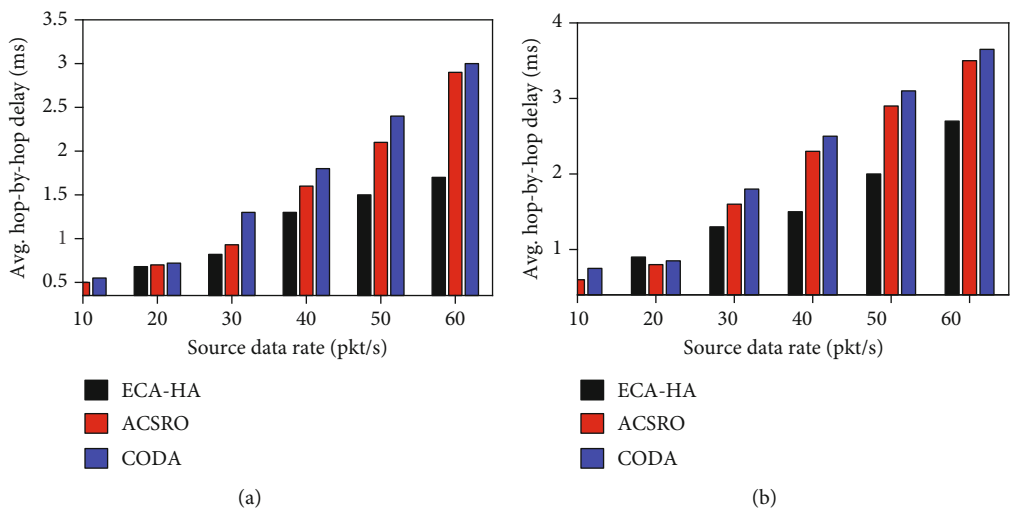


Figure 7: (a) Avg. hop delay per 20 nodes. (b) Avg. hop delay per 100 nodes.

to 91 packets per second and 89 packets per second for ACSRO and CODA, respectively. This is due to the fact that data transmission level congestion is controlled through the effective selection of next hop based on ant colony approach that provides alternate path having lower number of packets to forward. It can be also observed that average throughput in the network using CODA algorithm shows much variation than other state-of-art algorithms. This can be attributed to the reason CODA algorithm consumes more energy to forward the packets and rarely selects optimal next hop for the forwarding. Thus, overall it is clearly proved that ECA-HA is the best suitable algorithm for data forwarding in the wireless sensor network composed to 20 or more than 100 nodes environment. This can address the problem of congestion, missing packets in wireless sensor network, and upgradation of network throughput.

4.4. Average Hop-by-Hop Delay over Source Data Rate. A comparison result of average hop-by-hop delay between proposed algorithm ECA-HA and state-of-art algorithm is shown in Figures 7(a) and 7(b) in the scenario nodes 20 and 100, respectively. This metric plays very important to show the interpath interference control mechanism, overhead produce by the exchange of control packets, and link-layer transmission delay. It can be clearly observed from the result of Figure 7(a) that proposed algorithm ECA-HA has less delay about 33% and 38% compared to ACSRO and CODA, respectively, for the source data rate 40 packets per seconds and onwards, whereas in Figure 7(b), the ECA-HA delivers the packets to next hop faster about 39% comparing to ACSRO and 43% comparing to CODA. The reason behind is that proposed algorithm ECA-HA exchanges minimum number of control packets for the selection of routing path.

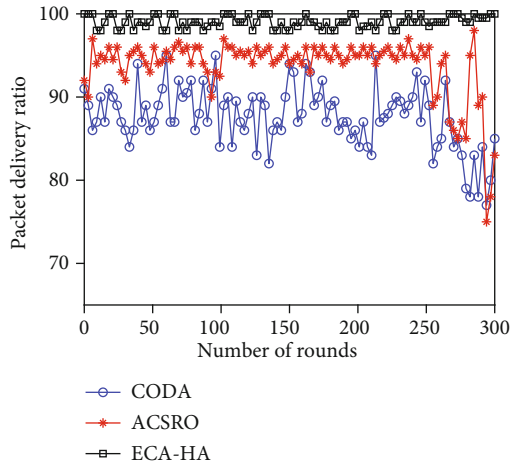


Figure 8: Packet delivery ratio.

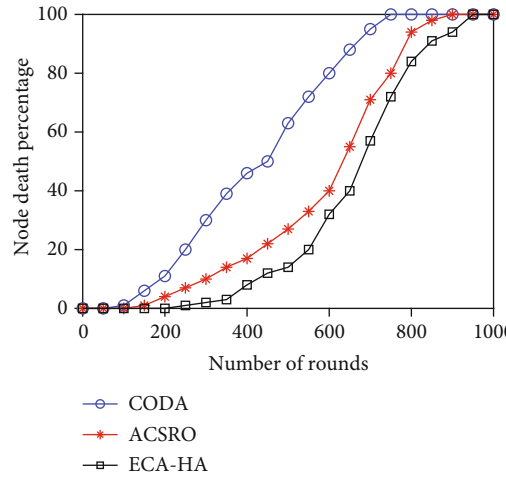


Figure 9: Node death percentage.

This can be attributing to the reasons of Huffman coding uses priority-based heap for the selection for next hop based on distance between node and base station. Further, data routing path is balanced using ant colony approach that helps in minimizing the hop-by-hop delay, whereas ACSRO and CODA are not able to optimize the number of control packet exchanges that inherently increases the interference too. This can be attributing to increase the congestion in the network in turn enhance the overall hop-by-hop delay.

4.5. Packet Delivery Ratio over Number of Rounds. A comparison of packet delivery ratio over number of rounds between proposed algorithm ECA-HA and the state-of-art algorithm is presented in Figure 8. Packet delivery ratio turns out to be a throughput measurement parameter; the higher the packet delivery ratio, the higher the throughput in the network. It is clearly observed from the result on the increasing number of rounds the packet delivery ratio is higher about 95%-98% for the proposed algorithm. This is due to the fact that proposed algorithm uses Huffman coding for the selection of routing path to deliver the packet with less congested path, whereas nature-inspired ACSRO algorithm performs better than traditional congestion avoidance algorithm CODA with 35% higher packet delivery ratio. This is because of the ACSRO selects the next hop for the routing purpose based on adaptive cuckoo search rate adjustment for optimized congestion avoidance. Finally, it is observed that proposed algorithm ECA-HA performs better than by 8% and 48% higher packet delivery with respect to ACSRO and CODA, respectively.

4.6. Node Death Percentage over Number of Rounds. A comparison of node death percentage over number of rounds between proposed algorithm ECA-HA and the state-of-art algorithm is presented in Figure 9. It is clearly observed from the result as the number of rounds increases, the death percentage of nodes also increases for all the three algorithms and all the nodes are dead at maximum 950 rounds. Further, it is noticeable that proposed ECA-HA-based congestion control routing algorithm has 20% and 40% lower node death percentage ratio rather than ACSRO and CODA, respectively, at 600 rounds. It is also worthy to note that proposed algorithm ECA-HA runs about 960 rounds, whereas ACSRO and CODA run only for 820 and 640 rounds. This is due to the fact ECA-HA chose the routing path with lower congestion rate and having more residual energy compare to ACSRO and CODA. And the without nature-inspired-based algorithm CODA selects any node for data forwarding with consideration of hop distance and residual energy, which leads to worst performance, and lifetime of the network is decreased.

5. Conclusion and Future Scope

This paper proposed a congestion control algorithm ECA-HA for wireless sensor networks that can be applied on most of the application like health care system, agriculture, and monitoring. The proposed algorithm uses ant colony approach using two algorithms for backward ant and forward ant algorithms to find more than one path for data routing. Further, Huffman coding was applied on the top to select one of the less congested paths as optimal data routing path from the sample space given by ant colony approach. The extensive performance results of the proposed algorithm outperform in terms of various parametric than state-of-art algorithms. Thus, the proposed algorithm forwards the packets using optimal data routing path in the current round that decreases the hop-by-hop delay, improves the packet delivery ratio, or decreases the node death percentage which ultimately improves the throughput of the network and lifetime, respectively. The only shortcoming of the proposed algorithm is that the network is increasing at a very fast rate but this approach is suitable for small search space. Therefore, in future, we will look forward for large search space. In the future, considering Internet of Things environment, we will include the more parameter such as dynamic traffic load or breakage in the link to find more optimal data routing path based on reinforcement learning approach.

Data Availability

The experimental data and associated settings will be made available to researchers and practitioners on individual request to the corresponding author, with the restrictions that it will solely be used for further research in literature progress, as the associated research data is being further utilized for development research by the team.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the SC&SS, Jawaharlal Nehru University, New Delhi.

References

[1] S. Yadav, "A study on congestion control mechanisms in wireless sensor networks," *Journal of Advanced Research in Dynamical & Control Systems*, vol. 10, pp. 842–850, 2015.

[2] F. Ullah, A. H. Abdullah, O. Kaiwartya, S. Kumar, and M. M. Arshad, "Medium access control (MAC) for wireless body area network (WBAN): superframe structure, multiple access technique, taxonomy, and challenges," *Human-centric Computing and Information Sciences*, vol. 7, no. 34, pp. 1–39, 2017.

[3] S. L. Yadav and R. L. Ujjwal, "Mitigating congestion in wireless sensor networks through clustering and queue assistance: a survey," *Journal of Intelligent Manufacturing*, pp. 1–16, 2020.

[4] S. L. Yadav and R. L. Ujjwal, "Sensor data fusion and clustering: a congestion detection and avoidance approach in wireless sensor networks," *Journal of Information and Optimization Sciences*, vol. 41, no. 7, pp. 1673–1688, 2020.

[5] A. Khasawneh, M. S. B. A. Latiff, O. Kaiwartya, and H. Chizari, "Next forwarding node selection in underwater wireless sensor networks (UWSNs): techniques and challenges," *Information*, vol. 8, no. 1, p. 3, 2017.

[6] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud et al., "Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4049–4062, 2018.

[7] O. Kaiwartya, A. H. Abdullah, Y. Cao et al., "T-MQM: testbed-based multi-metric quality measurement of sensor deployment for precision agriculture—a case study," *IEEE Sensors Journal*, vol. 16, no. 23, pp. 8649–8664, 2016.

[8] P. Kumar Kashyap, S. Kumar, U. Dohare, V. Kumar, and R. Kharel, "Green computing in sensors-enabled internet of things: neuro fuzzy logic-based load balancing," *Electronics*, vol. 8, no. 4, p. 384, 2019.

[9] A. Khatri, S. Kumar, O. Kaiwartya, N. Aslam, N. Meena, and A. H. Abdullah, "Towards green computing in wireless sensor networks: controlled mobility–aided balanced tree approach," *International Journal of Communication Systems*, vol. 31, no. 7, article e3463, 2018.

[10] S. Kumar, O. Kaiwartya, and A. H. Abdullah, "Green computing for wireless sensor networks: optimization and Huffman coding approach," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 592–609, 2017.

[11] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and A. H. Gandomi, "Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network," *IEEE Sensors Journal*, 2021.

[12] L. Farhan, O. Kaiwartya, L. Alzubaidi, W. Gheth, E. Dimla, and R. Kharel, "Toward interference aware IoT framework: energy and geo-location-based-modeling," *IEEE Access*, vol. 7, pp. 56617–56630, 2019.

[13] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 266–279, New York, NY, USA, Nov 2003.

[14] S. Brahma, M. Chatterjee, and K. Kwiat, "Congestion control and fairness in wireless sensor networks," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 413–418, Mannheim, Germany, 2010, March.

[15] K. Kashyap and S. Kumar, "Load-balanced distributed intra-clustering algorithm," in *2015 Annual IEEE India Conference (INDICON)*, pp. 1–6, New Delhi, India, 2015.

[16] C. Wang, K. Sohraby, V. Lawrence, B. Li, and Y. Hu, "Priority-based congestion control in wireless sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, pp. 1–8, Taichung, Taiwan, June 2006.

[17] M. Z. Ahmad and D. Turgut, "Congestion avoidance and fairness in wireless sensor networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pp. 1–6, New Orleans, LA, USA, 2008, November.

[18] C. Sergiou and V. Vassiliou, "DAlPaS: a performance aware congestion control algorithm in wireless sensor networks," in *18th International Conference on Telecommunications*, pp. 167–173, Ayia Napa, Cyprus, May 2011.

[19] V. Narawade and U. D. Kolekar, "ACSRO: adaptive cuckoo search based rate adjustment for optimized congestion avoidance and control in wireless sensor networks," *Alexandria Engineering Journal*, vol. 57, no. 1, pp. 131–145, 2018.

[20] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 205–210, San Antonio, TX, USA, 2009.

[21] N. Siddique and H. Adeli, "Nature inspired computing: an overview and some future directions," *Cognitive Computation*, vol. 7, no. 6, pp. 706–714, 2015.

[22] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "Congestion control in wireless sensor networks based on bird flocking behavior," *Computer Networks*, vol. 57, no. 5, pp. 1167–1191, 2013.

[23] M. A. Kafi, J. Ben-Othman, A. Ouadjaout, M. Bagaa, and N. Badache, "REFIACC: reliable, efficient, fair and interference-aware congestion control protocol for wireless sensor networks," *Computer Communications*, vol. 101, no. C, pp. 1–11, 2017.

[24] P. K. Kashyap, "Genetic-fuzzy based load balanced protocol for WSNs," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 2, pp. 2088–8708, 2019.

[25] P. K. Kashyap, S. Kumar, and A. Jaiswal, "Deep learning based offloading scheme for IoT networks towards green computing," in *IEEE International Conference on Industrial Internet (ICII)*, pp. 22–27, Orlando, FL, USA, Nov 2019.

[26] K. Singh, K. Singh, L. H. Son, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Computer Networks*, vol. 138, pp. 90–107, 2018.

[27] A. Jaiswal, S. Kumar, O. Kaiwartya et al., "Quantum learning enabled green communication for next generation wireless systems," *IEEE Transactions on Green Communications and Networking*, 2021.

[28] J. S. Lee and C. L. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1095–1103, 2017.

[29] H. El Alami and A. Najid, "MS-routing-G i: routing technique to minimise energy consumption and packet loss in WSNs with mobile sink," *IET Networks*, vol. 7, no. 6, pp. 422–428, 2018.

[30] H. El Alami and A. Najid, "ECH: an enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks," *IEEE Access*, vol. 7, pp. 107142–107153, 2019.

[31] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "A heuristic path search for congestion control in WSN," in *Industry Interactive Innovations in Science, Engineering and Technology*, Springer, Singapore, 2018.

[32] R. Logambigai, S. Ganapathy, and A. Kannan, "2018. Energy–efficient grid–based routing algorithm using intelligent fuzzy rules for wireless sensor networks," *Computers & Electrical Engineering*, vol. 68, pp. 62–75, 2018.

[33] G. Sangeetha, M. Vijayalakshmi, S. Ganapathy, and A. Kannan, "An improved congestion-aware routing mechanism in sensor networks using fuzzy rule sets," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 890–904, 2020.

[34] M. Selvi, S. V. N. Santhosh Kumar, S. Ganapathy, A. Ayyanar, H. Khanna Nehemiah, and A. Kannan, "An energy efficient clustered gravitational and fuzzy based routing algorithm in WSNs," *Wireless Personal Communications*, vol. 116, no. 1, pp. 61–90, 2021.

[35] M. Mandloi and V. Bhatia, "Congestion control based ant colony optimization algorithm for large MIMO detection," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3662–3669, 2015.

[36] A. A. Gurav and M. J. Nene, "Multiple optimal path identification using ant colony optimisation in wireless sensor network," *International Journal of Wireless & Mobile Networks*, vol. 5, no. 5, pp. 119–128, 2013.

[37] W. Ding, L. Tang, and S. Ji, "Optimizing routing based on congestion control for wireless sensor networks," *Wireless Networks*, vol. 22, no. 3, pp. 915–925, 2016.

[38] C. J. Raman and V. James, "FCC: fast congestion control scheme for wireless sensor networks using hybrid optimal routing algorithm," *Cluster Computing*, vol. 22, no. S5, pp. 12701–12711, 2019.

[39] A. A. Rezaee and F. Pasandideh, "A fuzzy congestion control protocol based on active queue management in wireless sensor networks with medical applications," *Wireless Personal Communications*, vol. 98, no. 1, pp. 815–842, 2018.

[40] J. Zhang, Z. Lin, P. W. Tsai, and L. Xu, "Entropy-driven data aggregation method for energy-efficient wireless sensor networks," *Information Fusion*, vol. 56, pp. 103–113, 2020.

[41] W. Osamy, A. Salim, and A. M. Khedr, "An information entropy based-clustering algorithm for heterogeneous wireless sensor networks," *Wireless Networks*, vol. 26, no. 3, pp. 1869–1886, 2020.

[42] J. N. Kapur and H. K. Kesavan, "Entropy Optimization Principles and Their Applications," in *Entropy and Energy Dissipation in Water Resources*, Springer, Dordrecht, 1992.

[43] A. K. Singh and Karmeshu, "Power law behavior of queue size: maximum entropy principle with shifted geometric mean constraint," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1335–1338, 2014.

[44] S. Sharma and S. Kumar, "Generation of power law: maximum entropy framework and super statistics," in *Man–Machine Interactions*, Advances in Intelligent Systems and Computing, pp. 45–59, Springer, Champions, 2016.

Hindawi
Journal of Sensors
Volume 2021, Article ID 5558860, 11 pages
https://doi.org/10.1155/2021/5558860

Research Article

An Exhaustive Research on the Application of Intrusion Detection Technology in Computer Network Security in Sensor Networks

Yajing Wang[1], **Juan Ma**[1], **Ashutosh Sharma**[2], **Pradeep Kumar Singh**[3], **Gurjot Singh Gaba**[4], **Mehedi Masud**[5] and **Mohammed Baz**[6]

[1]*Internet of Things Technology Department, Shanxi Vocational &Technical College of Finance & Trade, Taiyuan, 030031 Shanxi, China*
[2]*Institute of Computer Technology and Information Security, Southern Federal University, Russia*
[3]*Department of CSE, ABES Engineering College, Ghaziabad, Uttar Pradesh, India*
[4]*School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara, Punjab 144411, India*
[5]*Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia*
[6]*Department of Computer Engineering, College of Computer and Information Technology, Taif University, PO Box. 11099, Taif 21994, Saudi Arabia*

Correspondence should be addressed to Yajing Wang; yajingwang4@gmail.com and Mehedi Masud; mmasud@tu.edu.sa

Received 17 February 2021; Revised 7 May 2021; Accepted 13 May 2021; Published 29 May 2021

Academic Editor: Omprakash Kaiwartya

Intrusion detection is crucial in computer network security issues; therefore, this work is aimed at maximizing network security protection and its improvement by proposing various preventive techniques. Outlier detection and semisupervised clustering algorithms based on shared nearest neighbors are proposed in this work to address intrusion detection by converting it into a problem of mining outliers using the network behavior dataset. The algorithm uses shared nearest neighbors as similarity, judges whether it is an outlier according to the number of nearest neighbors of a data point, and performs semisupervised clustering on the dataset where outliers are deleted. In the process of semisupervised clustering, vast prior knowledge is added, and the dataset is clustered according to the principle of graph segmentation. The novelty of the proposed algorithm lies in outlier detection while effectively avoiding the dependence on parameters, thus eliminating the influence of outliers on clustering. This article uses real datasets: lypmphography and glass for simulation purposes. The simulation results show that the algorithm proposed in this paper can effectively detect outliers and has a good clustering effect. Furthermore, the experimentation reveals that the outlier detection-based SCA-SNN algorithm has the best practical effect on the dataset without outliers, clearly validating the clustering performance of the outlier detection-based SCA-SNN algorithm. Furthermore, compared to the other state-of-the-art anomaly detection method, it was revealed that the anomaly detection technology based on outlier mining does not require a training process. Thus, they overcome the current anomaly detection problems caused due to incomplete normal patterns in training samples.

1. Introduction

With the widespread advancement in the Internet and online platforms, network security requirements have also become inevitable [1, 2]. Various threats related to computer network security can be seen nowadays, like software bugs and intrusions. These bugs appear due to the large functionality and large size of the software or the operating system. The intruders who do not have access to this data may steal useful private information against the consent of the network users. However, the firewalls are placed in between two or more computers dedicated to isolating these networks based on determining rules or policies. But these firewalls are not enough to be secured from such types of attacks. This is the scenario where intrusion detection systems play a vital role in stopping

the cyber attacks and analyze the security problems at the time of such intrusions so that these situations can be tackled in the future [3–5]. The intrusion detection systems collect the computer network information to track the possibility of attacks or misuses against ethical concerns [6, 7]. There are several types of network data concerns that fall into the category to be protected by intrusion detection, like network traffic data, system status files, and system-level test data [8–10]. There exist various applications of network intrusion detection systems which are depicted in Figure 1.

The network traffic processing application can convert the traffic into various network parameter patterns, helpful in management. The prevention system is liable to detect the threats, and threat classification is done utilizing signature matching that is designated to match the input against the already present pattern. The other applications include threat reporting and anomaly detection that detects the traffic signatures [11, 12].

With the rapid development and application of computer network technology and the increasing number of computer network users, ensuring the security of information on the network has become a key technology of computer networks [13–15]. However, various security mechanisms have been developed to protect computer networks, such as user authorization and authentication, access control, data encryption, and data backup. But the above security mechanisms can no longer meet the current network security needs [16]. Network intrusions and attacks are still not uncommon. Therefore, intrusion detection is one of the key technologies that emerged in information and network security assurance. Introducing intrusion detection technology is equivalent to introducing a closed-loop security strategy [17, 18] into the computer system.

This article addresses intrusion detection by converting it into a problem of mining outliers using the network behavior dataset. A preventive technique for intrusion protection of computer network security is proposed to detect the outliers using the semisupervised clustering algorithms based on shared nearest neighbors. The nearest neighbor similarity criteria are used in this work to judge the outlier according to the number of nearest neighbors of a data point, and on this basis, semisupervised clustering is performed for deleting the outliers. The novelty of the proposed algorithm lies in outlier detection while effectively avoiding the dependence on parameters, thus eliminating the influence of outliers on clustering. This work used the real dataset for simulation and compared it with the other anomaly detection technologies. It was revealed that the anomaly detection technology based on outlier mining does not require a training process. This overcomes the current anomaly detection problems caused due to incomplete normal patterns in training samples. Furthermore, the proposed algorithm effectively detects outliers and provides good clustering outcomes based on the similarity.

The rest of this article is arranged as follows: Section 2 presents the state-of-the-art literature review followed by the research methods depicted in Section 3. Section 4 provides the results and discussion part of the experimental analysis done for the two datasets, followed by the concluding remarks in Section 5.

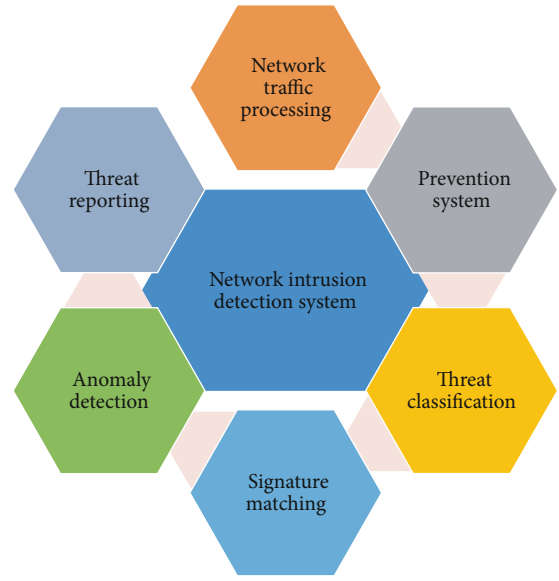


Figure 1: Applications of network intrusion detection system.

2. Literature Review

Domestic research on intrusion detection technology and methods started relatively late, but with the in-depth exploration of universities, scientific research institutes, and enterprises, the development is very rapid, and many new detection theories and results have been produced. The current research on intrusion detection technology mainly covers neural networks, data mining, support vector machines, artificial immunity, etc., involving smart grids, industrial infrastructure, industrial networks, and other related fields [19–22].

Sun et al. proposed an improved method of cascading transmission edges. Using the character interval, the character interval can be used to represent several consecutive characters, which can effectively reduce the number of transmission edges. In addition, the two methods before and after the improvement were compared through comparative experiments. The results show that the number of transmission edges can be reduced to 10% of that before the improvement, thereby increasing the efficiency of deep packet inspection [23].

Haojie et al. analyzed the potential security threats of 5G in-vehicle networks and focused on intrusion detection methods for in-vehicle networks. Four experimental scenarios were selected from potential attacks on the vehicle network, and real car data were collected to compile various attack databases for the first time. In order to find the appropriate method to identify different attacks, four lightweight intrusion detection methods are proposed to identify the abnormal behavior of the vehicle network. In addition, the research carried out a comparison of the detection performance between the four detection methods with the consideration of comprehensive evaluation indicators. The

evaluation results provide the best lightweight detection solution for the vehicle network. This article helps to understand the advantages of test methods in the detection performance of in-vehicle networks. Furthermore, it promotes the application of detection technologies to safety issues in the automotive industry [24].

Zhang et al. took intrusion detection system (IDS) as the research object, established an IDS model based on data mining, obtained experimental results, and drew relevant experimental conclusions. At the same time, it was compared with traditional IDS, and six experiments were carried out. As a result, the detection rate, false-negative rate, and false-positive rate of two different methods in six experiments were obtained. The experiment concludes that the intrusion detection system using data mining has better network protection and security performance, and the detection ability of network vulnerability intrusion is stronger. Thus, this research provides a new way to detect and research network protection security loopholes [25].

Kumar et al. proposed a model in which a set of training examples obtained by using a network analyzer (i.e., Wireshark) can be used to construct an HMM. Since it is not an intrusion detection system, the obtained file trace can be used as a training example to test the HMM model. It also predicts the probability value of each test sequence and indicates whether the sequence is abnormal. This article also shows a numerical example; the example calculates the best observation sequence for the HMM and state sequence probability [26].

The innovation of this paper is that the problem of intrusion detection can be converted into the problem of mining outliers in the network behavior dataset. Compared with other anomaly detection technologies, the anomaly detection technology based on outlier mining does not require a training process, which overcomes the current anomaly detection faced with the problem of high false alarm rate caused by incomplete normal patterns in training samples. This paper describes the outlier mining algorithm based on the similarity.

3. Research Methods

3.1. Classification of Intrusion Detection. Through the research of existing intrusion detection technology methods, intrusion detection technology can be classified from different angles:

(1) According to the source of detection data, there are three categories: host-based intrusion detection technology, network-based intrusion detection technology, and host- and network-based intrusion detection technology. The above three intrusion detection technologies all have their own advantages and disadvantages and can complement each other. However, a complete intrusion detection system must be distributed based on both the host and the network

(2) According to the detection technology: divided into anomaly detection technology and misuse detection technology. Anomaly detection technology can also be called behavior-based intrusion detection technology, which assumes that all intrusions have abnormal characteristics. On the other hand, misuse detection technology, also known as knowledge-based intrusion detection technology, expresses intrusion behavior in attack mode and attack signature

(3) According to the working method: it can be divided into offline detection and online detection. Offline detection: it is a non-real-time system that analyzes audit events after the event and checks for intrusions. Online detection: real-time online detection system, which includes real-time network data packet analysis and real-time host audit analysis

(4) The system network architecture is divided into centralized detection technology, distributed detection technology, and layered detection technology. The analysis result is transmitted to the adjacent upper layer, and the detection system of the higher layer only analyzes the analysis result of the next layer. In addition, the hierarchical detection system makes the system more scalable by analyzing the hierarchical data [27–30].

3.2. Intrusion Detection System and Working Principle. An intrusion detection system refers to the system used to detect various intrusion behaviors. It is an important part of the network security system. By monitoring the operation status of the network and computer system, various attack attempts, attack behaviors, or attack results are found. And then promptly issue an alarm or make a corresponding response to ensure the confidentiality, integrity, and availability of system resources. Intrusion detection systems have been widely used and researched as an important means to resist network intrusion attacks [31, 32]. The basic intrusion detection system for computer network security is depicted in Figure 2.

The intrusion detection system is a typical "snooping device." It does not bridge multiple physical network segments (usually only one listening port). It does not need to forward any traffic, but only needs to passively and silently collect the messages it cares about on the network. Based on the collected messages, the intrusion detection system extracts the corresponding traffic statistical characteristic values. It uses the built-in intrusion knowledge base to perform intelligent analysis and comparison with these traffic characteristics [33, 34]. According to the preset threshold, the message traffic with higher matching coupling will be considered an offense. The intrusion detection system will wake up and alarm or carry out a limited counterattack according to the corresponding configuration. The principle of intrusion detection is shown in Figure 3.

The workflow of an intrusion detection system is roughly divided into the following steps:

(1) Information collection. The first of intrusion detection is information collection, which includes the content of network traffic, the status, and behavior of user connection activities

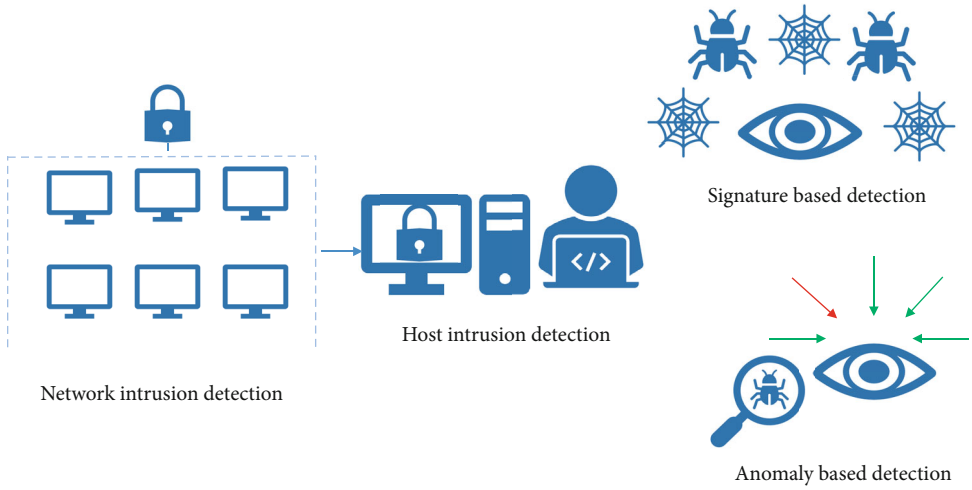


Figure 2: Basic intrusion detection system for computer network security.

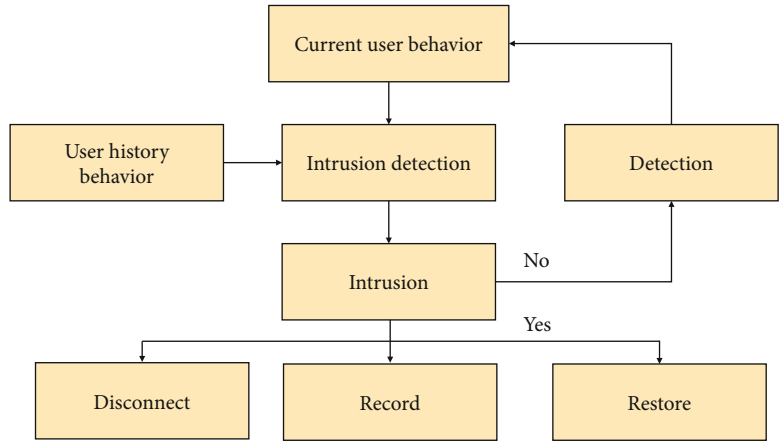


Figure 3: Intrusion detection principle.

(2) Signal analysis. The information collected above is generally analyzed by three technical means: pattern matching, statistical analysis, and completeness analysis. The first two methods are used for real-time intrusion detection, while integrity analysis is used for postmortem analysis

(3) Real-time recording, alarm, or limited counterattack. The fundamental task of IDS is to make appropriate responses to intrusions. These responses include detailed log records, real-time alarms, and limited counterattack sources. The only technical methods to identify intrusions are user characteristics, intruder characteristics, and activity-based. The structure of the intrusion detection system is shown in Figure 4

3.3. Intrusion Detection Technology Methods. At present, there are many standard intrusion detection technology methods, and a few are listed below for explanation.

(1) *Neural network anomaly detection.* This method can be self-learning and self-adaptable to user behavior and can effectively process and judge the possibility of intrusion according to the actual monitored information. The prediction of the error rate of the next event reflects the abnormal degree of user behavior to a certain extent. At present, this method is widely used, but the method is not yet mature, and there is no more complete product [35–38]

(2) *Probabilistic statistical anomaly detection.* This method is based on the modeling of historical user behavior, and based on early evidence or models, the audit system detects the user's use of the system in real time, according to the user behavior probability stored in the system. The statistical model is used to detect, and when suspicious user behavior is found, it keeps track and monitors and records the user's behavior

(3) *Expert system misuse detection.* Aiming at characteristic intrusion behaviors, expert systems are often used for detection. In the realization of the expert detection system, the knowledge of the safety expert is expressed through the rules of the If-Then structure (or a compound structure). Therefore,

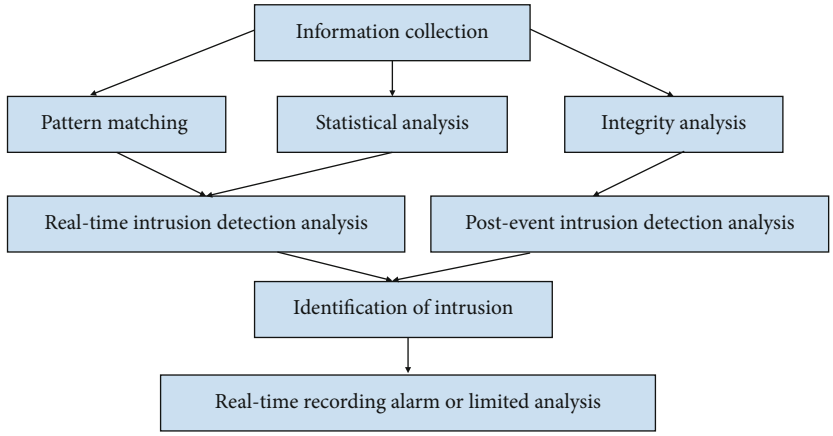


Figure 4: Intrusion detection system structure.

establishing an expert system depends on the completeness of the knowledge base, which depends on the completeness and real-time nature of audit records

(4) *Model-based intrusion detection.* Intruders often use specific behavioral procedures when attacking a system, such as the behavioral sequence of guessing passwords. This behavioral sequence constitutes a model with specific behavioral characteristics. According to the attack represented by this model, the behavioral characteristics of intention can detect malicious attack attempts in real time

Invasion technology has undergone rapid changes in scale and method, and intrusion methods and techniques have also progressed and developed. Outlier mining is an important direction of research on intrusion detection technology. Outlier mining is to mine a small part of abnormal data from a large amount of complex data, which is novel and significantly different from conventional data patterns. Outlier mining is often anomalous data mixed in a large amount of high-dimensional data, and these anomalous data will bring serious consequences. Currently, in the field of intrusion detection research, many scholars apply cluster analysis to anomaly detection. But through the analysis of the characteristics of intrusions, it can be considered that outlier mining technology is more suitable for anomaly-based intrusion detection than clustering technology. Because there is a clear difference between normal behavior and abnormal behavior, and in real applications, the number of abnormal behaviors is much lower than the number of normal behaviors [39–44]. Compared with the entire network behavior, the intrusion behavior is a small number of abnormal data, which can be treated as an isolated point in the dataset, which can better reflect the nature of the intrusion. Therefore, intrusion detection can be converted into the problem of mining outliers in the network behavior dataset. Compared with other anomaly detection technologies, the anomaly detection technology based on outlier mining does not require a training process. Therefore, it overcomes the current anomaly detection problems. They are faced with the problem of a high false alarm rate caused by incomplete normal patterns in training samples.

3.4. Steps Involved in Proposed Intrusion Detection Algorithm. The outlier mining algorithm proposed in this article is based on the similarity index described in the following steps.

Step 1. Enter the dataset: A matrix with n rows and m columns indicates that each record in the original network record set of n intrusion detection has m characteristic attributes. Suppose the domain $X = \{x_1, x_2, \cdots, x_n\}$ is the object to be detected, and each object has m indicators, namely, $x_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}$, $i = (1, 2, \cdots, n)$, expressed as a data matrix:

$$X = \begin{pmatrix} x_{11} & K & x_{1m} \\ M & O & M \\ x_{n1} & L & x_{nm} \end{pmatrix}. \tag{1}$$

Step 2. Find the set of isolated points in n objects: in order to judge the degree of dispersion of each object in x, first calculate the similarity coefficient r_{ij} between each object pair and form a similarity coefficient matrix, namely,

$$R = \begin{pmatrix} r_{11} & K & r_{1n} \\ M & O & M \\ r_{n1} & L & r_{nn} \end{pmatrix}, \tag{2}$$

$$r = 1 - \sqrt{\frac{1}{n}\sum_{k=1}^{m}\left(x_{ik} - x_{jk}\right)^2}, \tag{3}$$

$$p_i = \sum_{j=1}^{n} r_{ij}. \tag{4}$$

Among them, p_i is the sum of the i^{th} row of the relative coefficient matrix. The smaller the value, the farther the

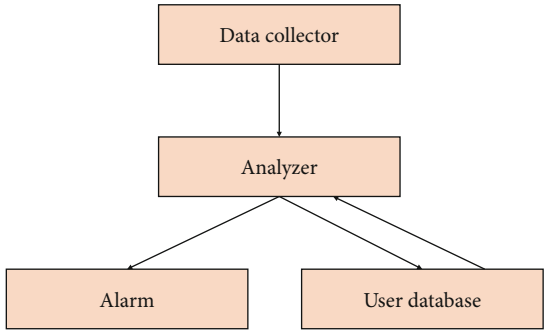


Figure 5: System structure of anomaly intrusion detection based on similarity and isolated point analysis method.

Table 1: Data object distribution of lypmphography dataset.

Classification	Classification	Percentage
General category	Categories 2 and 3	95.9%
Outlier class	Categories 1 and 4	4.1%

object i is from other objects, which is the postoption of the isolated point set.

$$\lambda_i = \frac{p_{\max} - p_i}{p_{\max}} \times 100\%. \quad (5)$$

Among them, λ is the threshold, and objects with $\lambda_i \geq \lambda$ are considered as outliers.

(1) Anomaly intrusion detection system based on similarity and outlier analysis method: the abnormal intrusion detection system is to obtain the audit data of the system and use them as the original data value of the intrusion detection data source and user behavior characteristics, and then use the similarity and outlier analysis method to divide the original data values into normal datasets and isolated datasets. Finally, point the dataset to determine whether it is under attack

(2) The system structure of abnormal intrusion detection is based on similarity and outlier analysis methods. It is composed of collectors, analyzers, alarms, and user databases. The structure diagram is shown in Figure 5

(3) Working principle of the proposed algorithm:

(i) The data collector mainly collects the original audit data and then transmits the data to the data analyzer

(ii) The data analyzer has the functions of data transmission and data analysis. On the one hand, it receives the alarm information from the qualitative data and transmits it to the alarm. On the other hand, the data analyzer transmits the quantitative data to the user database

Table 2: Data object distribution of the glass dataset.

Classification	Classification	Percentage
General category	Categories 1, 2, 3, and 7	89.8%
Outlier class	Categories 5 and 6	10.2%

Table 3: Outlier detection results on the lypmphography dataset.

K value	Direct isolation	Derivative outliers	Correct isolation points	Accuracy
8	3	12	4	66.7% (4/6)
12	5	10	4	66.7% (4/6)
16	8	15	6	66.7% (4/6)

Table 4: Outlier detection results on the glass dataset.

K value	Direct isolation	Derivative outliers	Correct isolation points	Accuracy
8	10	24	16	66.7% (16/22)
10	12	28	18	73.7% (18/22)
16	16	33	22	100% (22/22)

(iii) The user database uses an outlier mining algorithm based on similarity sum to divide quantitative data into the normal dataset and outlier dataset. Then, transfer these two datasets to the analyzer

(iv) The analyzer receives the alarm information from the outlier dataset, transmits the information to the alarm, and then transmits the normal dataset to the user database

(v) The user database updates the normally transmitted dataset so that the user database in the intrusion detection system can accurately describe user behavior characteristics. The description of algorithm for intrusion detection

Step 1. Get the original data x_i of the current user's resource usage at a certain moment.

Step 2. Calculate the degree of dispersion of each object in X; that is, calculate the similarity coefficient r_{ij} between each object.

Step 3. Calculate P_i and λ_i in the i^{th} row of the similarity coefficient matrix.

Step 4. If the object with $\lambda i \geq \lambda$ is considered an outlier set, there is abnormal behavior and alarms; otherwise, it belongs to normal user behavior. The user database is updated.

From the perspective of time consumption, it is mainly the comparison of distance. Although the anomaly detection

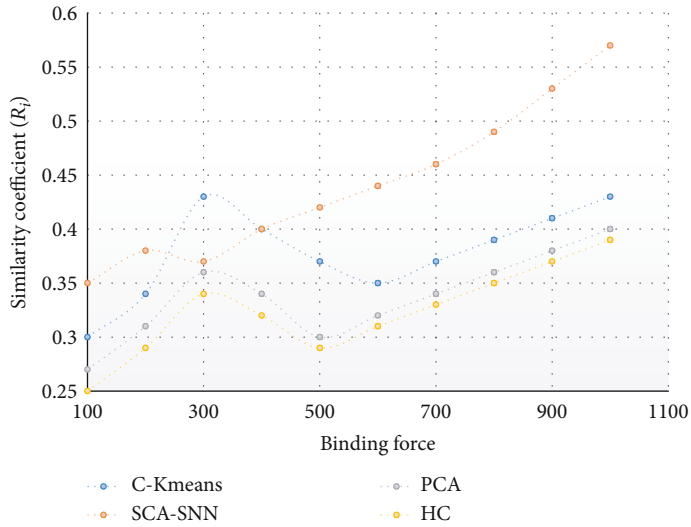


Figure 6: Experimental results on the lypmphography dataset.

technology of outlier mining adds extra time and space consumption than the cluster-based anomaly detection technology, it also improves the algorithm's performance and improves the performance of intrusion attacks and detection rate.

4. Results and Discussion

The experimental datasets in this article are all from UCI real datasets, and the experimental results are the average of data obtained from multiple experiments. The performance judgment of outlier detection is mainly based on analyzing the proportion of correct outliers detected in all outliers, and the evaluation function of semisupervised clustering algorithm is used.

The known number of paired constraints is the initial set of constraints randomly generated by the system. The known constraints are subtracted from the evaluation index because, in the semisupervised clustering algorithm, the known supervision information cannot reflect the effect of the clustering algorithm. The experiment uses the lypmphography dataset and the glass dataset for comparison experiments. The object distribution of the dataset is shown in Tables 1 and 2.

The experimental results of outlier detection are shown in the table. The first column in the table is the K value. The second column indicates the number of isolated points obtained by analyzing the nearest neighbors of the data points; that is, the data points with very few nearest neighbors are direct. It is judged as an isolated point. The second column indicates the number of isolated points obtained from the nearest neighbor set of the isolated point called a derived isolated point. The fourth column refers to the true isolated point among the isolated points obtained in the second column. Finally, the last column is the correct rate of outlier detection.

The experimental results of the lypmphography dataset are shown in Table 3. Since the number of real categories in the dataset is 4, the experiment starts training from $K = 4$. When $K = 8$, we get that the nearest neighbor set of a data point contains very few objects, so we determine it as an outlier and analyze the data points in the nearest neighbor set of the outlier. So, when the K value is 8, we get 12 outliers, including 4 correct outliers. When $K = 12$, although the accuracy rate of outlier detection is not improved from the table above, the number of outliers obtained from analyzing the characteristics of the classes decreases, and some data points that are judged incorrectly are removed.

From this perspective, it is clear that the detection rate has increased by 7%. When $K = 16$, all 6 outliers were detected, and the detection rate reached 100%. The algorithm also has apparent effects on the glass dataset (as shown in Table 4).

The two semisupervised clustering algorithms: C-Kmeans and Sine Cosine Algorithm-based sharing nearest neighbor (SCA-SNN), are evaluated in this study for outlier detection for both the lypmphography and glass dataset. Furthermore, the semisupervised clustering is performed on the "denoising" dataset after detecting the outliers. The experimental results obtained from these methods are also compared with other state-of-the-art methods like hierarchical clustering (HC) and principle component analysis (PCA) to determine the effectiveness of semisupervised clustering. The experimental results are shown in Figure 6–9.

Figure 6 presents four different algorithms for the lypmphography dataset experimental outcomes before finding the outliers and without performing the denoising step. The experimental dataset utilized in Figure 7 is the "denoising" lypmphography dataset, which only contains the second and third types of the original lypmphography dataset. For experimental comparison on this dataset (done in Figure 7), it can be seen that as the number of paired constraints increases, the effect of the SCA-SNN algorithm is steadily increasing among all other algorithms. However, after removing the outliers, the C-Kmeans algorithm also provides relatively stable performance, and there is no significant fluctuation of the clustering results. But from the overall

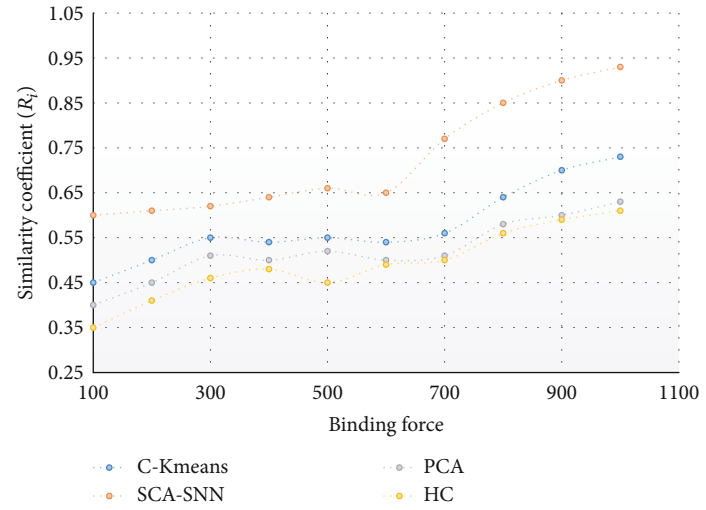


Figure 7: Experimental results of the "denoising" lypmphography dataset.

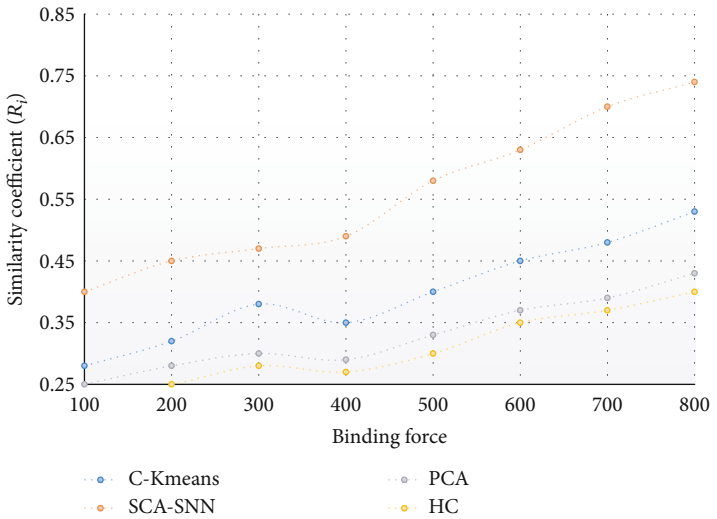


Figure 8: Experimental results on the glass dataset.

clustering results, the performance of the SCA-SNN algorithm is better than the C-Kmeans, PCA, and HC algorithm.

All the algorithms do not have noticeable results on the original lypmphography dataset. Although the experimental results have improved as the number of paired constraints increases when the number of constraints reaches 1000, the correct judgment rate of the C-Kmeans algorithm is only 0.48, and the SCA-SNN algorithm only reaches 0.58, which indicates that the data is concentrated. Furthermore, the outlier data caused a great impact on the clustering results and weakened the guiding role of the paired constraints, resulting in the entire clustering algorithm without good results.

Figures 8 and 9 are the experimental results on the glass dataset. It can be found from Figure 8 that the C-Kmeans algorithm exhibits its instability due to the existence of "noise" data. From the overall perspective of the clustering results, the clustering performance of the SCA-SNN algorithm is always better than that of the C-Kmeans, PCA, and HC algorithm.

Regardless of whether there are outliers in the dataset, the clustering effect of the SCA-SNN algorithm is better than that of the C-Kmeans algorithm and the other state-of-the-art algorithms, especially after removing the outliers. On the set, the SCA-SNN algorithm has better experimental results.

From the above four experimental results, the outlier detection-based SCA-SNN algorithm has the best experimental effect on the dataset without outliers, which shows that the detection of outliers is a crucial process and fully validates the clustering performance of the outlier detection-based SCA-SNN algorithm. In many practical applications, the dataset often contains some outliers. These outliers may contain potentially valuable information. Therefore, mining outliers can effectively improve the performance of clustering

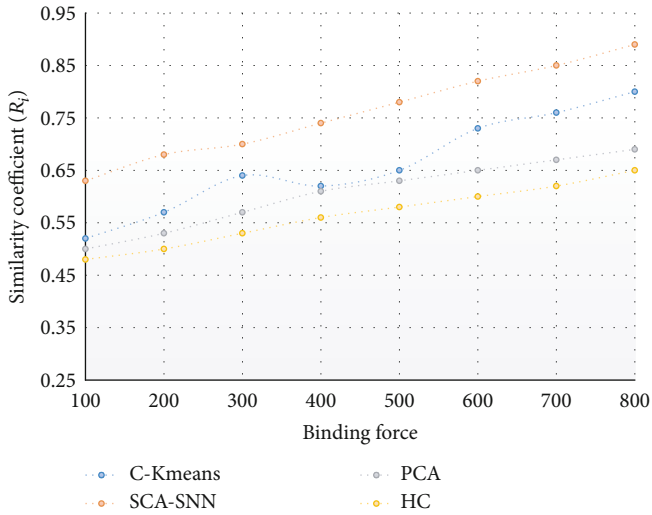


Figure 9: Experimental results of the "denoising" glass dataset.

and get the correct classification. It can also help people obtain more valuable information.

5. Conclusion

This paper proposes an outlier detection and semisupervised clustering algorithm based on nearest neighbor similarity. The wood algorithm uses the C-Kmeans algorithm to train the dataset, which can obtain a reasonable and accurate data sharing nearest neighbor set, and quickly and accurately detect global outliers based on the obtained results, which also has a significant effect on local outliers. The algorithm effectively avoids the insufficient preprocessing of noise points and the influence of inaccurate input parameters on the results. Also, it overcomes the problem of large calculations such as the Jarvis-Patrick algorithm. In the process of semisupervised clustering, the acquired paired prior knowledge is expanded to maximize the guiding effect of prior knowledge. The algorithm detects outliers and effectively avoids the dependence on parameters and eliminates the influence of outliers on clustering. The algorithm combines prior knowledge and expands, making the clustering process "rules to follow." Experiments on real datasets show that the outlier detection algorithm combined with semisupervised clustering results in the best clustering results. Furthermore, the experimentation reveals that the outlier detection-based SCA-SNN algorithm has the best experimental effect on the dataset without outliers. This approach shows that the detection of outliers is crucial and fully validates the clustering performance of the outlier detection-based SCA-SNN algorithm.

With the increasingly prominent network security issues, the research of intrusion detection technology has attracted more and more attention. An intrusion detection algorithm based on outlier data mining is given based on the in-depth study of data mining intrusion detection technology. Outlier mining technology can complete anomaly detection work. When the abnormal data is much smaller than the normal data, the detection result is better than anomaly detection technology based on clustering. In general, the statistical distribution of abnormal and normal behavior in-network data meets the conditions of use of outlier mining. Network security has always been a concern of people. However, with the further development of the network and the diversification of hacker attacks, there is still much research and challenging issues to be solved urgently.

Data Availability

All data has been shared within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding this study.

Acknowledgments

The authors would like to acknowledge the support of Taif University Researchers Supporting Project number (TURSP-2020/239), Taif University, Taif, Saudi Arabia.

References

[1] M. Masud, G. S. Gaba, S. Alqahtani et al., "A lightweight and robust secure key establishment protocol for Internet of medical things in COVID-19 patients care," *IEEE Internet of Things Journal*, 2021.

[2] M. Masud, M. Alazab, K. Choudhary, and G. S. Gaba, "3P-SAKE: privacy-preserving and physically secured authenticated key establishment protocol for wireless industrial networks," *Computer Communications*, vol. 175, pp. 82–90, 2021.

[3] R. G. Bace, *Intrusion detection*, Sams Publishing, 2000.

[4] K. Scarfone and P. Mell, *Guide to intrusion detection and prevention systems (idps)*, vol. 800, no. 2007, 2007NIST Special Publication, 2007.

[5] G. Rathee, A. Sharma, R. Kumar, F. Ahmad, and R. Iqbal, "A trust management scheme to secure mobile information

centric networks," *Computer Communications*, vol. 151, pp. 66–75, 2020.

[6] M. Poongodi, A. Sharma, V. Vijayakumar et al., "Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system," *Computers & Electrical Engineering*, vol. 81, article 106527, 2020.

[7] B. Dayıoğlu, *Use of Passive Network Mapping to Enhange Network Intrusion Detection, [M.S. thesis]*, University Library, Middle East Technical University, Turkey, 2001.

[8] T. Lappas and K. Pelechrinis, *Data Mining Techniques for (Network) Intrusion Detection Systems*, vol. 92521, Department of Computer Science and Engineering UC, Riverside, Riverside CA, 2007.

[9] G. Dhiman, K. K. Singh, M. Soni et al., "MOSOA: a new multi-objective seagull optimization algorithm," *Expert Systems with Applications*, vol. 167, article 114150, 2021.

[10] G. Rathee, A. Sharma, H. Saini, R. Kumar, and R. Iqbal, "A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology," *Multimedia Tools and Applications*, vol. 79, no. 15-16, article 7835, pp. 9711–9733, 2020.

[11] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[12] V. Singh and S. Puthran, "Intrusion detection system using data mining a review," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 587–592, Jalgaon, India, 2016.

[13] D. Rathore and A. Jain, "Design hybrid method for intrusion detection using ensemble cluster classification and som network," *International Journal of Advanced Computer Research*, vol. 2, no. 3, pp. 181–186, 2019.

[14] M. Masud, G. S. Gaba, K. Choudhary, R. Alroobaea, and M. S. Hossain, "A robust and lightweight secure access scheme for cloud based E-healthcare services," *Peer-to-Peer Networking and Applications*, pp. 1–15, 2021.

[15] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare," *IEEE Internet of Things Journal*, 2021.

[16] W. Meng, E. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: a review," *IEEE Access*, vol. 6, no. 1, pp. 10179–10188, 2018.

[17] F. Farahnakian and J. Heikkonen, "Anomaly-based intrusion detection using deep neural networks," *International Journal of Digital Content Technology and its Applications*, vol. 12, pp. 70–118, 2018.

[18] T. Qian, Y. Wang, M. Zhang, and J. Liu, "Intrusion detection method based on deep neural network," *Huazhong Keji Daxue Xuebao*, vol. 46, no. 1, pp. 6–10, 2018.

[19] R. Priyadharshini and E. J. Leavline, "Cuckoo optimisation based intrusion detection system for cloud computing," *International Journal of Computer Network and Information Security*, vol. 10, no. 11, pp. 42–49, 2018.

[20] A. U. Makarfi, K. M. Rabie, O. Kaiwartya, X. Li, and R. Kharel, "Physical layer security in vehicular networks with reconfigurable intelligent surfaces," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, Antwerp, Belgium, 2020.

[21] A. Jayaswal and R. Nahar, "Detecting network intrusion through a deep learning approach," *International Journal of Computer Applications*, vol. 180, no. 14, pp. 15–19, 2018.

[22] S. Kumar, K. Singh, S. Kumar, O. Kaiwartya, Y. Cao, and H. Zhou, "Delimitated anti jammer scheme for Internet of vehicle: machine learning based security approach," *IEEE Access*, vol. 7, pp. 113311–113323, 2019.

[23] R. Sun, L. Shi, C. Yin, and J. Wang, "An improved method in deep packet inspection based on regular expression," *Journal of Supercomputing*, vol. 75, no. 6, pp. 3317–3333, 2019.

[24] H. Ji, Y. Wang, H. Qin, Y. Wang, and H. Li, "Comparative performance evaluation of intrusion detection methods for in-vehicle networks," *IEEE Access*, vol. 6, pp. 37523–37532, 2018.

[25] J. Zhang, "Detection of network protection security vulnerability intrusion based on data mining," *International Journal of Network Security*, vol. 21, no. 6, pp. 979–984, 2019.

[26] P. Narwal, D. Kumar, and S. N. Singh, "A hidden markov model combined with markov games for intrusion detection in cloud," *Journal of Cases on Information Technology*, vol. 21, no. 4, pp. 14–26, 2019.

[27] H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, "An intrusion detection framework based on hybrid multi-level data mining," *International Journal of Parallel Programming*, vol. 47, no. 4, pp. 740–758, 2019.

[28] A. Yang, Y. Zhuansun, C. Liu, J. Li, and C. Zhang, "Design of intrusion detection system for internet of things based on improved bp neural network," *IEEE Access*, vol. 7, pp. 106043–106052, 2019.

[29] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to Internet of things deployment: survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020.

[30] S. Naseer, Y. Saleem, S. Khalid et al., "Enhanced network anomaly detection based on deep neural networks," *IEEE access*, vol. 6, pp. 48231–48246, 2018.

[31] X. Li, M. Xu, P. Vijayakumar, N. Kumar, and X. Liu, "Detection of low-frequency and multi-stage attacks in industrial Internet of things," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8820–8831, 2020.

[32] Y. Xun, J. Liu, and Y. Zhang, "Side-channel analysis for intelligent and connected vehicle security: a new perspective," *IEEE Network*, vol. 34, no. 2, pp. 150–157, 2020.

[33] A. Gupta, R. K. Jha, P. Gandotra, and S. Jain, "Bandwidth spoofing and intrusion detection system for multistage 5g wireless communication network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 618–632, 2018.

[34] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019.

[35] M. Poongodi, A. Sharma, M. Hamdi, M. Maode, and N. Chilamkurti, "Smart healthcare in smart cities: wireless patient monitoring system using IoT," *The Journal of Supercomputing*, no. article 3765, pp. 1–26, 2021.

[36] X. Xu, L. Li, and A. Sharma, "Controlling messy errors in virtual reconstruction of random sports image capture points for complex systems," *International journal of system assurance engineering and management*, pp. 1–8, 2021.

[37] G. K. Sodhi, S. Kaur, G. S. Gaba, L. Kansal, A. Sharma, and G. Dhiman, "COVID-19: role of robotics, artificial intelligence,

and machine learning during pandemic," *Current Medical Imaging*, vol. 17, 2021.

[38] Y. Liu, Q. Sun, A. Sharma, A. Sharma, and G. Dhiman, "Line monitoring and identification based on roadmap towards edge computing," *Wireless personal communications*, no. article 8272, pp. 1–24, 2021.

[39] M. Fan and A. Sharma, "Design and implementation of construction cost prediction model based on SVM and LSSVM in industries 4.0," *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, pp. 145–157, 2021.

[40] H. Sun, M. Fan, and A. Sharma, "Design and implementation of construction prediction and management platform based on building information modelling and three-dimensional simulation technology in industry 4.0," *IET collaborative intelligent manufacturing*, 2021.

[41] X. Ren, C. Li, X. Ma et al., "Design of multi-information fusion based intelligent electrical fire detection system for green buildings," *Sustainability*, vol. 13, no. 6, p. 3405, 2021.

[42] A. Sharma and R. Kumar, "A framework for pre-computated multi-constrained quickest QoS path algorithm," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 3-6, pp. 73–77, 2017.

[43] M. Poongodi, M. Hamdi, A. Sharma, M. Ma, and P. K. Singh, "DDoS detection mechanism using trust-based evaluation system in VANET," *IEEE Access*, vol. 7, pp. 183532–183544, 2019.

[44] D. Kumar, A. Sharma, R. Kumar, and N. Sharma, "A holistic survey on disaster and disruption in optical communication network," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 13, no. 2, pp. 130–135, 2020.

Hindawi
Journal of Sensors
Volume 2021, Article ID 5590472, 14 pages
https://doi.org/10.1155/2021/5590472

Research Article

Quantum Clone Elite Genetic Algorithm-Based Evaluation Mechanism for Maximizing Network Efficiency in Soil Moisture Wireless Sensor Networks

Jing Xiao, Yang Liu, and Jie Zhou

College of Information Science and Technology, Shihezi University, Shihezi 832000, China

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 7 January 2021; Revised 19 March 2021; Accepted 25 April 2021; Published 17 May 2021

Academic Editor: Omprakash Kaiwartya

In agriculture, soil moisture wireless sensor networks (SMWSNs) are used to monitor the growth of crops for obtaining higher yields. The purpose of this paper is to improve the network efficiency of SMWSNs. Therefore, we propose a novel network efficiency evaluation mechanism which is suitable for soil moisture sensors and design a sensor target allocation model (STAM) for the actual agricultural situation. After that, a quantum clone elite genetic algorithm (QCEGA) is proposed; then, QCEGA is applied to optimize the STAM for obtaining optimal results. QCEGA uses the parallel mechanism of quantum computing to encode individuals, integrates the quantum revolving gate in quantum computing and the concept of cloning in biology to avoid the algorithm from falling into local optimum, and applies the elite strategy to speed up the convergence of the algorithm. Subsequently, the proposed algorithm is compared with simulated annealing (SA) and particle swarm optimization (PSO). Under the novel network efficiency evaluation mechanism, the simulation results demonstrate that the network efficiency based on QCEGA is higher than that of SA and PSO; what is more, QCEGA has better convergence performance. In comparison with traditional wireless sensor network efficiency evaluation approaches, our methods are more in line with the development of modern agriculture and can effectively improve the efficiency of SMWSNs, thus ensuring that crops can have a better growth condition.

1. Introduction

Although the current crop yields have increased, the irrigation management of crops has not been satisfactory [1]. At present, the number of people engaged in agricultural activities is on a downward trend, resulting in increasing labor costs. The Internet of things technology develops quickly, and soil moisture sensor networks (SMWSNs) are already used in the actual production process. Moreover, the development of artificial intelligence, cloud computing, and big data has also greatly improved SMWSN efficiency [2–4]. However, due to the particularity of soil moisture sensors, the original evaluation mechanism for wireless sensor networks is not well applicable to SMWSNs, so it is urgent to propose a novel network efficiency evaluation mechanism [5].

In previous works, it proposes to use sensor nodes to dynamically monitor the land that needs to be irrigated [6–8], which can effectively save human resource. Nonetheless, the network efficiency optimization problem is too complex to be solved by traditional mathematical methods [9, 10]. Yu et al. [11] use PSO to increase the network efficiency, whose effect is better than before, but it has a problem of falling into local optimum. Tang et al. [12] use SA to improve the soil moisture monitoring ability, which has low algorithm complexity. However, it has the disadvantages of local optimum and slow convergence ability.

In this paper, a novel network efficiency evaluation mechanism, a sensor node target allocation model of SMWSNs, and a novel algorithm are proposed. The purpose of the evaluation mechanism and the model is to maximize the network efficiency of the SMWSNs. In addition, to verify the model's effectiveness, the mathematical model of target allocation is firstly shown. Subsequently, the quantum operator, clonal operator, and elite operator are applied into the

traditional genetic algorithm (GA). The simulation results show that the proposed algorithm effectively improves the network efficiency.

The main contributions can be listed as follows:

(1) A novel evaluation mechanism for SMWSNs is proposed

(2) We have designed a novel sensor target allocation model suitable for SMWSNs, which comprehensively considers different sensor positions and soil depth values

(3) A quantum clone elite genetic algorithm (QCEGA) that combines the concepts of quantum computing and cloning is proposed. This algorithm can obtain the optimal target allocation method and has good convergence performance

(4) The target allocation method based on QCEGA is compared with SA and PSO in terms of network efficiency, convergence performance, and computational complexity

The structure of the paper can be shown as follows. Related works are given in Section 2. Section 3 shows the network efficiency evaluation mechanism and the target allocation model. In Section 4, to obtain the optimal solution, QCEGA is introduced. Section 5 presents the effectiveness of QCEGA through simulation experiments and makes discussion. In Section 5, the conclusion part is given.

2. Related Work

Recently, there are relatively few researches on soil moisture wireless sensor networks. However, many scholars have proposed ways to improve the efficiency of wireless sensor networks (WSNs).

In [13], with the purpose of enhancing the allocation efficiency of WSNs, the authors improve the wireless sensor network coverage model by analyzing the structural characteristics of the wireless sensor network. However, because it introduces the optimization links into the PSO, it has increased the computational complexity. On the other hand, to increase the energy management efficiency of WRSNs, Gong et al. [14] propose a two-phase scheme for obtaining the best energy allocation method by dealing with the model's game equilibrium. Although the two-phase scheme improves coverage efficiency and energy supply efficiency, it has poor convergence performance. In addition, a method of node optimization coverage for passive monitoring system of 3D-WSN based on a link model is proposed in [15]. The method constructs a three-dimensional WSN link coverage model based on a wireless link-aware area and uses a network coverage based on a cube to describe the network's service quality. Cao et al. [16] mention the concept of edge computing, and a localization method for WSNs is proposed in this paper. A dynamic deployment technology which bases an improved genetic algorithm (IDDT-GA) is proposed to minimize the number of nodes and reduce the overlap area between adjacent nodes [17]. However, the algorithm has long running time. Zanaj et al. [18] propose a customizable heuristic method for WSN topology design based on a genetic algorithm. This method enables users to construct network topologies which are selected by optimizing different potential configurations, such as coverage, energy efficiency, system node degrees in the environment, and network lifetime, but the setting of related weight coefficients is a problem worthy of careful consideration.

In addition, in terms of optimizing the network efficiency of WSNs, Kumar et al. [19] consider the impact of multiple parameters on the energy consumption of WSNs and propose a lifetime maximization technology based on an ant colony algorithm and Huffman coding. After that, the authors verify the validity of four performance indicators: energy consumption, average remaining energy, number of live sensors, and standard deviation of energy. On the other hand, to reduce the cost of WSN resource sharing, a nondominated sorting genetic algorithm is proposed to maximize the fault tolerance of the network and minimize the delay [20].

In recent years, quantum optimization algorithms have been increasingly used to improve the network efficiency of WSNs. Wu et al. [21] propose a positioning algorithm based on quantum particle swarm optimization, which uses the parallelism of quantum computing to improve the accuracy of positioning. Similarly, Kanchan and Pushparaj [22] also propose a quantum heuristic PSO, but their purpose is to extend the lifetime of the network by clustering the nodes in WSNs. In addition, Prasad and Panigrahi [23] merge the quantum concept into PSO to optimize the maximum likelihood (ML) parameters for enhancing the positioning accuracy of WSNs. An algorithm that combines quantum computing and artificial bee colony is proposed to optimize the search of bees to improve food source substitution strategies, thereby optimizing data collection and distribution in WSNs [24]. Chu et al. [25] use quantum computing to ameliorate the performance of the symbiosis organism search, which improves the algorithm's running speed and convergence ability. However, the simple use of quantum optimization algorithms cannot effectively improve the quality of the solution, and some other techniques need to be combined to further optimize the search capabilities of the algorithm.

In this paper, a novel genetic algorithm combining quantum optimization, cloning operator, and elite operator is proposed. The algorithm has good performance in terms of solution quality and convergence speed in enhancing network efficiency in SMWSNs.

3. Evaluation Mechanism and System Model

3.1. Problem Description. In the SMWSNs, moisture sensors should be placed in the land to monitor the moisture status of crops. Different from other wireless sensors, multiple moisture sensors need to be placed at the same target point to work together, and these sensors are located at different depth values. We need to place limited soil moisture sensors in suitable locations to obtain the best network efficiency and then grasp the moisture information of crops in time to

Table 1: List of symbols.

Notation	Description
m	Number of soil moisture sensor nodes
n	Number of target points
x	Number of sensors that need to be placed at each target point
y	Number of depth values
D	Sensor placement advantage matrix
D^*	Actual sensor placement advantage matrix
V	Sensor monitoring effect matrix
Y	Solution in decimal form
Y^*	Solution in binary form
L	Fitness matrix
l	Fitness value of each sensor placement
fit	Fitness function

irrigate them. In addition, Table 1 is an explanation of the symbols that appear in this part.

3.2. Evaluation Mechanism. To better describe the network efficiency of SMWSNs, we should consider not only the difficulty of placing sensors but also the monitoring effects of soil moisture sensors at different depths. Therefore, we establish two evaluation matrices: one is the sensor placement advantage matrix and the other is the sensor monitoring effect matrix. In view of the limitations of the actual site, we use a random method to generate these two matrices. If it is applied to an actual project, we only need to replace the values in the matrix with real values, which will not affect the problem to be solved. The so-called network efficiency refers to the sum of the monitoring effects of all soil moisture sensors located at different depths. The monitoring effect of each sensor is affected by both the placement advantage matrix and the monitoring effect matrix. In detail, different soil depth values can be expressed as shown in Figure 1.

3.3. Sensor Target Allocation Model. To maximize the network efficiency of SMWSNs, a mathematical model is shown with reference to some restrictions of calculating abilities and constrained monitoring range. In [26], it uses a model based on first-in first-out logic to optimize the task allocation problem in SMWSNs. Paper [27] studies the problem of complex application allocation and proposes a model that considers multifactor and sensor heterogeneity. The model can assign complex applications to appropriate sensor nodes, thereby maximizing the network efficiency of WSNs.

A SMWSN-based sensor allocation model is proposed in this paper. The model can be described concisely as supposing that there are m soil moisture sensor nodes and n target points in SMWSNs. Each target point is evenly distributed in a certain area and has y depth values to choose. x sensor nodes should be put into every target point for monitoring the moisture of the soil ($x < y$). Sensor nodes located at different depths will get different monitoring advantage values. The goal of the paper is to obtain the highest network efficiency value by assigning sensor nodes at different depths.

It is assumed that the monitoring effect of each depth has been evaluated before sensor allocation, and the advantages of sensor placement have been evaluated in advance. The advantage of the j_{th} sensor allocated to the b_{th} target point can be expressed as $D_{j,b}$. The advantage matrix of the sensors being assigned to the target points can be shown as

$$D = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,B-1} & d_{1,B} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,B-1} & d_{2,B} \\ \vdots & & d_{j,b} & & \vdots \\ d_{J-1,1} & d_{J-1,2} & \cdots & d_{J-1,B-1} & d_{J-1,B} \\ d_{J,1} & d_{J,2} & \cdots & d_{J,B-1} & d_{J,B} \end{pmatrix} \quad (J \in \{1,2,\cdots,m\}, B \in \{1,2,\cdots,n\}), \tag{1}$$

where m is the number of soil moisture sensors and n is the number of targets. Each depth has the value of advantage which is shown in

$$V = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,K-1} & v_{1,K} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,K-1} & v_{2,K} \\ \vdots & & v_{b,k} & & \vdots \\ v_{B-1,1} & v_{B-1,2} & \cdots & v_{B-1,K-1} & v_{B-1,K} \\ v_{B,1} & v_{B,2} & \cdots & v_{B,K-1} & v_{B,K} \end{pmatrix} \quad (B \in \{1,2,\cdots,n\}, K \in \{1,2,\cdots,y\}), \tag{2}$$

where y is the number of depth values.

Specifically, if an individual has 4 depth values, 10 soil moisture sensor nodes, and 5 target points, 2 sensor nodes need to be placed in each target point. A solution is generated randomly by the individual in (3).

$$Y = \begin{pmatrix} 2 & 1 & 2 & 3 & 1 \\ 4 & 4 & 3 & 4 & 2 \end{pmatrix}. \tag{3}$$

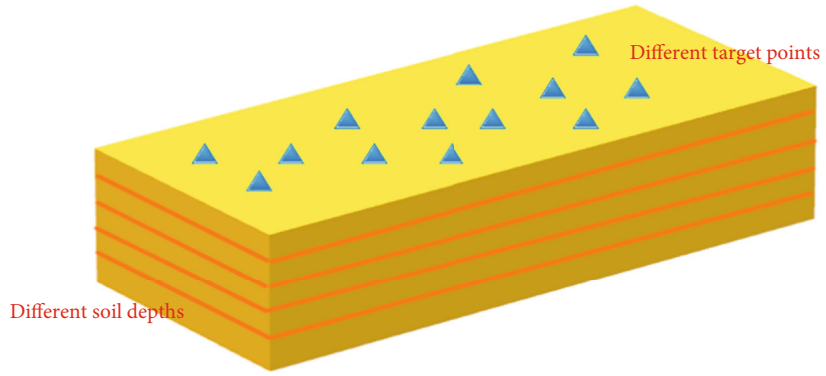


Figure 1: Schematic diagram of different soil depths.

To facilitate the execution of the next operation, the decimal code is ought to be converted to a binary code, and the target allocation matrix in binary form can be expressed as

$$Y^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}. \quad (4)$$

In (4), $Y^*_{j,b} = 1$ stands for the sensor at the j_{th} depth being placed to the b_{th} target point. Otherwise, the sensor at the j_{th} depth is not placed to the b_{th} target point.

According to equations (1) and (4), D^* is calculated in (5). It represents the actual sensor placement advantage value.

$$D^* = D \cdot Y^*, \quad (5)$$

where Y^* is the target allocation matrix in binary form and D is the sensor placement advantage matrix. The value in D is the advantage value of the sensor at the j_{th} depth being placed to the b_{th} target point. In the previous example, D^* can be expressed as

$$D^* = \begin{pmatrix} 0 & d_{1,2} & 0 & 0 & d_{5,1} \\ d_{2,1} & 0 & d_{2,3} & 0 & d_{5,2} \\ 0 & 0 & d_{3,3} & d_{3,4} & 0 \\ d_{4,1} & d_{1,4} & 0 & d_{4,4} & 0 \end{pmatrix}. \quad (6)$$

Since the advantage values of sensor nodes at different soil depths are placed in D^*, with the aim of obtaining the overall SMWSN advantage value, it is necessary to multiply D^* with the sensor monitoring effect matrix V for obtaining the fitness matrix L, which is shown in

$$L = V \cdot D^*. \quad (7)$$

According to the example cited above, L can be specifically expressed as

$$L = \begin{pmatrix} 0 & l_{1,2} & 0 & 0 & l_{5,1} \\ l_{2,1} & 0 & l_{2,3} & 0 & l_{5,2} \\ 0 & 0 & l_{3,3} & l_{3,4} & 0 \\ l_{4,1} & l_{1,4} & 0 & l_{4,4} & 0 \end{pmatrix}. \quad (8)$$

After the fitness matrix L is obtained, the fitness function in the target allocation problem in SMWSNs can be expressed as

$$\text{fit} = \sum_{j=1}^{y} \sum_{b=1}^{n} L_{j,b}, \quad (9)$$

where y is the number of soil depth values and n is the number of target points.

In SMWSNs, not only should the allocation of the soil moisture sensor nodes meet the actual needs but also it needs to increase the overall network efficiency. The network efficiency of SMWSNs represents the total network benefit monitored by soil moisture sensor nodes. Due to the fact that the soil moisture sensor nodes in SMWSNs are homogeneous, the monitoring efficiency of every sensor node is roughly the same. For SMWSNs, the allocation of soil moisture sensor nodes can usually achieve better service quality and higher overall network efficiency. Therefore, it is necessary to allocate sensor nodes according to the soil moisture monitoring capabilities of sensor nodes, and the sensors locate at different soil depths of different target points.

4. QCEGA-Based Target Allocation in SMWSNs to Maximize Network Efficiency

To obtain a better performance of the target allocation in SMWSNs, a QCEGA-based allocation method is proposed. The method is inspired by the traditional genetic algorithm, and it effectively avoids falling into local optimum. The strategy allows QCEGA to handle the distribution of soil moisture sensors and get the best solution. In the method of QCEGA, the significant improvement is the addition of quantum operator, clone operator, and elite operator to the traditional GA.

Compared with traditional optimization algorithms, GA has the advantages of wide range of use, strong robustness, less restricted conditions, and simple use. GA basically has no restrictions on the problem to be solved and does not require complex mathematical algorithm support. With the development of modern computer technology, its superiority is more obvious when studying some complex problems with multiple factors. With the continuous furthering of research, GA is constantly improving, and its application range is getting wider and wider. However, classic GA is prone to fall into being premature and has poor performance.

The procedures of QCEGA can be described as population encoding, initializing the population, calculating fitness, selection, crossover, mutation, quantum operator, clonal operator, elite operator, and termination condition.

4.1. Population Encoding. The above-mentioned sensor node target allocation is the problem of placing the soil moisture sensor at an appropriate depth. Since the moisture sensors used in this problem have the same structure and the sensors at different target points do not affect each other, binary coding can be used [28, 29]. The idea of coding is to treat the allocation sequence of all sensors as the various genes in the chromosome. The sequence of the genes represents the order of the depth value of the soil. An individual carries a complete allocation schema. For example, there are 6 different target points, each target point has 6 different soil depth values, and 3 soil moisture sensors need to be placed in each target point. Then, the corresponding coding matrix can be expressed as

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{10}$$

In (10), the first column indicates that there are 6 depth values at the first target point, and this column selects the second, third, and fifth depth values to place the soil moisture sensor. The analysis of the remaining columns is the same as that of the first column.

4.2. Initializing the Population. Since the problem of sensor target allocation is to place multiple soil moisture sensors in one target point, the population should be three-dimensional. Due to the particularity of the problem, it is necessary to ensure that a specified number of sensors are placed on a target point to better complete the task of soil moisture monitoring [30]. Therefore, the coding of the population needs to be restricted. Assuming that the quantity of individuals is P, then the population and restriction can be shown as

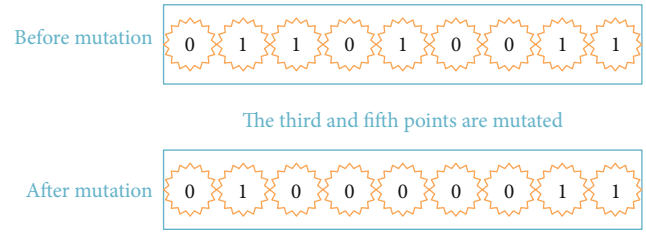


Figure 2: Mutation operation.

$$\text{pop} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,M-1} & h_{1,M} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,M-1} & h_{2,M} \\ \vdots & & h_{n,m} & & \vdots \\ h_{N-1,1} & h_{N-1,2} & \cdots & h_{N-1,M-1} & h_{N-1,M} \\ h_{N,1} & h_{N,2} & \cdots & h_{N,m-1} & h_{N,M} \end{bmatrix} \quad (h_{n,m} \in \{0, 1\}), \tag{11}$$

$$\sum_{n=1}^{N} h_{n,m} = xm \in \{1, 2, \cdots, M\}, \tag{12}$$

where N represents the quantity of target points, M is the quantity of soil depth values, and x is the quantity of sensors that should be placed at each target point.

4.3. Calculating Fitness. With the purpose of solving the problem of target allocation in SMWSNs, individual fitness calculation is an important tool. A high fitness of an individual indicates that a target allocation plan based on its genetic sequence can obtain a better performance. On the contrary, an individual with a lower fitness suggests that its plan is not optimal. QCEGA excludes individuals with low fitness through selection operation and retains individuals with high adaptability. Fitness is the criterion for judging the pros and cons of individuals, and it is calculated according to equation (9).

4.4. Selection. In the genetic algorithm, the selection operation is to make the genes of high-fitness individuals have a greater probability of being inherited to the next generation. In solving the problem of sensor target allocation, the goal is to maximize the network efficiency, so the probability of selection operation is proportional to the fitness of the individual. Assuming that there are G individuals in the population, the probability of the g_{th} individual being determined can be presented as

$$W_{\text{select}}(G_g) = \frac{F(G_g)}{\sum_{i=1}^{G} F(G_i)}, \tag{13}$$

where W_{select} represents the selection probability of an individual, G_g is the g_{th} individual in the population, and $F(G_g)$ represents the fitness value of G_g.

4.5. Crossover. To better solve the problem of sensor target allocation in SMWSNs, the traditional two-point crossover operator needs to be improved [31]. Since the individuals

Input: The quantum probability amplitude matrix $\boldsymbol{Q}$, the number of target points $\boldsymbol{n}$, the number of soil depths $\boldsymbol{y}$, quantum probability amplitude increment Δ
Output: The updated quantum probability amplitude matrix $\boldsymbol{Q}$

```
for i from 1 to n do
    for j from 1 to y do
        if Q (i, j) ==1 and Q (i, j) <1 then
            Q (i, j) = Q (i, j) + Δ
        else if Q (i, j) ==0 and Q (i, j) >0 then
            Q (i, j) = Q (i, j) - Δ
        end if
    end for
end for
Return Q
```

Algorithm 1. Quantum revolving gate.

Input: The quantum probability amplitude matrix $\boldsymbol{Q}$, the number of target points $\boldsymbol{n}$, the number of soil depths $\boldsymbol{y}$, probability $\boldsymbol{R}$, random number $\boldsymbol{r}$ between 0 and 1
Output: The updated quantum probability amplitude matrix $\boldsymbol{Q}$

```
for i from 1 to n do
    for j from 1 to y do
        if r < R then
            Q (i, j) =1 - Q (i, j)
        end if
    end for
end for
Return Q
```

Algorithm 2. Quantum NOT gate.

in the population are three-dimensional, the crossover operation can be transformed into the crossover of the two-dimensional matrix. Supposing there are m target points, the specific crossover operation is to select a cross point $q\,(1 < q < m)$ and then keep the genes in front of q unchanged and exchange the genes behind q. To better understand the crossover process, we can give an example. Before crossover, two individuals can be shown as equations (15) and (16).

$$\text{individual1}_{\text{before}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}, \tag{14}$$

$$\text{individual2}_{\text{before}} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}. \tag{15}$$

The crosspoint q can be 1 to 3; if we take $q = 2$, then the two individuals after crossover can be expressed as equations (17) and (18).

$$\text{individual1}_{\text{after}} = \begin{bmatrix} a_{11} & a_{12} & b_{13} & b_{13} \\ a_{21} & a_{22} & b_{23} & b_{24} \\ a_{31} & a_{32} & b_{33} & b_{34} \end{bmatrix}, \tag{16}$$

$$\text{individual2}_{\text{after}} = \begin{bmatrix} b_{11} & b_{12} & a_{13} & a_{14} \\ b_{21} & b_{22} & a_{23} & a_{24} \\ b_{31} & b_{32} & a_{33} & a_{34} \end{bmatrix}. \tag{17}$$

4.6. Mutation. The mutation operation is to break the original gene combination of the individual and enable the population to evolve in more diverse directions. Through the mutation of genes, the local search ability of the algorithm can be improved. Due to the fact that binary coding is used in QCEGA, the mutation operation is to reverse the selected mutation bit on the premise of mutation probability P. For example, if the position before mutation is 1, then the position after mutation is 0. The mutation process is shown in Figure 2.

4.7. Quantum Operator. The quantum operator is a probabilistic search method based on quantum computing. Compared with the classical swarm intelligent algorithm, quantum operators can effectively improve the algorithm's global search capability and convergence speed [32]. QCEGA

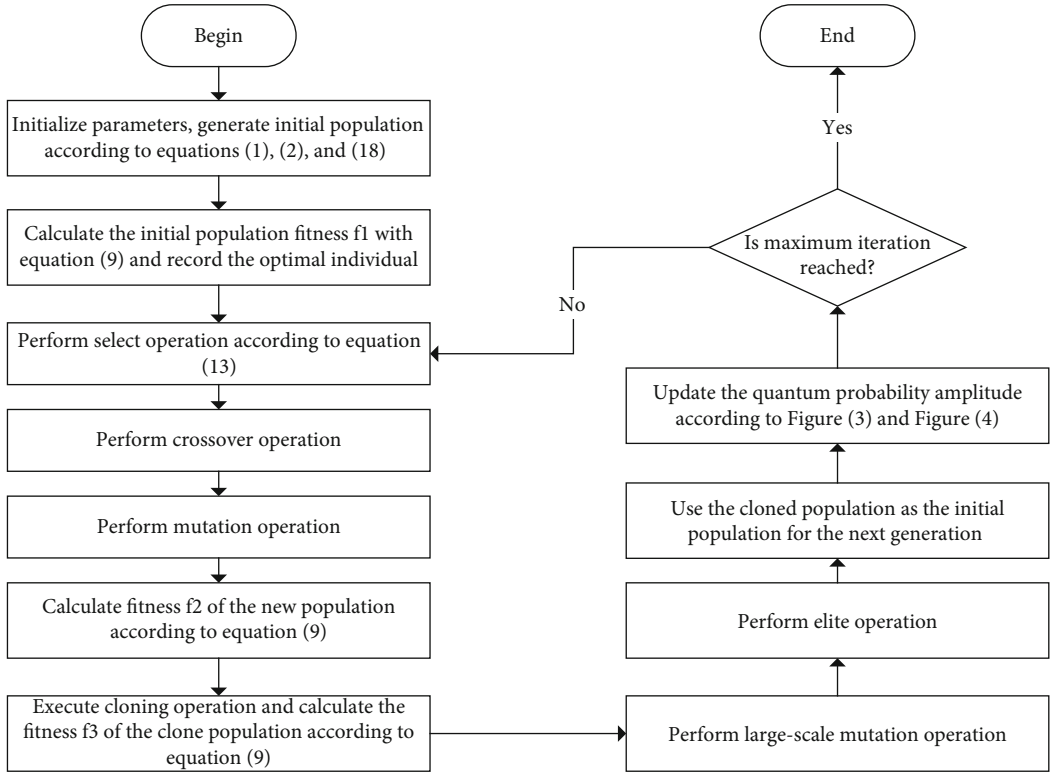


Figure 3: Flowchart of QCEGA.

is a novel algorithm that combines quantum operator. It mainly uses concepts of qubits and quantum superposition states in quantum computing to improve the global search and optimization ability [33]. In QCEGA, each individual in the population is represented by a set of binary numbers and a set of qubit probability amplitudes. Therefore, each individual has the ability to search the solution space in parallel, which effectively improves the search efficiency. In QCEGA, each individual can update the optimal solution through the quantum revolving gate during the evolution process, thus realizing the quantum search mechanism in the iterative process.

In QCEGA, the quantum probability amplitude is used to accelerate the convergence speed and improve the global search capability of the algorithm. The quantum probability amplitude is shown in equation (18).

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,J-1} & q_{1,J} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,J-1} & q_{2,J} \\ \vdots & & q_{i,j} & & \vdots \\ q_{I-1,1} & q_{I-1,2} & \cdots & q_{I-1,J-1} & q_{I-1,J} \\ q_{I,1} & q_{I,2} & \cdots & q_{I,J-1} & q_{I,J} \end{bmatrix} \quad \left(q_{i,j} \in \{0, 1\}\right), \tag{18}$$

where Q is the quantum probability amplitude matrix, I is the number of target points, and J is the number of soil depths.

After all individuals in the population are updated, the quantum probability amplitude will be updated according to the optimal individual in the last iteration with the quantum revolving gate. The quantum revolving gate is shown in Algorithm 1.

In addition, the quantum NOT gate is used to increase the diversity of solutions in the target allocation problem in SMWSNs. The quantum NOT gate is shown in Algorithm 2.

4.8. Clone Operator. To enhance the speed of convergence, QCEGA uses a clone operator to make clonal expansion of the outstanding individuals in the iterative process [34]. The probability of an individual being selected for cloning is determined by its advantage value. In each cloning process, a certain proportion of fitness individuals are selected for cloning. For example, assuming that there are R individuals in the population, clonal expansion is performed by selecting S individuals with the highest advantage as the mother during each iteration ($S < R$). After the cloning process, to increase the diversity of the population, it is necessary to perform mutation operations on the generated individuals. The mutation operation here includes quantum revolving gate operation and quantum NOT operation.

4.9. Elite Operator. The elite operator of QCEGA is the establishment of the elite population. The specific operation is to establish a population from the most adaptive individuals of the previous generation and then use the elite population to replace the least adaptive individuals in the next generation [35]. Due to the fact that the optimal individual may be destroyed by mutation and crossover operations, the elite operator can retain the optimal one to ensure the performance of the algorithm.

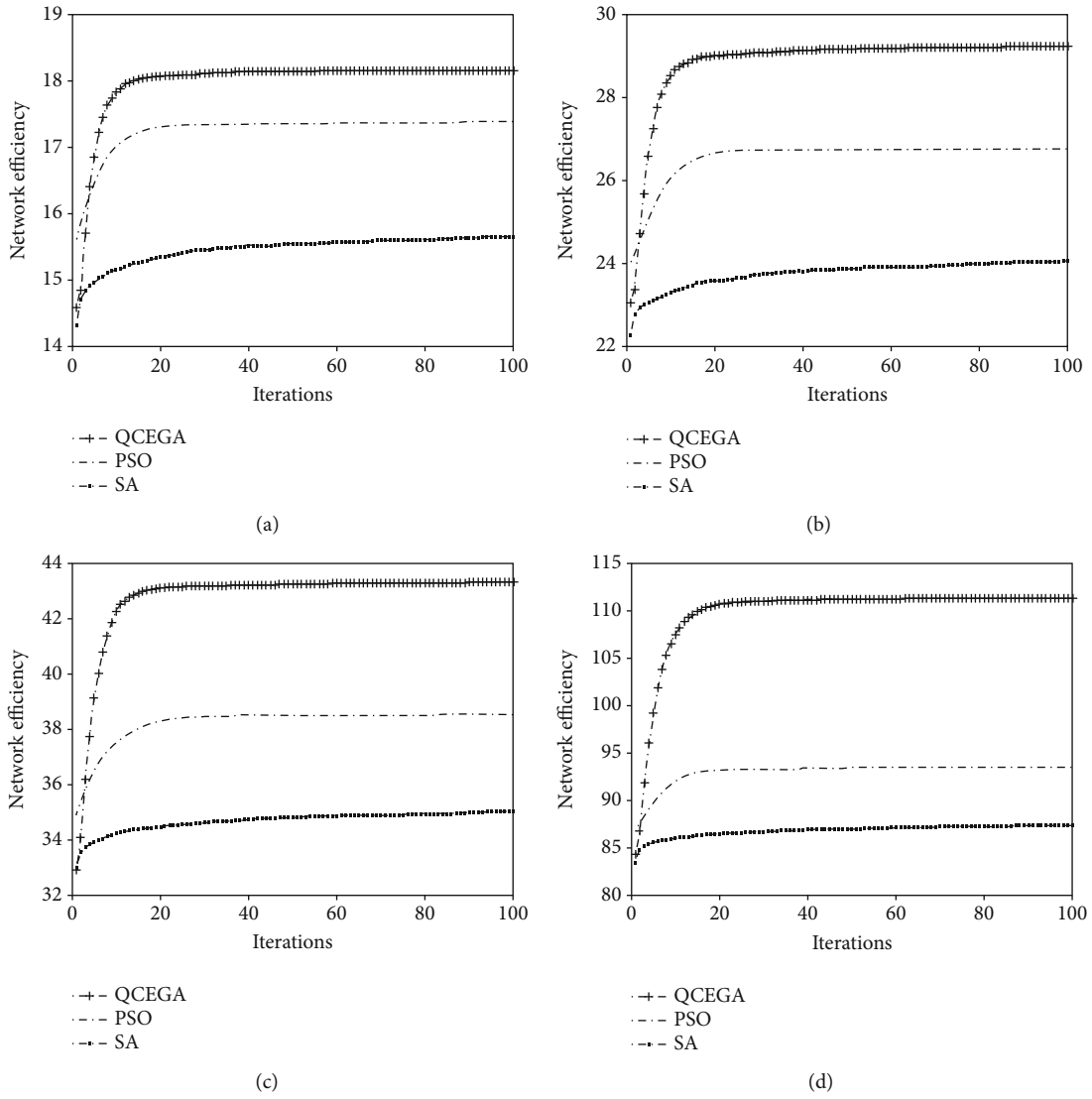


Figure 4: The comparison of the network efficiency of the three algorithms: (a) 15 target points, 6 soil depths, and 3 sensors per target point; (b) 20 target points, 8 soil depths, and 4 sensors per target point; (c) 25 target points, 10 soil depths, and 5 sensors per target point; (d) 50 target points, 12 soil depths, and 6 sensors per target point.

4.10. Steps of the Algorithm. The following steps can be used to describe the execution of QCEGA.

Step 1. Set the initial parameters in QCEGA, generate a matrix of the advantage of placing sensors according to equation (1), generate a matrix of sensor monitoring effects using equation (2), and generate a quantum probability amplitude matrix according to equation (18).

Step 2. Obtain the fitness value of the initial population using equation (9), and record the optimal network efficiency of the individual.

Step 3. Execute the selection operation according to equation (13).

Step 4. Execute the crossover operation.

Step 5. Execute the mutation operation.

Step 6. Calculate the fitness of the new population with equation (9).

Step 7. Execute the clone operator.

Step 8. Do high-probability mutation of the cloned population and make calculation of the new population's fitness according to equation (9).

Step 9. Execute the elite operator.

Step 10. Assign the obtained clonal population to the next-generation population.

Step 11. Update the quantum probability amplitude according to Algorithms 1 and 2.

Step 12. If the maximum iteration is not reached, return to step 3; otherwise, the algorithm terminates execution and outputs the optimal solution.

The algorithm flowchart is shown in Figure 3.

5. Results and Discussion

The definition of network efficiency of SMWSNs is the advantage value that can be obtained under a given soil

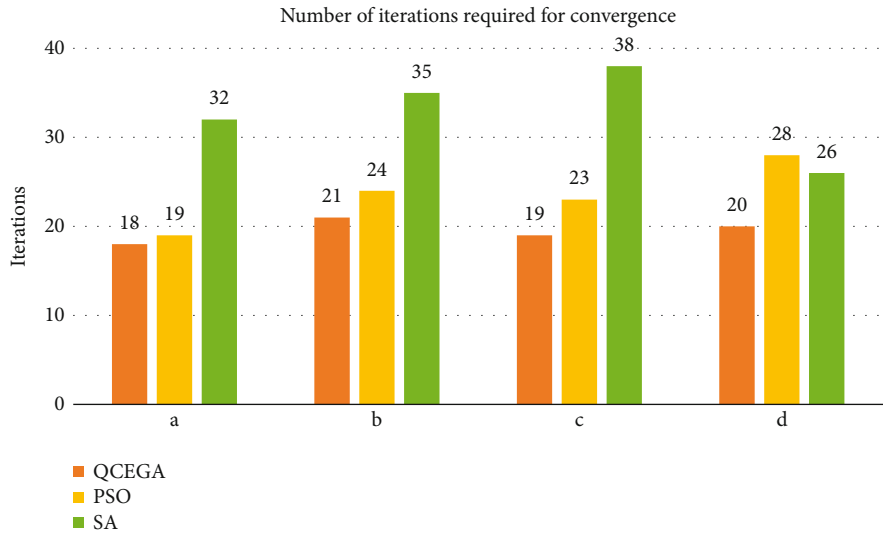


Figure 5: Iterations required for the algorithms to converge: a—15 target points, 6 soil depths, and 3 sensors per target point; b—20 target points, 8 soil depths, and 4 sensors per target point; c—25 target points, 10 soil depths, and 5 sensors per target point; and d—50 target points, 12 soil depths, and 6 sensors per target point.

moisture sensor target allocation scheme. The advantage value consists of two parts. One part is the advantage value of the sensor placement position determined by the distance between the sprinkler device and the target points. The closer the sprinkler device is, the higher the advantage value is. Another part of the advantage value comes from the depth of the soil. The lower the moisture value, the higher the advantage of the depth position. In SMWSNs, the target allocation of soil moisture sensors involves consideration of different target nodes and different soil depths. If the sensor nodes are placed at different target nodes and soil depths, their monitoring effects on the growth environment of crops are dissimilar. What is more, the calculation formula of the advantage is carried out according to equation (9).

The network efficiency of the proposed QCEGA method in target allocation is tested, and QCEGA makes a comparison with the SA and PSO algorithms in SMWSNs for target allocation. Then, the performance of QCEGA is in comparison with different quantities of target points and soil moisture sensors. The whole testing cases are executed on a computer equipped with Pentium 2.1 GHz CPU.

The algorithm efficiency of QCEGA needs to be compared with that of SA and PSO. To make the execution conditions of the algorithm as the same as possible, we set the number of iterations of QCEGA, SA, and PSO to 100 generations, and the size of the population is 40. In QCEGA, the crossover probability and the mutation probability are 0.8 and 0.05, respectively, and the value of the quantum probability amplitude matrix is initialized to 0.5. In SA, the initial temperature is 200, the annealing rate is exponentially decreased, and the decrease factor is 0.96. In PSO, the value of the individual learning factor is 2, the social learning factor is 2, and the speed boundary value is 5.

According to Figures 4(a)–4(d), the convergence of QCEGA, SA, and PSO is shown. Figure 4(a) suggests that QCEGA performs better than SA and PSO under the condition of 15 target points, 6 soil depth values, and 3 soil moisture sensors placed in each target point. Specifically, Figure 4(a) suggests that the network efficiency of SA is lower than that of PSO and the network efficiency of PSO is lower than that of QCEGA. In the first 50 iterations, the network efficiency value of QCEGA varies hugely, and as the times of iterations increase, its convergence speed is accelerated. From the iteration process 50 to 100, the network efficiency value of QCEGA is 18.16; however, the performance of PSO is worse than that of QCEGA, and its network efficiency value is 17.39. The network efficiency value of SA is 15.66. With roughly 50 generations of accumulation, QCEGA's network efficiency advantages are better than those of PSO and SA; QCEGA's performance has been enhanced with its quantum operator, clone operator, and elite operator. In a total of 100 iterations, the network efficiency of SA is lower than that of PSO, the convergence speed of QCEGA is greater, and it generates an optimal solution. In Figures 4(b)–4(d), the

Table 2: The percentage increase of the results.

	QCEGA	PSO	SA
18 target points 7 soil depths 3 sensors per target point	22.72%	10.23%	8.27%
24 target points 8 soil depths 4 sensors per target point	23.30%	9.87%	7.84%
30 target points 10 soil depths 4 sensors per target point	29.97%	10.36%	8.17%
36 target points 12 soil depths 5 sensors per target point	31.29%	11.13%	7.12%
40 target points 15 soil depths 6 sensors per target point	31.67%	7.48%	6.56%

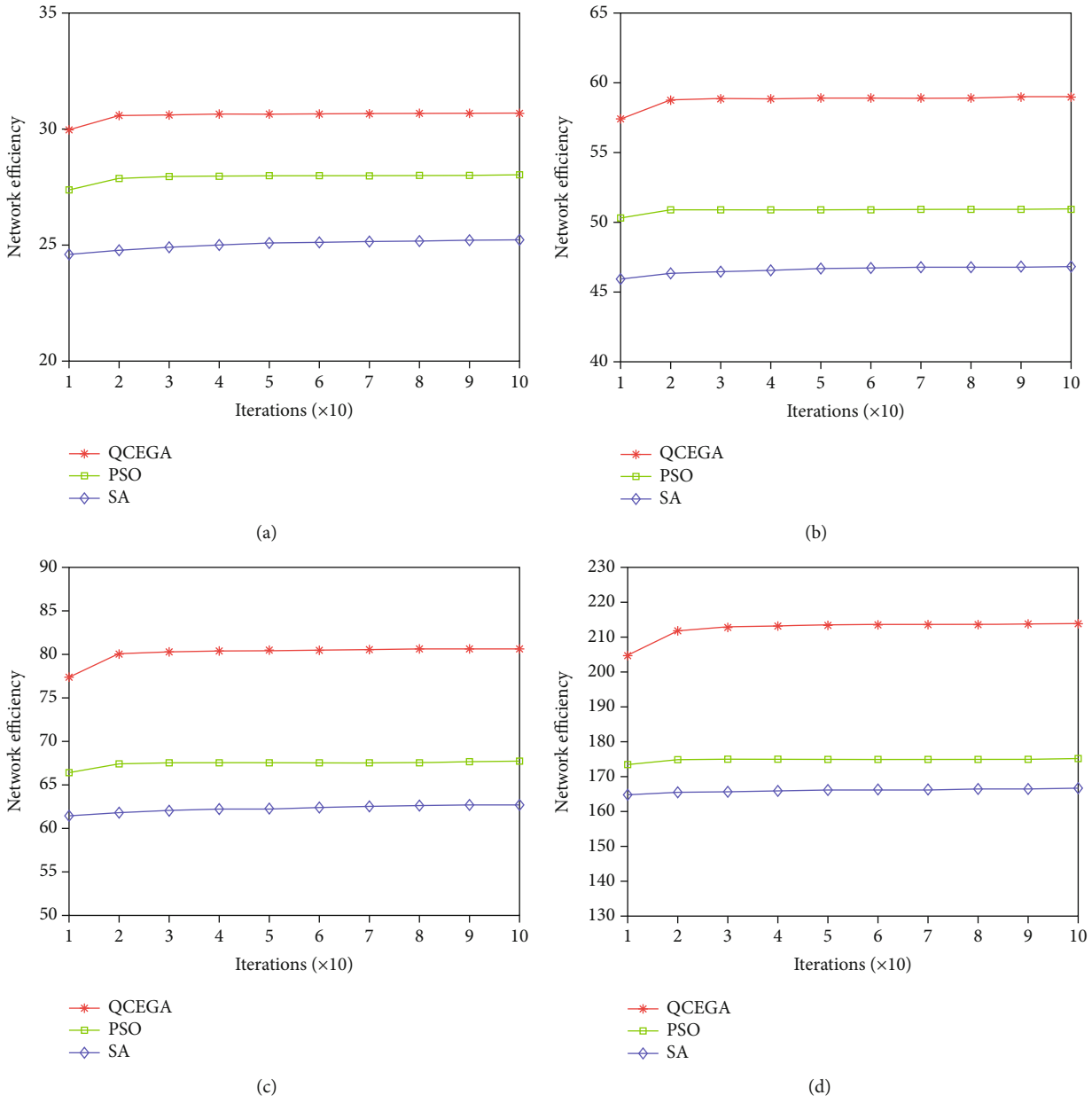


Figure 6: The line chart of the three algorithms compared every ten generations: (a) 30 target points, 6 soil depths, and 3 sensors per target point; (b) 40 target points, 8 soil depths, and 4 sensors per target point; (c) 50 target points, 10 soil depths, and 5 sensors per target point; (d) 100 target points, 12 soil depths, and 6 sensors per target point.

network efficiency of QCEGA is higher than that of SA and PSO under the situation that the target points are 20, 25, and 50 and the number of soil depths is 8, 10, and 12. The average network efficiency of QCEGA does not converge prematurely, which is a common phenomenon of SA and PSO. In general, QCEGA has a faster convergence rate than SA and PSO under the same number of iterations.

Figure 5 shows the number of iterations required for the three algorithms to achieve convergence under different simulation conditions. The standard of convergence is the range fluctuation percentage judgment method. The fluctuation range defined in this experiment is 1% to 2%; that is to say, if the increase percentage of the network efficiency obtained by the algorithm is between 1% and 2%, it can be considered convergent. In Figure 5 a, we can find that the convergence iterations required by QCEGA only require 18 generations, while SA requires the most convergence iterations, which are 32 generations. In Figure 5 a–c, the convergence performance of PSO is between SA and QCEGA, but in Figure 5 d, the number of iterations for SA to reach convergence is 26 generations, which is lower than 28 iterations of PSO. But at this time SA fell into premature convergence. In Figure 5 a–d, the number of iterations required for QCEGA to reach convergence is few, which can indicate that it has good convergence performance.

Table 2 shows the percentage increase of the solutions obtained by the three algorithms compared to the unoptimized scheme. In 5 different situations, QCEGA's

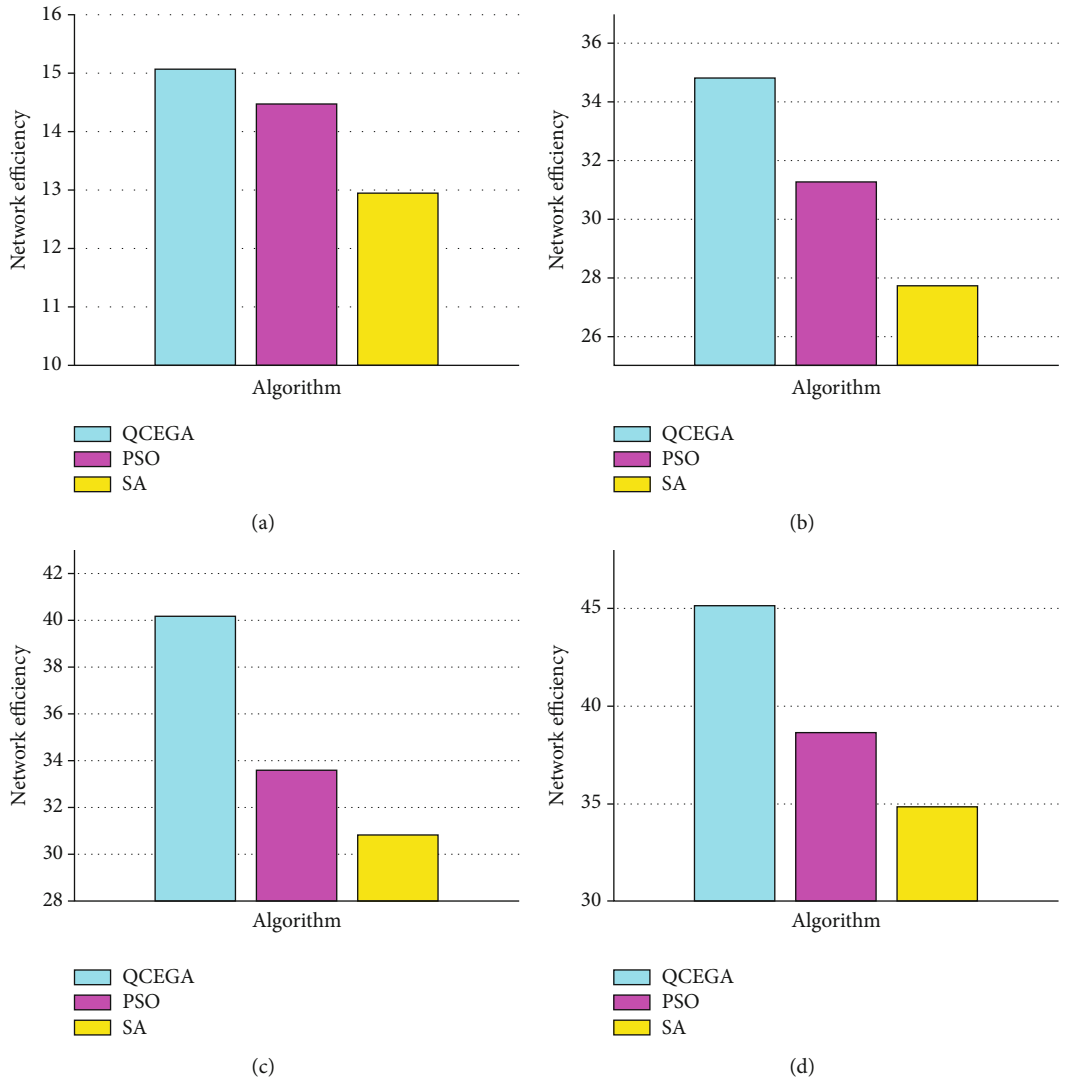


Figure 7: Comparison of the solution quality of three algorithms: (a) 20 target points, 5 soil depths, and 2 sensors per target point; (b) 25 target points, 7 soil depths, and 3 sensors per target point; (c) 30 target points, 10 soil depths, and 4 sensors per target point; (d) 35 target points, 12 soil depths, and 5 sensors per target point.

improvement percentage is the highest among the three algorithms, which are 22.72%, 23.30%, 29.97%, 31.29%, and 31.67%, respectively, compared to before optimization. The worst performance among the three algorithms is in SA. This is because it falls into the local optimum early and has a weak ability to jump out of it. PSO also has the weakness of falling into local convergence. Neither PSO nor SA can obtain the optimal solution to the SMWSN target assignment problem. Therefore, it can be seen that QCEGA can effectively improve the sensor target allocation scheme in SMWSNs.

In the line chart of Figure 6, the performance of QCEGA, SA, and PSO is compared in the form of one point every 10 generations. In Figure 6(a), when the conditions are 30 target points, 6 soil depths, and 3 sensors per target point, QCEGA obtains higher network efficiency than SA and PSO at the first point. Since then, the network efficiency value obtained based on QCEGA has been higher than that of PSO and SA. In addition, the network efficiency value of SA is lower than that of PSO, but both PSO and SA fall into the local optimum. In Figures 6(b)–6(d), similar conclusions can be drawn. In general, in solving the problem of SMWSN target allocation, QCEGA can jump out of the local optimum for obtaining the optimal target allocation scheme.

Two sensors, 3 sensors, 4 sensors, and 5 sensors are placed at each target point in Figure 7. The fitness function used to calculate the network efficiency can be obtained from formula (9). In Figure 7(a), the network efficiency value is improved after placing 2 sensors at each target point and 100 iterations. The result suggests that the effect of QCEGA is better than that of PSO and the effect of PSO is better than that of SA. In Figures 7(b)–7(d), when 3, 4, and 5 sensors are placed at each target point, comparable results are given. QCEGA always has a better performance than SA and PSO under the condition of placing a different number of sensors at each target point.

It can be seen from Table 3 that in the specific implementation process, the algorithm complexity of QCEGA and SA is all $O(m * n^2)$ and the algorithm complexity of PSO is

Table 3: The computational complexity of the three algorithms.

Algorithm	QCEGA	PSO	SA
Complexity	$O(m * n^2)$	$O(m * (n^2 + n))$	$O(m * n^2)$

Table 4: Comparison of algorithm running time (s).

	QCEGA	PSO	SA
18 target points 7 soil depths 3 sensors per target point	3.34	7.00	4.37
24 target points 8 soil depths 4 sensors per target point	4.14	10.51	6.09
30 target points 10 soil depths 4 sensors per target point	5.13	12.30	7.17
36 target points 12 soil depths 5 sensors per target point	6.34	16.72	9.31
40 target points 15 soil depths 6 sensors per target point	7.79	21.17	11.51

$O(m * (n^2 + n))$, where m is the number of outer cycles and n is the number of individuals in the population. However, because PSO needs to continuously update the velocity and position of particles, SA, which is not efficient in solving the problem, has been continuously looping within the group. As a result, the actual complexity of PSO and SA in solving the SMWSN target allocation problem is higher than that of QCEGA. Although QCEGA requires quantum probability amplitude update and cloning, these operations are simple double loops. In addition, it can be seen from Figures 5 and 7 that the convergence performance and solution quality of QCEGA are better than those of PSO and SA. To further compare the complexity of the three algorithms, the running time of the algorithms is given in Table 4.

According to the running time of the three algorithms in Table 4, it can be further proven that the complexity of QCEGA is lower than that of PSO and SA. When the simulation conditions are 18 target points, 7 soil depths, and 3 sensors per target point, the running time of QCEGA only takes 3.34 s, the running time of SA takes 4.37 s, and the running time of PSO is the longest, reaching 7.00 s. As the scale of the problem continues to expand, the running time of PSO and SA has increased more than that of QCEGA. With the simulation conditions of 40 target points, 15 soil depths, and 6 sensors per target point, the running time of QCEGA is 63.20% lower than that of PSO and 32.31% lower than that of SA. Therefore, QCEGA has excellent running performance.

6. Conclusions

In conclusion, this paper puts forward an evaluation mechanism and a quantum clone elite genetic algorithm (QCEGA) for the allocation of sensors in SMWSNs. From the perspective of soil irrigation, this paper makes a mathematical model for the agricultural scenario and then uses the proposed algorithm to solve the mathematical model. The results show that it is effective in solving the problem of target allocation. By introducing QCEGA into the sensor target allocation problem, the results prove the effectiveness of the QCEGA-based target allocation in optimizing network efficiency of the SMWSNs. By applying QCEGA, a more efficient irrigation scheme can be obtained when land irrigation is carried out. Not only does the scheme obtain the maximum monitoring effect of the soil moisture sensor but also it saves water, which is especially useful in some places where water resources are relatively scarce. Through the suitable deployment of sensors, limited resources can be used as much as possible.

Future research should carefully consider other methods that can further improve the network efficiency of SMWSNs, such as combining routing optimization algorithms, clustering SMWSNs, and machine learning techniques.

Data Availability

The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This paper was funded by the project of Youth and Middle-Aged Scientific and Technological Innovation Leading Talents Program of the Corps (grant number 2018CB006), the China Postdoctoral Science Foundation (grant number 220531), the Corps Innovative Talents Plan (grant number 2020CB001), the Funding Project for High Level Talents Research in Shihezi University (grant number RCZK2018C38), and the Project of Shihezi University (grant number ZZZC201915B).

References

[1] L. Incrocci, R. B. Thompson, M. D. Fernandez-Fernandez et al., "Irrigation management of European greenhouse vegetable crops," *Agricultural Water Management*, vol. 242, p. 106393, 2020.

[2] A. Abera, N. E. C. Verhoest, S. Tilahun, H. Inyang, and J. Nyssen, "Assessment of irrigation expansion and implications for water resources by using RS and GIS techniques in the Lake Tana Basin of Ethiopia," *Environmental Monitoring and Assessment*, vol. 193, no. 1, p. 13, 2020.

[3] I. Abd-Elaty, L. Pugliese, M. Zelenakova, P. Mesaros, and A. E. Shinawi, "Simulation-based solutions reducing soil and groundwater contamination from fertilizers in arid and semi-

arid regions: case study the Eastern Nile Delta, Egypt," *International Journal of Environmental Research and Public Health*, vol. 17, no. 24, p. 9373, 2020.

[4] P. Bamurigire, A. Vodacek, A. Valko, and S. Rutabayiro Ngoga, "Simulation of Internet of things water management for efficient rice irrigation in Rwanda," *Agriculture-Basel*, vol. 10, no. 10, p. 431, 2020.

[5] J. Y. Xue, Z. Huo, S. Wang et al., "A novel regional irrigation water productivity model coupling irrigation- and drainage-driven soil hydrology and salinity dynamics and shallow groundwater movement in arid regions in China," *Hydrology and Earth System Sciences*, vol. 24, no. 5, pp. 2399–2418, 2020.

[6] G. Javvaji and S. K. Udgata, "Soft computing approach for multi-objective task allocation problem in wireless sensor network," *Evolutionary Intelligence*, pp. 1–13, 2020.

[7] E. Bonnah, S. Ju, and W. Cai, "Coverage maximization in wireless sensor networks using minimal exposure path and particle swarm optimization," *Sensing and Imaging*, vol. 21, no. 1, 2020.

[8] N. Thi My Binh, A. Mellouk, H. Thi Thanh Binh, L. Vu Loi, D. Lam San, and T. Hai Anh, "An elite hybrid particle swarm optimization for solving minimal exposure path problem in mobile wireless sensor networks," *Sensors*, vol. 20, no. 9, p. 2586, 2020.

[9] L. Nagarajan and S. Thangavelu, "Hybrid grey wolf sunflower optimisation algorithm for energy-efficient cluster head selection in wireless sensor networks for lifetime enhancement," *IET Communications*, vol. 15, no. 3, pp. 384–396, 2021.

[10] D. Hosahalli and K. G. Srinivas, "Enhanced reinforcement learning assisted dynamic power management model for Internet-of-things centric wireless sensor network," *IET Communications*, vol. 14, no. 21, pp. 3748–3760, 2020.

[11] J. Yu, Z. Yu, M. Ding, and W. Ye, "Research on the tenacity survivability of wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 9, pp. 3535–3544, 2020.

[12] M. Tang, Y. Xin, and Y. Qiao, "Multi-objective resource allocation algorithm for wireless sensor network based on improved simulated annealing," *Adhoc & Sensor Wireless Networks*, vol. 47, no. 1-4, pp. 157–173, 2020.

[13] Y. Zhang, "Coverage optimization and simulation of wireless sensor networks based on particle swarm optimization," *International Journal of Wireless Information Networks*, vol. 27, no. 4, 2019.

[14] C. Gong, C. Guo, H. Xu, C. Zhou, and X. Yuan, "A joint optimization strategy of coverage planning and energy scheduling for wireless rechargeable sensor networks," *Processes*, vol. 8, no. 10, p. 1324, 2020.

[15] Z. Hao, N. Qu, X. Dang, and J. Hou, "Node optimization coverage method under link model in passive monitoring system of three-dimensional wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 15, no. 8, Article ID 155014771986987, 2019.

[16] Y. Cao, Y. Zhao, and F. Dai, "Node localization in wireless sensor networks based on quantum annealing algorithm and edge computing," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Atlanta, GA, USA, 2019.

[17] H. Zaineldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, "An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, 2020.

[18] E. Zanaj, E. Gambi, B. Zanaj, and D. Disha, *Customizable hierarchical wireless sensor networks based on genetic algorithm*.

[19] S. Kumar, O. Kaiwartya, and A. H. Abdullah, "Green computing for wireless sensor networks: optimization and Huffman coding approach," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 592–609, 2017.

[20] A. Bansal, O. Kaiwartya, R. K. Singh, and S. Prakash, "Maximizing fault tolerance and minimizing delay in virtual network embedding using NSGA-II," in *Proceeding of the Third International Symposium on Women in Computing and Informatics*, pp. 124–130, New York, NY, USA, August 2015.

[21] W. L. Wu, X. B. Wen, H. X. Xu, L. M. Yuan, and Q. X. Meng, "Accurate range-free localization based on quantum particle swarm optimization in heterogeneous wireless sensor networks," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 3, pp. 1083–1097, 2018.

[22] P. Kanchan and S. D. Pushparaj, "A quantum inspired PSO algorithm for energy efficient clustering in wireless sensor networks," *Cogent Engineering*, vol. 5, no. 1, pp. 1–16, 2018.

[23] M. S. Prasad and T. Panigrahi, "Distributed maximum likelihood DOA estimation algorithm for correlated signals in wireless sensor network," *Wireless Personal Communications*, vol. 105, no. 4, pp. 1527–1544, 2019.

[24] Y. Z. Li, Y. Zhao, and Y. Y. Zhang, "A spanning tree construction algorithm for industrial wireless sensor networks based on quantum artificial bee colony," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, Article ID 176, 2019.

[25] S. C. Chu, Z. G. Du, and J. S. Pan, "Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks," *Applied Sciences*, vol. 10, no. 3, p. 19, 2020.

[26] A. Niccolai, F. Grimaccia, M. Mussetta, and R. Zich, "Optimal task allocation in wireless sensor networks by means of social network optimization," *Mathematics*, vol. 7, no. 4, p. 315, 2019.

[27] X. Yin, K. Zhang, B. Li, A. K. Sangaiah, and J. Wang, "A task allocation strategy for complex applications in heterogeneous cluster–based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, Article ID 155014771879535, 2018.

[28] I. Hilali-Jaghdam, A. Ben Ishak, S. Abdel-Khalek, and A. Jamal, "Quantum and classical genetic algorithms for multilevel segmentation of medical images: a comparative study," *Computer Communications*, vol. 162, pp. 83–93, 2020.

[29] T.-C. Chang, K. M. Haupfear, J. Yu et al., "A novel algorithm comprehensively characterizes human RH genes using whole-genome sequencing data," *Blood Advances*, vol. 4, no. 18, pp. 4347–4357, 2020.

[30] I. Araya, M. Moyano, and C. Sanchez, "A beam search algorithm for the biobjective container loading problem," *European Journal of Operational Research*, vol. 286, no. 2, pp. 417–431, 2020.

[31] A. Krouska and M. Virvou, "An enhanced genetic algorithm for heterogeneous group formation based on multi-characteristics in social-networking-based learning," *Ieee*

Transactions on Learning Technologies, vol. 13, no. 3, pp. 465–476, 2020.

[32] D. Wang, H. Chen, T. Li, J. Wan, and Y. Huang, "A novel quantum grasshopper optimization algorithm for feature selection," *International Journal of Approximate Reasoning*, vol. 127, pp. 33–53, 2020.

[33] A. Kaveh, M. Kamalinejad, and H. Arzani, "Quantum evolutionary algorithm hybridized with enhanced colliding bodies for optimization," *Structures*, vol. 28, pp. 1479–1501, 2020.

[34] Y. Sun, Y. Gao, and X. Shi, "Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity," *Mathematics*, vol. 7, no. 2, p. 146, 2019.

[35] Q. Li, Z. Cao, W. Ding, and Q. Li, "A multi-objective adaptive evolutionary algorithm to extract communities in networks," *Swarm and Evolutionary Computation*, vol. 52, article 100629, 2020.