



Research article

Improved beluga whale optimization algorithm based cluster routing in wireless sensor networks

Hao Yuan¹, Qiang Chen^{1,*}, Hongbing Li^{2,3,*}, Die Zeng¹, Tianwen Wu¹, Yuning Wang² and Wei Zhang^{2,3}

- ¹ Chongqing Key Laboratory of Geological Environmental Monitoring and Disaster Early Warning in the Three Gorges Reservoir Area, Chongqing Three Gorges University, Chongqing 404120, China
- ² Internet of Things and Intelligent Control Technology Chongqing Engineering Research Center, Chongqing Three Gorges University, Chongqing 404120, China
- ³ Chongqing Municipal Key Laboratory of Intelligent Information Processing and Control, Chongqing Three Gorges University, Chongqing 404120, China

* **Correspondence:** Email: qiangchen@sanxiau.edu.cn, sxxyllhb@163.com.

Abstract: Cluster routing is a critical routing approach in wireless sensor networks (WSNs). However, the uneven distribution of selected cluster head nodes and impractical data transmission paths can result in uneven depletion of network energy. For this purpose, we introduce a new routing strategy for clustered wireless sensor networks that utilizes an improved beluga whale optimization algorithm, called tCBWO-DPR. In the selection process of cluster heads, we introduce a new excitation function to evaluate and select more suitable candidate cluster heads by establishing the correlation between the energy of node and the positional relationship of nodes. In addition, the beluga whale optimization (BWO) algorithm has been improved by incorporating the cosine factor and t-distribution to enhance its local and global search capabilities, as well as to improve its convergence speed and ability. For the data transmission path, we use Prim's algorithm to construct a spanning tree and introduce DPR for determining the optimal route between cluster heads based on the correlation distances of cluster heads. This effectively shortens the data transmission path and enhances network stability. Simulation results show that the improved beluga whale optimization based algorithm can effectively improve the survival cycle and reduce the average energy consumption of the network.

Keywords: wireless sensor networks; beluga whale optimization; hierarchical routing; minimum spanning tree; heuristic algorithm

1. Introduction

The Internet of Things (IoT) technology enables intelligent identification, positioning, tracking, and supervision of various devices by connecting them [1]. Wireless sensor networks (WSNs) are a critical IoT technology consisting of a multitude of miniature sensor nodes primarily used for environmental monitoring and related applications [2–5]. These nodes autonomously form a network, enabling comprehensive and efficient environmental monitoring [6].

The key issue for wireless sensor networks is reducing network energy consumption to improve the lifetime of the network. Due to their tiny size and long-term deployment in complex environments, wireless sensor nodes are typically powered by using batteries. Currently, WSNs are usually grouped by hierarchical routing to improve the network survival cycle. Hierarchical routing is represented by Low Energy Adaptive Clustering Hierarchy (LEACH) [7], the main idea of which is to reduce the energy consumption of data transmission by dividing sensor nodes into multiple clusters and forwarding data to the base station through cluster heads within clusters to achieve data transmission in clusters to improve the network life cycle.

Cluster routing is the most commonly used routing method to achieve energy efficiency in wireless sensor networks in recent years. However, optimal determination of cluster heads (CHs) is a challenging task due to dynamic changes in network conditions [8]. In practice, this has proven to be a multi-policy optimization problem [9]. For solving multi-strategy problems, the objective function is usually established according to the specific problem, and its extremes are solved using meta-heuristic algorithms, e.g., the 4PL problem [10], the localization routing problem [11], etc. Among these methods, meta-heuristics are often used to select the appropriate CHs during the setup phase of each round [12]. However, these techniques run iterative algorithms at the beginning of each round, which takes a long time, and meta-heuristics are not stable. Instead, heuristics are problem-dependent algorithms that can produce viable solutions in a reasonable amount of time. However, adjusting multi-criteria heuristics that incorporate different features of the network can be a daunting task, so more appropriate methods are needed to solve this problem.

In WSN, node energy and location will have an important impact on network lifetime and stability. In the network clustering stage, due to the interaction and complexity between nodes, the algorithm faces a huge computational burden and a high degree of uncertainty, which belongs to the category of NP (nondeterministic polynomial) problems [13]. A meta-heuristic algorithm is a key means to solve NP problems. The latest heuristic algorithm launched in 2022 is the beluga whale optimization (BWO) algorithm, which promises robust performance [14]. The advantages of this algorithm are that the convergence speed is fast and the results are relatively stable. It can be widely used in various scenarios.

Inter-cluster routing involves the transmission and forwarding of data between multiple clusters [15]. Its routing problem has high complexity and computational complexity. To improve routing efficiency, heuristic algorithms select paths and nodes in a targeted manner by introducing heuristic information and empirical rules into the search process. Select paths and nodes in a targeted manner, reducing the amount of computation and solving complex routing problems. The Prim heuristic algorithm is more efficient and scalable when dealing with large-scale networks than precise algorithms.

Address the limitations of a single meta-heuristic algorithm and heuristic algorithm when it comes to hierarchical routing in wireless sensor networks. We propose a new hybrid algorithm that combines the advantages of the two algorithms. By improving the stability of the BWO algorithm, combined with

the WSN transmission energy consumption model, the Prim algorithm is improved, referred to as the tCBWO-DPR algorithm. The algorithm is capable of providing high-quality data transmission services in each round, which can significantly improve the overall performance of the network.

The main contributions of this study are summarized as follows:

- The beluga whale optimization (BWO) is improved by the student's t-distribution and cosine boundary policy, which we call tCBWO. The BWO algorithm is in the local development stage most of the time during the execution of the algorithm. By improving the cosine boundary policy BWO, the algorithm can carry out a wider range of exploration in the initial stage, and use the t-distribution to enrich the diversity of the population. The stability and convergence properties of the tCBWO algorithm are verified by CEC 2017.
- The DPR heuristic algorithm is proposed to be applied to WSNs. Combined with the characteristics of WSNs, the improved prim algorithm is used for inter-cluster transmission. According to the radio sensor model, by optimizing the prim algorithm, the problem that neighboring base station nodes will become aggregation nodes is optimized, which effectively improves the stability of the network.
- A new hierarchical routing scheme for WSNs has been proposed. This method combines the advantages of meta-heuristic algorithms and heuristic algorithms. The stability and convergence performance of tCBWO are used to select the cluster head, and the inter-cluster transmission path is established by the DPR algorithm. The feasibility of the conjecture is verified by ablation and comparison experiments.

The structure of the article is as follows: Section 2 describes the recent research progress. Section 3 describes the WSNs network model. Section 4 describes the tCBWO process. Section 5 introduces the implementation steps of the tCBWO-DPR algorithm. Section 6 includes the performance analysis of the tCBWO and tCBWO-DPR algorithms, while Section 7 summarizes the findings presented in this work.

2. Related work

Wireless sensor networks usually adopt a hierarchical routing approach to divide sensor nodes into multiple clusters. In each cluster, the cluster head is responsible for collecting and forwarding data from the sensor nodes to the base station, thus reducing the energy consumption of data transmission and improving the network lifecycle. This cluster-based data transmission method effectively optimizes the energy utilization efficiency of wireless sensor networks and achieves efficient and reliable data transmission.

In traditional hierarchical routing, cluster head nodes carry higher network energy consumption, which affects network stability. To solve this problem, existing methods usually select nodes closer to the base station as cluster head nodes, but due to the small number of such nodes, the adjacent nodes are overburdened. In small-scale hierarchical routing, nodes are often used for single-hop transmission, but the communication cost rises sharply when the distance from the base station is too far. Therefore, how to select suitable cluster head nodes and establish efficient data transmission paths is an important problem to be solved in WSNs.

In order to solve the above problems, numerous scholars have proposed new approaches. Current approaches to the above problem mainly include: heuristic algorithms, meta-heuristic algorithms, machine learning, and hybrid algorithms.

2.1. Heuristic algorithm

LEACH is a hierarchical routing protocol that was first proposed by Heinzelman et al. in 2000 [16]. It uses a heuristic algorithm to divide the WSNs into clusters for data exchange and collaboration between nodes. By dividing nodes into clusters, the LEACH protocol reduces the communication distance between nodes, reduces energy consumption, and improves the scalability and balance of the network.

LEACH-C is a centrally clustered routing algorithm based on LEACH proposed by Heinzelman et al. in 2002 [17]. This protocol starts with a phase where all nodes send their location information and remaining energy values to the base station. After receiving this information, the base station calculates the average energy value of all nodes and selects nodes whose energy is not lower than average value as a candidate node. By implementing energy-aware mechanisms, the energy efficiency and reliability of the network can be effectively improved, and the network lifetime can be extended.

Shahraki et al. [18] proposed the energy-aware routing algorithm (ERA). This algorithm combines parameters such as historical activity records of sensor nodes, local and global states of nodes to select cluster headsets, and intra-cluster members. Compared with the classical algorithm, this algorithm has improved both network stability and network lifetime.

Due to the simple execution steps of the heuristic algorithm, it can quickly obtain feasible solutions to various NP problems and is widely used in WSNs. However, due to its susceptibility to local optima and its potential for lengthy computations, traditional optimization techniques have been replaced by meta-heuristics that offer stronger convergence performance.

2.2. Meta-heuristic algorithm

Pitchaimanickam et al. [19] proposed HFAPSO, a clustered routing algorithm based on LEACH-C in a centralized manner. The algorithm uses a hybrid firefly algorithm and particle swarm optimization (HFAPSO) to select the best cluster head while considering the residual energy and the distance between the cluster head and the cluster members. The algorithm effectively improves the life cycle, residual energy, number of survival nodes, and throughput of wireless sensor networks.

Yu et al. [20] proposed a cluster routing algorithm based on a hybrid genetic taboo search (CRGT). This algorithm introduces two parameters, node residual energy, and node distance to the Sink node, in the cluster head election phase to optimize the threshold function for a more reasonable selection of cluster head nodes. In the clustering phase, ordinary nodes join the least costly cluster according to the cost function to balance the energy consumption of network nodes. During the data transmission, the algorithm uses a hybrid genetic forbidden search algorithm to select the optimal path with the lowest energy consumption. Compared with other algorithms, this algorithm improves both network stability and network lifetime.

Guo et al. [21] proposed a WSN clustered routing algorithm combining the sine cosine algorithm and the Lévy variant (SCA-Lévy). This algorithm constructs a new fitness function in the cluster head election phase by considering factors such as the location relationship between nodes and node energy. A sine cosine algorithm with an improved step factor is used to select the cluster head nodes, and Lévy

variation is introduced to increase the diversity of the population and the search space, thus improving the performance of the algorithm. This effectively extends the life cycle of the network and can well balance the load of the network nodes.

A meta-heuristic algorithm can comprehensively consider the position relationship of network nodes and the change of node energy, which can effectively improve the equilibrium of the network. However, the current meta-heuristic algorithms generally have poor stability and a large number of calculations, which will also have a certain impact on the stability and life cycle of the network.

2.3. Machine learning

Luyao et al. [22] proposed a density peak clustering algorithm called MKN-DPC, which is based on the mutual k-nearest neighbor method. The algorithm takes into account the local characteristics of the network node distribution, dynamically calculates the local density of each data point, and combines the data point density and signal strength to automatically select the initial cluster heads and assign the remaining nodes. Finally, the algorithm uses a merging strategy to combine mergeable clusters to prevent the clusters from being divided excessively.

Esmaeili et al. [23] proposed a combined algorithm based on metaheuristics-machine learning techniques (CMML). The algorithm trains and tests a set of wireless sensor networks offline using heuristic algorithms and evaluates the performance of the trained models. The trained CMML model is then applied to generate an online clustering solution to generate a suitable set of cluster heads. The algorithm effectively improves the survival time of the network and improves the stability of the network.

Debasis et al. [24] proposed an energy efficient clustering algorithm (EECA). In EECA, the network is divided into several different regions and one node is selected as the cluster head (CH) in each region by an artificial neural network (ANN). The ANN calculates the scores of these nodes on the basis of four parameters, such as residual energy, number of detected events, distance to the base station, and number of neighbors so as to select the sensor node with the highest score as a CH in its region. To avoid the formation of huge clusters, a maximum cluster size limit is also defined. Only sensor nodes close to the event transmit data to the CH, ensuring that redundant data is not transmitted to the CH. This scheme reduces the transmission of repetitive and redundant data and improves the network lifetime to some extent.

In machine learning algorithms, hierarchical routing can be performed in constant time and effectively improve the life cycle of the network. The amount of computation will increase dramatically as the number of nodes grows, which will waste a lot of limited computing resources, which is not good for WSNs.

2.4. Hybrid algorithms

Zeynab et al. [25] proposed a fuzzy routing protocol based on swarm intelligence (SIF). The algorithm first uses a fuzzy c-mean clustering algorithm to divide all sensor nodes in the network into uniformly distributed clusters, and then uses the Mamdani fuzzy inference system to select the appropriate cluster heads to overcome the uncertainty in WSNs. Then, the fuzzy rule base table of SIF is optimized by a hybrid population intelligence algorithm using firefly algorithm and simulated annealing (FA-SA algorithm) to extend the life cycle of the network. This approach can improve the survival time of WSNs more effectively.

Radhika et al. [26] proposed an energy-aware clustering via a fuzzy unequal model (ECFU). The model constructs sensor clusters in the clustering phase by using the energy and location relationships of sensors as input to a fuzzy inference system (FIS). After forming clusters, the data transmission is reduced by using a mechanism that minimizes the total number of transmitted packets within the network. The data transmission is done by detecting similar data from different neighboring nodes using the mechanism of minimizing the total number of transmitted packets within the network to reduce the total amount of data transmission.

Mehra et al. [27] proposed a fuzzy logic based balanced cost cluster head (CH) selection algorithm called fuzzy based enhanced cluster head selection (FBECS). The algorithm fully considers the residual energy of the sensor nodes, the distance from the off-convergence point, and the density of nearby nodes as inputs, and utilizes a fuzzy inference system to perform a comprehensive evaluation. The FBECS algorithm determines the optimal cluster head (CH) by calculating an index of eligibility for each sensor node to be selected as a CH, which in turn identifies the optimal CH by taking into account the probability of assigning the CH to each node. The goal of this strategy is to ensure a balanced distribution of load throughout the network. By utilizing probabilistic allocation to select the best candidate for the cluster coordinator role, it can effectively improve the energy utilization efficiency of the system and further enhance the load balance of the network.

Hybrid algorithms have been the main research direction in recent years. They can integrate the advantages of multiple algorithms, compensate for their shortcomings, and leverage each other's strengths through learning. This paper combines the advantages of heuristics and meta-heuristic algorithms. Meta-heuristic algorithms are used to implement the cluster head selection process with large data scales to improve accuracy. In inter-cluster communication, heuristics are used to reduce computing costs and improve efficiency.

3. Model declaration

3.1. Network model

In a wireless sensor network, sensor nodes are usually deployed in a designated detection area to communicate between devices and forward data to a base station through a self-organizing network. The base station transmits the received data to the center through the internet. Since data transmission consumes more energy from sensor nodes at longer distances, it leads to a reduction in the lifetime of the network. The hierarchical routing model is shown in Figure 1. Hierarchical routing achieves balanced energy consumption to extend the network lifetime by dividing the nodes in the monitoring area into clusters and forwarding the information from the nodes within the clusters to the base station through the cluster head node.

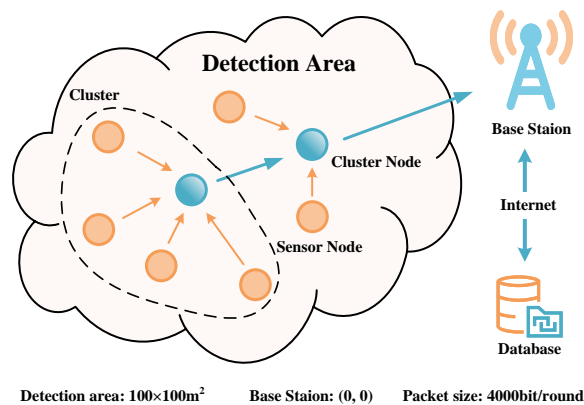


Figure 1. WSNs clustering diagram.

3.2. Radio transmission model

In the operation of wireless sensors, the energy consumed for transmission is about 3000 times more than when processing data of the same size [28]. To facilitate subsequent calculations, we only consider the energy consumption for data transmission. During data transmission, the radio cell directly consumes the energy of the sensor. The radio transmission unit is divided into a data transmitting unit and a data receiving unit. In the data-sending cell, the transmission energy consumption is positively related to the transmission distance, in accordance with the first-order radio transmission model. The energy consumption of the transmission circuit can be expressed by

$$E_{TX}(k, d) = \begin{cases} kE_{elec} + kE_{fs} \times d^2, & d < d_0 \\ kE_{elec} + kE_{mp} \times d^4, & d \geq d_0, \end{cases} \quad (3.1)$$

where E_{elec} is the energy consumed per bit of data processed by the transmitter circuit, E_{mp} represents the energy required by multi-path, and E_{fs} is the amplification energy for free space. The parameter k represents transmitted packet size, and d represents the maximum distance that messages can be transmitted, for which the value is $d_0 = \sqrt{E_{fs}/E_{mp}}$.

Transmission energy consumption of the receiving cell is proportional to the received packet size, and its energy consumption can be expressed by

$$E_{RX}(k) = kE_{elec}, \quad (3.2)$$

where E_{elec} is the amount of energy consumed to process each bit of the data in the receiving circuit. The parameter k is the size of the received packets.

4. tCBWO: improved BWO

4.1. Beluga whale optimization

The beluga whale optimization (BWO) algorithm is a new heuristic algorithm proposed in 2022. The algorithm is a novel algorithm formed by simulating the behavior of beluga whales. The algorithm collects numerous algorithmic advantages. The algorithm can be widely applied to unimodal and multimodal optimization problems by adapting parameters to reduce the amount of parameter control. The algorithm is divided into three main parts: the global exploration phase, the local development phase and the fall phase.

4.1.1. Initialization phase

At the beginning of the algorithm execution, the population of beluga whales is generated by randomization. The global search capability in the solution space is guaranteed. The development phase, on the other hand, controls the local search capability in the solution space. In combination with the whale fall probability, the current position of the whale can be changed. The beluga whale optimization algorithm initially generates an initial population of whales X at random:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}, \quad (4.1)$$

where n is the population size of beluga whales, and d is the dimensionality of the variable space. For all beluga whales, the corresponding fitness values are shown in Eq (4.2).

$$F_X = \begin{bmatrix} f(x_{1,1}, x_{1,2}, \cdots, x_{1,d}) \\ f(x_{2,1}, x_{2,2}, \cdots, x_{2,d}) \\ \vdots \\ f(x_{n,1}, x_{n,2}, \cdots, x_{n,d}) \end{bmatrix}. \quad (4.2)$$

Whales enter a global exploration phase or a localized exploitation phase depending on the equilibrium factor B_f . Equation (4.3) is the expression of the equilibrium factor B_f .

$$B_f = B_0 \left(1 - \frac{T}{2T_{max}}\right), \quad (4.3)$$

where T is the number of current iterations, and T_{max} is the maximum number of iterations. B_0 is the random number in the interval (0, 1). When $B_f > 0.5$, the whale enters into the global development phase. Otherwise, it enters into the local development phase.

4.1.2. Global exploration phase

In the global exploration phase, BWO integrates the swimming behavior of beluga whales in different postures by synchronization or mirroring. According to its characteristics, the BWO algorithm adopts the position update strategy in Eq (4.4) during the global search.

$$\begin{cases} x_{i,j}^{T+1} = x_{i,p_j}^T + (x_{r,p_1}^T - x_{i,p_j}^T)(1 + r_1) \sin(2\pi r_2), j = \text{even} \\ x_{i,j}^{T+1} = x_{i,p_j}^T + (x_{r,p_1}^T - x_{i,p_j}^T)(1 + r_1) \cos(2\pi r_2), j = \text{odd}, \end{cases} \quad (4.4)$$

where $x_{i,j}^{T+1}$ represents the new coordinates of the i^{th} beluga in the j^{th} dimension, and it is a random number chosen in the d^{th} dimension. r_1 and r_1 are random numbers in the interval [0, 1].

4.1.3. Partial development phase

The development phase of BWO was inspired by the feeding behavior of beluga whales. Beluga whales can move to cooperate in foraging based on the location of nearby beluga during foraging. Thus,

beluga whales hunt by sharing each other's location information. In the development phase of BWO, the Lévy flight strategy is used to improve the convergence performance of the algorithm (by assuming that belugas can capture prey with the Lévy flight strategy). The beluga whale development process is represented in the algorithm by

$$x_i^{T+1} = r_3 x_{best}^T - r_4 x_i^T + C_1 L_F(x_r^T - x_i^T), \quad (4.5)$$

where x_i^{T+1} denotes the next generation individual of the i^{th} solution, x_{best}^T denotes the current optimal solution, x_r^T is the randomly selected solution, x_i^T denotes the i^{th} solution, r_3 and r_4 are random numbers in the interval $[0, 1]$, and C_1 is the adaptive equilibrium factor, the BWO set $C_1 = 2r_4(1 - T/T_{max})$. L_F denotes the Lévy flight factor.

4.1.4. Whale fall phase

During beluga migration and foraging, other natural predators disturb them, which can cause their location to change. This disturbance can cause belugas to fall, and these fallen belugas can attract other organisms to reproduce here. To simulate this whale-fall behavior, the BWO algorithm randomly selects some belugas to fall with a certain probability during the iterative process to simulate small changes in the population. In the BWO algorithm, it is set that when B_f is smaller than W_f , the algorithm enters the whale fall phase. The fall probability of the whale population is $W_f = 0.1 - 0.05 \frac{T}{T_{max}}$. The location of the whale fall is set in BWO as shown in Eq (4.6).

$$X_i^{T+1} = r_5 X_i^T - r_6 X_r^T + r_7 X_{step}, \quad (4.6)$$

where r_5 , r_6 , and r_7 are random numbers in the interval $(0, 1)$. X_{step} is the step size of the beluga fall, and its value is related to the number of iterations, which is calculated as

$$X_{step} = (ub - lb) \exp(-C_2 \frac{T}{T_{max}}), \quad (4.7)$$

where X_{step} is the step factor between the whale population fall probability and population size, the BWO set $C_2 = 2nW_f$. ub and lb represent the upper and lower bounds of the variable space, respectively.

4.2. *tCBWO: t cosine beluga whales optimization*

The standard BWO algorithm determines the exploitation state by establishing linear boundary conditions in the initial stage. As can be seen from Eq (4.3), the global exploration phase and local exploitation state of the original BWO algorithm do not have a uniform proportion, and most of the time is in the exploitation phase, which is fatal to the optimization algorithm. This will make the algorithm fall into local optima in advance, as shown in Figure 2(a). To overcome this phenomenon, a dynamic boundary adjustment strategy based on the cosine factor is proposed in this paper to improve the global exploration performance of the algorithm.

4.2.1. Boundary adjustment strategy based cosine factor

To overcoming the weak global exploration capability of BWO algorithm, we propose to improve the algorithm by using the cosine factor boundary strategy. The optimized CB_f is calculated as

$$CB_f = \max -(\max - \min) \cdot \cos(\frac{\pi T}{2T_{max}}), \quad (4.8)$$

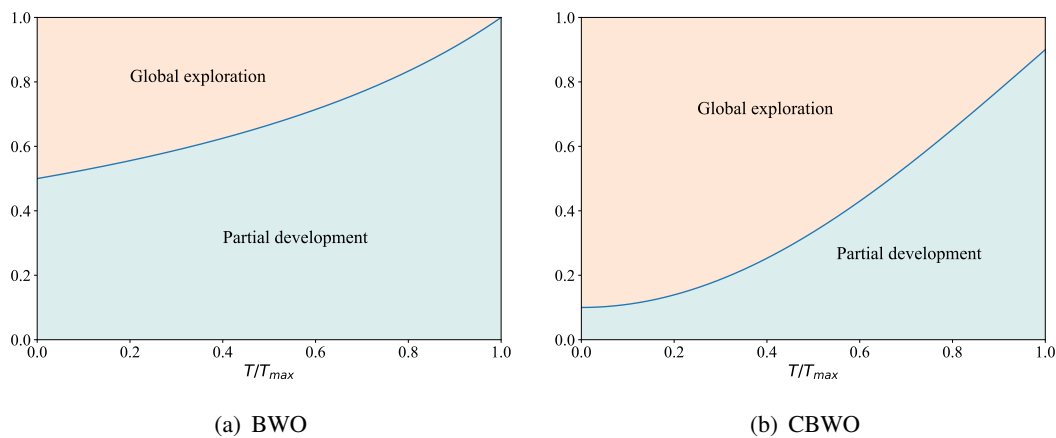


Figure 2. Phase distribution of the algorithm.

where CB_f is the balance factor based on the cosine factor. max and min are the maximum and minimum values of the balancing factor. The text sets $max = 0.9$, $min = 0.1$.

With the maximum and minimum balance factors, the algorithm can ensure that a fraction of the elements are being developed or explored, either at the beginning or at the end of the algorithm.

When the random number $B_0 > CB_f$, the algorithm enters the global exploration phase; when the random number $B_0 \leq CB_f$, the algorithm enters the local exploitation phase. The global exploration phase and its improved local exploitation state distribution are shown in Figure 2(b). By increasing the proportion of global exploration phases, we are able to find the global optimal solution faster.

4.2.2. Random perturbation in Student's t-distribution

In the local development process of the BWO algorithm, the Lévy flight strategy is utilized to avoid being trapped in local optima. However, the Lévy flight has equal jump probabilities at the beginning and end of the algorithm, and does not fully consider the impact of iteration updates on the probability. Therefore, to enhance algorithm's ability to avoid local optima, we introduce the t-distribution optimization approach.

The t-distribution is also known as the Student's t-distribution, and its distribution curve is closely related to the degree of freedom [29]. As shown in the figure, when the degree of freedom n is small, the distribution curve is flatter, and when the degree of freedom n is equal to 1, the t-distribution curve can be approximated by the Cauchy distribution curve; the greater the degree of freedom of the t-distribution, the closer its distribution curve is to the standard normal distribution curve [30]. The probability density of the t-distribution $Z \sim t(n)$ with degrees of freedom n is shown in Eq (4.9).

$$f_Z(x) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}, \quad (4.9)$$

where n is the degrees of freedom. $\Gamma(x)$ is the gamma function, where $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$, ($x > 0$).

Based on this algorithm principle, we use the influence of degrees of freedom on the production of the t-distribution to establish a new local development strategy

$$x_i^{T+1} = r_3 x_{best}^T - r_4 x_i^T + t(T)C_1(x_r^T - x_i^T), \quad (4.10)$$

where x_i^{T+1} represents the next generation individual of the i^{th} solution, x_{best}^T represents the current optimal solution, x_r is the randomly selected solution, x_i represents the i^{th} solution, r_3 and r_4 are random numbers in the interval (0, 1), and C_1 is the adaptive equilibrium factor, where $C_1 = 2r_4(1 - T/T_{max})$. $t(T)$ obeys the t-distribution $Z \sim t(T)$.

Algorithm 1 is the pseudocode of the tCBWO algorithm proposed in this section.

Algorithm 1 tCBWO algorithm

Input: Population size n , Maximum iterations T_{max} , fitness functions

- 1: Initialize the population of whales randomly within the input space
- 2: Evaluate the fitness of each whale
- 3: Record the best solution and its fitness
- 4: **for** $T = 1$ to T_{max} **do**
- 5: Calculate the equilibrium factors CB_f and W_f respectively using Eqs (4.8) and (4.6)
- 6: Generate the random number matrix B_0
- 7: **for** each whale in the population **do**
- 8: **if** $B_0[i] > CB_f$ **then** ▷ perform the exploration stage
- 9: Select a random whale r from the population
- 10: Generate two random numbers r_1 and r_2
- 11: Generate a random permutation P_j of the dimensions.
- 12: Update the position of i -th beluga whale using Eq (4.4)
- 13: **else** ▷ perform the exploitation stage
- 14: Generate two random numbers r_3 and r_4
- 15: Generate a t-distribution value $t(T)$
- 16: Update the position of i -th beluga whale using Eq (4.10)
- 17: **end if**
- 18: Handle any out-of-bounds values by resetting them to the closest bound.
- 19: **end for**
- 20: **for** each whale in the population **do** ▷ Enter the beluga whale fall phase
- 21: **if** $B_0[i] > W_f$ **then**
- 22: Update the step factor C_2 and the step size X_{step}
- 23: Update the position of i -th beluga whale using Eq (4.6)
- 24: Handle any out-of-bounds values by resetting them to the closest bound
- 25: **end if**
- 26: **end for**
- 27: Evaluate the fitness function of each beluga whale.
- 28: If a new best solution is found, record it and its fitness value
- 29: **end for**

Output: best solution, best fitness functions value

5. tCBWO-DPR for WSNs

In this section, combined with the network characteristics of WSNs, we plan the network with the advantages of metaheuristic tCBWO and classical heuristic Prim to improve network performance,

which we call tCBWO-DPR. Its approximate process is shown in Figure 3.

In Section 5.1, cluster heads are selected using improved tCBWO. In the cluster head selection process, we establish a cluster head evaluation function through the relationship between node energy and node position. We select the nodes that are more suitable for selecting cluster heads by the tCBWO algorithm proposed in Section 3. The set of selected cluster heads is shown in Figure 3.

In Section 5.2, we improve the Prim algorithm to establish inter-cluster routing. Combined with the wireless sensor network energy consumption model, we improve the Prim algorithm based on the nodes' suitability for data transmission characteristics in the form of a single hop within a range of 104.4 meters. The main method is that the nodes within 104.4 meters from the base station are directly linked to the base station. The remaining cluster head nodes are directly connected using the Prim algorithm. This section is depicted in Figure 3 under the "Constructing inter-cluster routing" section.

In Section 5.3, the inter-cluster routing setup process is described. Ordinary nodes are directly connected by picking the nearest node to them to minimize the effect of distance on energy consumption. This section is depicted in Figure 3 under the "Constructing intra-cluster routing" section.

In Section 5.4, the complexity of the algorithm is analyzed in detail.

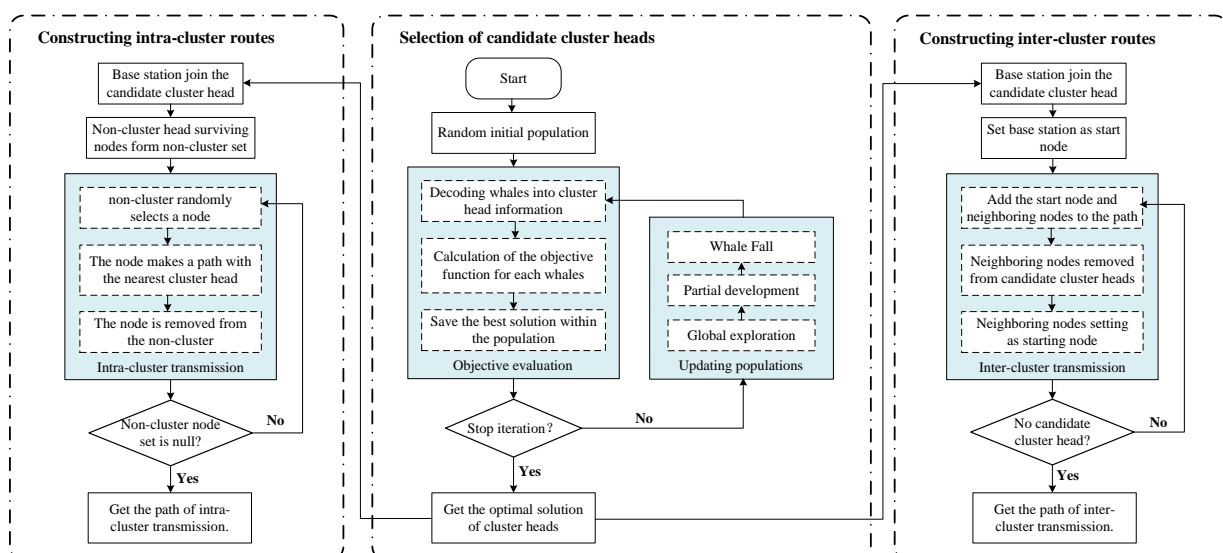


Figure 3. Flowchart of tCBWO-DPR clustering.

5.1. Election of cluster heads

This is the key step in the cluster head election process. Network nodes are managed and controlled uniformly by the base station. The base station calculates the candidate cluster heads according to the node location and energy information and calculates the number of subclusters. The threshold for the number of cluster head nodes set in this paper is 15% of the surviving nodes. Then, we combine this information with the selected cluster heads of the nodes, using the adaptive function Eq (5.1) defined in this paper to calculate the evaluation index of the candidate cluster headset, and continuously update the optimal value through tCBWO until the optimal cluster headset is selected. This process is a key part of the selection of cluster heads. At this stage, the node position and energy information are fully

considered to ensure the quality of the candidate cluster headset.

Based on the residual energy information of nodes and the positional relationship between nodes, combined with the optimization algorithm, the cluster head is selected in the next stage. The optimization function established by the cluster head election in this paper is

$$Fitness = \alpha g_1 + (1 - \alpha)g_2, \quad (5.1)$$

where α and $(1 - \alpha)$ are the weight parameters of the variables. g_1 and g_2 represent the node energy factor and the node position balance factors, respectively.

1) Energy factor (g_1)

As the energy is directly related to the lifetime of the sensor network, nodes with more remaining energy are used as cluster heads to better balance the energy consumption of the network. The expression is as follows [19]:

$$g_1 = \frac{\sum_i^M E(n_i)}{\sum_j^N E(CH_j)}, \quad (5.2)$$

where $\sum_i^M E(n_i)$ is the sum of the residual energy of the network, and $\sum_j^N E(CH_j)$ is the sum of the residual energy of the cluster headset.

2) Node position balance factor (g_1)

The positional relationship of nodes will affect the data transmission problem. When the distance between nodes exceeds a certain threshold, the energy of the transmitted data packet and the distance between nodes show a power-of-four relationship, that is, it will rise sharply as the distance increases. Therefore, the network is more balanced by the coverage of network nodes by the cluster head. Its expression is shown in Eq (5.3).

$$g_2 = \frac{M}{\sum_{j=1}^N \sum_{i=1}^M Count(dist(CH_j, n_i) < d_0)}, \quad (5.3)$$

where $dist(CH_j, n)$ is the distance from the cluster head j to the neighbor node i , and d_0 is the effective coverage of the cluster head node signal, which is set to $d_0 = 40$ in this paper. N is the number of cluster heads, and M is the number of network nodes. $\sum_{i=1}^M Count(dist(CH_j, n_i) < d_0)$ is the number of nodes that the cluster head can effectively cover.

5.2. Inter-cluster transmission

To promote energy consumption equality across the network, this section presents an elaboration on the inter-cluster transmission approach. In this paper, we have enhanced the Prim algorithm to cater to the specific characteristics of WSNs. Although the Prim algorithm guarantees a minimum transmission path at each step, it causes all the network nodes' data to converge to the node closest to the base station, neglecting the fact that multi-hop transmissions waste network energy [31]. Thus, we have proposed the

DPR algorithm, which combines the Prim algorithm and WSN transmission features, to establish an inter-cluster path planning algorithm. This algorithm addresses the challenges mentioned above, leading to optimal energy consumption across the network.

5.2.1. Classical Prim algorithm

The Prim algorithm is a classical algorithm of graph theory, which can be used to find the minimum path of the network population in weighted graphs [32]. The algorithm entails finding the optimal path through greedy strategies. Through this algorithm, the overall shortest path in dense networks can be obtained simply and quickly.

However, since the data of the end node of the WSNs will be forwarded by the previous node, the algorithm simply uses the node nearest the source node as the aggregation node, resulting in the node closest to the source node bearing too high of an energy overhead, which affects the stability of the network. This is fatal and will seriously affect the stability of the network.

5.2.2. DPR: Improved Prim algorithm

In order to overcome the limitation that the classical Prim algorithm treats the distance starting node as a new convergence node, we propose an improved Prim algorithm based on a distance factor. According to the network energy consumption model, when the distance between nodes and the base station is less than 104.4 m, nodes can effectively reduce the overall network overhead by using the method of communicating directly with the base station [33]. To this end, in the inter-cluster communication stage, we communicate directly with the base station with the cluster head node that is less than 104.4 from the base station location. At the subsequent stage, we continue to use the Prim algorithm to build inter-cluster communication.

The specific implementation process is shown in Figure 4. The distribution and positional relationship of cluster heads are shown in Figure 4(a), which is the base station, and A, B, C, E, and D are the candidate cluster heads, respectively. The DPR algorithm execution process is as follows:

- 1) Construct candidate cluster head $C = \{A, B, C, E, D, S\}$, visited set $V = \{\}$, and routes set $R = \{\}$.
- 2) Starting from base station S, remove base station S from candidate set C and add it to V. $C = \{A, B, C, E, D\}$, $V = \{S\}$, $R = \{\}$.
- 3) Find nodes A and C whose distance S is less than $d = 104.4$ from the candidate set, and construct paths (A, S) and (C, S) with S respectively, add them to R, and remove A and C from the candidate set C. Remove and add to the visited set V, as shown in Figure 4(b). $C = \{B, E, D\}$, $V = \{S, A, C\}$, $R = \{(A, S), (C, S)\}$.
- 4) Starting from the visited set V, the paths to the candidate set C are (A, B), (C, D) and (A, D), where (C, D) has the smallest path and can form a minimum spanning tree, so add (D, C) to path R, remove D from the candidate set C, and add D to V, as shown in Figure 4(c). $C = \{B, E\}$, $V = \{S, A, C, D\}$, $R = \{(A, S), (C, S), (D, C)\}$.
- 5) Starting from the visited set V, the paths to the candidate set C are (A, B), (D, B) and (E, D), where the paths of (A, B) are the smallest and can form a minimum spanning tree, so add (B, A) to path

R, remove B from the candidate set C, and add B to V, as shown in Figure 4(d). $C = \{E\}$, $V = \{S, A, B, C, D\}$, $R = \{(A, S), (C, S), (D, C), (B, A)\}$

6) Starting from the visited set V, the paths to the candidate set C are (B, E) and (D, E), where (B, E) has the smallest path and can form a minimum spanning tree, so (E, B) is added to path R, E is removed from the candidate set C, and E is added to V, as shown in Figure 4(e). $C = \{\}$, $V = \{S, A, B, C, D, E\}$, $R = \{(A, S), (C, S), (D, C), (B, A), (E, B)\}$

7) After the above operations, all elements in C are moved into V, and (A, S), (C, S), (D, C), (B, A), and (E, B) form a spanning tree, as shown in Figure 4(f).

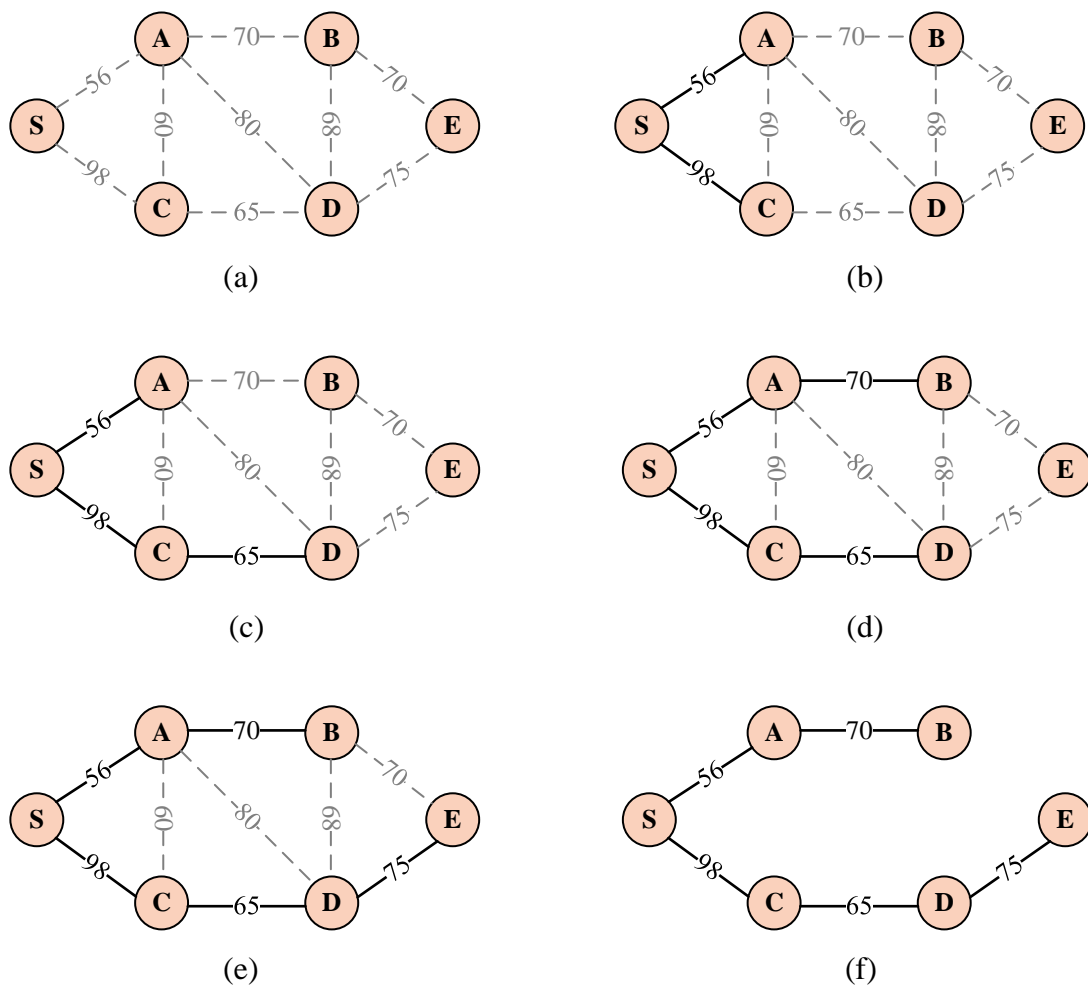


Figure 4. DPR execution process.

Algorithm 2 is a pseudocode for establishing inter-cluster routing.

Algorithm 2 DPR algorithm for routing planning

Input: cluster head candidates, cluster head positional, sink

```

1: visited=set ()           ▶ Initialize an empty set visited containing the sink node
2: vistied.add(sink)       ▶ Initialize a set candidates containing all nodes except the sink
3: routes= []             ▶ Sets an empty list routes
4: while candidates do
5:   neighbors ← Candidate nodes within 104 meters of the base station
6:   if neighbors != None then ▶ Add the nearest node to visited, remove it from candidates, and
   add the route to routes
7:     vistied.add(neighbors)
8:     candidates.remove(neighbors)
9:     routes.append((neighbors, sink))
10:  else                 ▶ find the nearest unvisited node to the visited set
11:    while visited do
12:      min_src ← Find the nearest unreachable node from the candidate set to the visited node
13:      min_dst ← Find the nearest visited node from the visited set to the candidate node
14:      vistied.add(min_src)
15:      candidates.remove(min_src)
16:      routes.append((min_src, min_dst))
17:    end while
18:  end if
19: end while
Output: routes

```

5.3. Intra-cluster transmission

Based on the network energy consumption model, the energy consumed for transmission between nodes in a wireless sensor network decreases as nodes get closer to each other. Therefore, we can effectively reduce the node energy overhead by choosing the method of direct connection with the node closest to the cluster head node that has not yet joined the cluster headset. This method can not only maximize the physical distance relationship between nodes, but also avoid unnecessary energy waste.

5.4. Algorithm complexity

In the cluster head election process, the evaluation function of cluster head nodes is evaluated by the tCBWO algorithm. The tCBWO algorithm is an improved algorithm based on BWO. The computational complexity of the cluster head election process is moderate. In the cluster head election process, the time complexity is $O(\text{pop-size} * \text{iterations} * n)$, where pop-size is the set population size, iterations are the number of iterations of the whale population, and n is the number of cluster head candidates. The algorithm involves multiple mathematical operations, including positive selection functions, random number generation, and array operations. Multiple loop nesting is used. For medium-scale NP problems, the algorithm is effective and feasible.

In the process of inter-cluster transmission, the improved DPrim algorithm is used to establish the inter-cluster transmission path. By introducing the distance factor, the data at the end will not converge on the same node. The computational complexity of inter-cluster routing is moderate. The computational complexity of inter-cluster routing is $O(n^2)$, and n is the number of cluster head candidate sets. In WSNs, the number of cluster head nodes of hierarchical routing usually occupies between 10 and 20% of all nodes, so its complexity is actually not large. Compared with the tCBWO algorithm, the computational complexity is small, and the algorithm is feasible.

Intra-cluster transmission is mainly performed by nodes that have not joined the candidate cluster head. They compare the distance to the cluster head node and select the closest node to connect. Therefore, its time complexity is $O((N - n) * n)$, where N is the number of network nodes and n is the number of cluster head nodes. The algorithm complexity of the process is relatively large compared to inter-cluster routing, but it is better than cluster head election and inter-cluster path. The algorithm can be further optimized to improve its performance.

6. Simulation experiments

The test was run on a computer with an Intel Core i5-10500 3.1 GHz processor and 16 GB of RAM at 2400 MHz, using PyCharm 2021.3 as the test IDE and Python version 3.10.

6.1. Experimental analysis of tCBWO

The test functions chosen to evaluate the performance of the tCBWO algorithm in this paper are derived from CEC2017 [34]. These benchmark functions are often used to evaluate the convergence performance of optimization algorithms. In CEC2017, the test functions are classified into single-peaked functions ($F1-F3$), simple multi-peaked functions ($F4-F10$), hybrid functions ($F11-F20$), combined functions ($F21-F30$) with test intervals of $[-100, 100]$, and test dimensions of 10, 30 and 50, respectively. The test difficulty increases sharply as the dimension increases, and the optimal values of all these functions are $\min(Fi(x) = 100 * i)$, where i is the function ID.

This test evaluated the tCBWO algorithm in comparison with HHO [35], SPBO [36], PSO [37], AOA [38], SSA [39] and the original BWO algorithm. For problem spaces of different dimensions (10, 30, 50), each algorithm was run 50 times repeatedly, and each run performed $50 * 500$ benchmark function evaluations. The exploration space range for all algorithms was $[-100, 100]$. The parameter settings used in the experiments are detailed in Table 1. The source code for the benchmark function is available at <https://github.com/tilleyd/cec2017-py>.

Table 1. Algorithm parameters.

| Algorithm | Parameters |
|------------|-----------------------------|
| PSO (1995) | $w = 0.3, c_1 = 2, c_2 = 2$ |
| AOA (2021) | $\alpha = 5, \mu = 0.5$ |

Tables 2–4 show the convergence energy of each optimization function under different functions and dimensions, where “mean” represents the average level in 50 tests, and “std” represents the stability

level in the test. The smaller the value, the better. Bold characters are optimal solutions for multiple algorithms with the same function and the same dimension.

Table 2 presents the results of CEC2017 test functions performed in 10-dimensions. This result shows that the tCBWO algorithm performs well in terms of stability and convergence ability, significantly outperforming the other algorithms. Among the 30 test functions, the tCBWO algorithm shows optimal performance on the other test functions, except for the test functions F10, F12, F15 and F26, where the mean values are not optimal. In terms of balance, only F11, F12, F14, F15, F22, F23, and F24 did not reach the optimal level of stability. Compared to the BWO algorithm, tCBWO is better than BWO algorithm in terms of both convergence ability and convergence stability of the algorithm. Combined with the convergence curve Figure 5, it can be seen that tCBWO is ultimately superior to other algorithms in terms of its ability to converge compared to other algorithms. Compared to the original BWO algorithm, there is a significant improvement in convergence speed and convergence results.

In Table 3, we show 30-dimensional results for CEC017 test functions. We also find that the tCBWO algorithm performs well in terms of stability and convergence. In terms of convergence of the algorithm, tCBWO is not an optimal solution for functions F1, F5, F7, F9, F10, F11, F16, F19, F25, F26 and F29. But, compared to the BWO algorithm, the tCBWO algorithm still shows some improvement for F1, F5, F11, F19, F25 and F26. Combined with the convergence curve in Figure 6, it can be seen that the final convergence results of tCBWO are mostly superior compared to other algorithms. And, compared with the original BWO algorithm, the convergence speed and convergence results have some degree of improvement.

Table 4 shows the results of CEC2017 test functions in 50-dimensions. In the high-dimensional complex functions, we have observed that the tCBWO algorithm shows an advantage among the six algorithms only in F1, F2, F5, F7, F17, F18, F20 and F26. However, compared to the BWO algorithm, the convergence of the tCBWO is relatively weak in the F3, F6, F8, F9, F10 and F11 functions. Therefore, the performance of the tCBWO algorithm is improved in higher dimensional functions compared to the BWO algorithm. Combined with the convergence curve in Figure 7, it can be seen that tCBWO has a certain degree of improvement in convergence speed and convergence results compared to the original BWO algorithm in 50-dimensional space.

Combining all these experimental results, we comprehensively analyze the performance of the tCBWO algorithm under different dimensions. Under 10-dimensions, the tCBWO algorithm significantly outperforms other algorithms in terms of stability and convergence ability, and performs well on most of the test functions, except for a few test functions that do not reach the optimal mean. In the 30th dimension, the tCBWO algorithm still outperforms in terms of stability and convergence ability, and achieves significant improvements on some functions relative to the BWO algorithm. However, in the 50th dimension, although the tCBWO algorithm still performs superiorly on some of the tested functions, it is slightly less convergent on some of the higher dimensional functions compared to the BWO algorithm.

In order to analyze the differences with other algorithms, we also performed the Wilcoxon rank sum test. The Wilcoxon results can reflect the variability between the results, and, when the p-value is less than 0.05, it means that the correlation between the algorithms is small. Table 5 shows the experimental results of tCBWO and other different algorithms run 50 times under the CEC2017 benchmark function. It can be found that tCBWO has good results in the Wilcoxon rank sum test compared to other algorithms. tCBWO has run data that is different from the results of other algorithms.

Table 2. Results for the 10-dimensional test function in CEC2017.

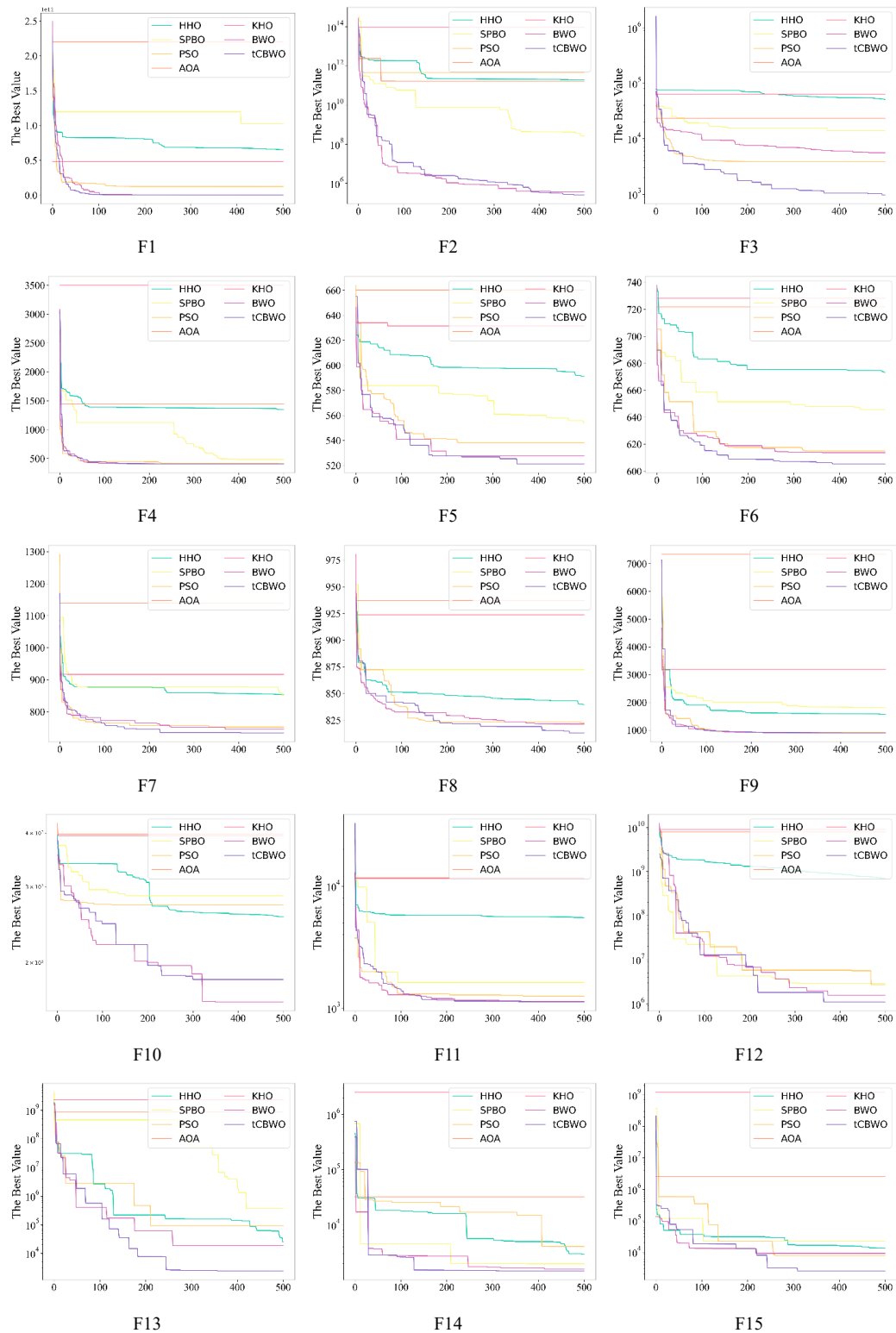
| | | HHO (2019) | SPBO (2020) | PSO (1995) | AOA (2021) | KHA (2016) | BWO (2022) | tCBWO |
|-----|------|------------|-------------|------------|------------------|------------|------------------|------------------|
| F1 | mean | 1.073E+11 | 3.924E+10 | 1.860E+10 | 1.074E+11 | 2.102E+11 | 4.014E+07 | 2.541E+06 |
| | std | 4.199E+10 | 4.048E+10 | 2.200E+10 | 3.494E+10 | 6.296E+10 | 2.467E+07 | 1.531E+06 |
| F2 | mean | 1.173E+12 | 2.368E+12 | 1.011E+11 | 5.746E+12 | 1.337E+15 | 5.925E+05 | 1.516E+05 |
| | std | 4.362E+12 | 1.527E+13 | 5.292E+11 | 1.482E+13 | 3.949E+15 | 1.314E+06 | 2.061E+05 |
| F3 | mean | 5.044E+04 | 2.769E+04 | 1.103E+04 | 4.066E+04 | 1.654E+05 | 3.448E+03 | 1.982E+03 |
| | std | 3.868E+04 | 1.755E+04 | 1.046E+04 | 2.252E+04 | 1.911E+05 | 1.132E+03 | 9.285E+02 |
| F4 | mean | 1.106E+03 | 6.429E+02 | 5.142E+02 | 1.195E+03 | 2.647E+03 | 4.095E+02 | 4.071E+02 |
| | std | 4.451E+02 | 2.465E+02 | 1.423E+02 | 1.409E+02 | 1.186E+03 | 1.009E+01 | 6.547E-01 |
| F5 | mean | 5.839E+02 | 5.768E+02 | 5.423E+02 | 6.046E+02 | 6.346E+02 | 5.228E+02 | 5.217E+02 |
| | std | 2.146E+01 | 2.581E+01 | 1.181E+01 | 2.016E+01 | 2.227E+01 | 5.145E+00 | 3.776E+00 |
| F6 | mean | 6.604E+02 | 6.579E+02 | 6.215E+02 | 6.707E+02 | 6.997E+02 | 6.063E+02 | 6.045E+02 |
| | std | 1.747E+01 | 1.878E+01 | 9.416E+00 | 1.409E+01 | 1.619E+01 | 2.432E+00 | 1.833E+00 |
| F7 | mean | 8.225E+02 | 8.081E+02 | 7.730E+02 | 9.206E+02 | 8.990E+02 | 7.420E+02 | 7.363E+02 |
| | std | 3.764E+01 | 3.439E+01 | 3.498E+01 | 3.148E+01 | 2.358E+01 | 6.591E+00 | 4.146E+00 |
| F8 | mean | 8.524E+02 | 8.511E+02 | 8.419E+02 | 9.141E+02 | 9.146E+02 | 8.201E+02 | 8.190E+02 |
| | std | 1.644E+01 | 1.932E+01 | 1.225E+01 | 1.914E+01 | 1.782E+01 | 5.973E+00 | 4.182E+00 |
| F9 | mean | 1.778E+03 | 1.795E+03 | 1.100E+03 | 3.170E+03 | 3.249E+03 | 9.039E+02 | 9.027E+02 |
| | std | 3.992E+02 | 5.233E+02 | 2.464E+02 | 7.020E+02 | 7.125E+02 | 2.640E+00 | 1.427E+00 |
| F10 | mean | 2.696E+03 | 2.714E+03 | 2.194E+03 | 3.228E+03 | 3.345E+03 | 1.630E+03 | 1.690E+03 |
| | std | 2.756E+02 | 4.584E+02 | 3.482E+02 | 2.438E+02 | 3.408E+02 | 2.137E+02 | 1.642E+02 |
| F11 | mean | 5.096E+03 | 2.459E+03 | 1.518E+03 | 8.716E+03 | 2.163E+04 | 1.132E+03 | 1.130E+03 |
| | std | 4.265E+03 | 3.552E+03 | 5.707E+02 | 8.696E+03 | 1.660E+04 | 1.038E+01 | 1.149E+01 |
| F12 | mean | 1.952E+09 | 4.237E+08 | 3.019E+08 | 6.247E+09 | 1.139E+10 | 1.383E+06 | 2.872E+06 |
| | std | 3.113E+09 | 8.262E+08 | 1.288E+09 | 2.653E+09 | 6.683E+09 | 1.245E+06 | 1.767E+06 |
| F13 | mean | 1.780E+04 | 1.950E+07 | 8.830E+04 | 6.574E+08 | 2.252E+09 | 1.498E+04 | 9.440E+03 |
| | std | 1.553E+04 | 6.184E+07 | 1.183E+05 | 6.246E+08 | 2.320E+09 | 9.839E+03 | 8.602E+03 |
| F14 | mean | 4.678E+03 | 1.247E+04 | 4.090E+03 | 6.981E+06 | 5.215E+06 | 2.104E+03 | 2.086E+03 |
| | std | 4.501E+03 | 2.110E+04 | 6.854E+03 | 7.592E+06 | 1.207E+07 | 6.001E+02 | 6.186E+02 |
| F15 | mean | 6.302E+04 | 3.818E+05 | 2.391E+04 | 1.296E+08 | 1.575E+08 | 4.138E+03 | 5.076E+03 |
| | std | 6.968E+04 | 1.649E+06 | 3.696E+04 | 1.436E+08 | 2.707E+08 | 2.087E+03 | 3.108E+03 |
| F16 | mean | 2.194E+03 | 2.086E+03 | 1.800E+03 | 2.354E+03 | 2.609E+03 | 1.696E+03 | 1.632E+03 |
| | std | 1.666E+02 | 1.876E+02 | 1.545E+02 | 1.776E+02 | 2.392E+02 | 8.903E+01 | 1.839E+01 |
| F17 | mean | 1.913E+03 | 1.871E+03 | 1.822E+03 | 2.284E+03 | 2.345E+03 | 1.737E+03 | 1.734E+03 |
| | std | 1.112E+02 | 9.366E+01 | 5.279E+01 | 1.777E+02 | 1.663E+02 | 7.746E+00 | 5.750E+00 |
| F18 | mean | 3.030E+04 | 1.778E+07 | 4.419E+04 | 1.974E+08 | 9.571E+08 | 1.675E+04 | 1.053E+04 |
| | std | 9.898E+04 | 4.956E+07 | 2.717E+04 | 1.339E+08 | 9.103E+08 | 1.286E+04 | 4.542E+03 |
| F19 | mean | 6.589E+07 | 1.520E+06 | 5.084E+04 | 1.409E+08 | 4.206E+08 | 5.908E+03 | 3.513E+03 |
| | std | 2.395E+08 | 3.314E+06 | 6.261E+04 | 1.687E+08 | 5.367E+08 | 3.820E+03 | 2.086E+03 |
| F20 | mean | 2.295E+03 | 2.247E+03 | 2.127E+03 | 2.408E+03 | 2.530E+03 | 2.032E+03 | 2.026E+03 |
| | std | 9.932E+01 | 9.680E+01 | 6.860E+01 | 9.718E+01 | 1.213E+02 | 9.979E+00 | 2.677E+00 |
| F21 | mean | 2.374E+03 | 2.348E+03 | 2.318E+03 | 2.385E+03 | 2.429E+03 | 2.236E+03 | 2.209E+03 |
| | std | 3.379E+01 | 5.094E+01 | 5.587E+01 | 5.562E+01 | 3.798E+01 | 4.938E+01 | 2.601E+01 |
| F22 | mean | 3.122E+03 | 2.564E+03 | 2.409E+03 | 3.437E+03 | 4.123E+03 | 2.308E+03 | 2.276E+03 |
| | std | 3.982E+02 | 3.701E+02 | 1.756E+02 | 3.890E+02 | 5.174E+02 | 1.537E+00 | 3.865E+01 |
| F23 | mean | 2.708E+03 | 2.703E+03 | 2.660E+03 | 2.727E+03 | 2.765E+03 | 2.620E+03 | 2.615E+03 |
| | std | 3.729E+01 | 4.550E+01 | 2.495E+01 | 2.167E+01 | 5.206E+01 | 4.541E+01 | 4.319E+01 |
| F24 | mean | 2.854E+03 | 2.833E+03 | 2.784E+03 | 2.899E+03 | 2.907E+03 | 2.692E+03 | 2.584E+03 |
| | std | 5.954E+01 | 5.427E+01 | 3.874E+01 | 3.313E+01 | 5.688E+01 | 1.092E+02 | 1.110E+02 |
| F25 | mean | 3.536E+03 | 3.118E+03 | 3.026E+03 | 3.566E+03 | 4.125E+03 | 2.919E+03 | 2.906E+03 |
| | std | 2.833E+02 | 2.311E+02 | 1.092E+02 | 1.494E+02 | 4.184E+02 | 2.086E+01 | 8.027E+00 |
| F26 | mean | 4.158E+03 | 3.801E+03 | 3.318E+03 | 4.091E+03 | 4.853E+03 | 2.896E+03 | 2.924E+03 |
| | std | 5.319E+02 | 5.075E+02 | 3.880E+02 | 6.331E+02 | 4.748E+02 | 7.391E+01 | 6.491E+01 |
| F27 | mean | 3.275E+03 | 3.192E+03 | 3.023E+03 | 3.208E+03 | 3.216E+03 | 3.098E+03 | 3.093E+03 |
| | std | 9.848E+01 | 8.031E+01 | 2.536E+01 | 6.154E+01 | 7.903E+01 | 5.378E+00 | 1.582E+00 |
| F28 | mean | 3.666E+03 | 3.607E+03 | 3.435E+03 | 3.698E+03 | 3.924E+03 | 3.263E+03 | 3.166E+03 |
| | std | 1.847E+02 | 1.931E+02 | 1.417E+02 | 2.177E+02 | 1.937E+02 | 1.332E+02 | 1.872E+01 |
| F29 | mean | 3.560E+03 | 3.446E+03 | 3.259E+03 | 3.597E+03 | 3.819E+03 | 3.204E+03 | 3.182E+03 |
| | std | 1.736E+02 | 1.411E+02 | 8.384E+01 | 1.662E+02 | 2.418E+02 | 2.906E+01 | 1.596E+01 |
| F30 | mean | 4.677E+07 | 3.592E+07 | 3.697E+06 | 7.223E+07 | 3.863E+08 | 3.379E+05 | 1.033E+05 |
| | std | 4.332E+07 | 4.451E+07 | 3.594E+06 | 9.774E+07 | 3.713E+08 | 3.724E+05 | 1.497E+05 |

Table 3. Results for the 30-dimensional test function in CEC2017.

| | | HHO (2019) | SPBO (2020) | PSO (1995) | AOA (2021) | KHA (2016) | BWO (2022) | tCBWO |
|-----|------|------------|------------------|------------------|------------------|------------------|------------------|------------------|
| F1 | mean | 5.746E+11 | 3.747E+11 | 1.825E+11 | 6.280E+11 | 8.463E+11 | 4.337E+11 | 3.902E+11 |
| | std | 8.870E+10 | 1.635E+11 | 1.226E+11 | 7.121E+10 | 1.036E+10 | 6.590E+10 | 7.250E+10 |
| F2 | mean | 1.160E+51 | 3.490E+47 | 9.520E+48 | 4.970E+47 | 3.680E+54 | 1.760E+44 | 1.740E+43 |
| | std | 5.750E+51 | 1.330E+48 | 6.660E+49 | 7.550E+47 | 2.510E+55 | 7.510E+44 | 9.350E+43 |
| F3 | mean | 2.752E+05 | 2.083E+05 | 1.529E+05 | 5.888E+05 | 2.732E+08 | 7.602E+04 | 9.440E+04 |
| | std | 5.815E+04 | 7.183E+04 | 4.775E+04 | 6.602E+05 | 4.144E+08 | 6.912E+03 | 1.187E+04 |
| F4 | mean | 1.498E+04 | 1.015E+04 | 5.131E+03 | 1.691E+04 | 2.593E+04 | 1.102E+04 | 9.788E+03 |
| | std | 3.128E+03 | 6.388E+03 | 5.559E+03 | 4.875E+03 | 6.081E+03 | 2.347E+03 | 2.354E+03 |
| F5 | mean | 9.365E+02 | 8.844E+02 | 8.029E+02 | 1.013E+03 | 1.040E+03 | 8.983E+02 | 8.981E+02 |
| | std | 7.224E+01 | 5.033E+01 | 4.897E+01 | 4.894E+01 | 4.955E+01 | 3.166E+01 | 3.595E+01 |
| F6 | mean | 7.117E+02 | 7.081E+02 | 6.753E+02 | 7.255E+02 | 7.597E+02 | 7.019E+02 | 7.016E+02 |
| | std | 1.531E+01 | 2.005E+01 | 1.693E+01 | 1.541E+01 | 1.802E+01 | 9.683E+00 | 1.205E+01 |
| F7 | mean | 1.411E+03 | 1.487E+03 | 1.266E+03 | 1.646E+03 | 1.558E+03 | 1.326E+03 | 1.336E+03 |
| | std | 6.706E+01 | 1.400E+02 | 1.540E+02 | 3.186E+01 | 6.264E+01 | 6.183E+01 | 5.993E+01 |
| F8 | mean | 1.143E+03 | 1.122E+03 | 1.084E+03 | 1.241E+03 | 1.268E+03 | 1.123E+03 | 1.113E+03 |
| | std | 3.733E+01 | 6.245E+01 | 4.446E+01 | 2.585E+01 | 3.716E+01 | 2.209E+01 | 2.394E+01 |
| F9 | mean | 1.029E+04 | 1.256E+04 | 8.648E+03 | 1.895E+04 | 2.872E+04 | 1.021E+04 | 1.079E+04 |
| | std | 3.664E+03 | 4.354E+03 | 3.742E+03 | 3.305E+03 | 4.590E+03 | 1.259E+03 | 1.318E+03 |
| F10 | mean | 9.071E+03 | 8.658E+03 | 7.534E+03 | 9.257E+03 | 1.036E+04 | 8.465E+03 | 8.674E+03 |
| | std | 6.059E+02 | 9.422E+02 | 7.235E+02 | 5.298E+02 | 5.252E+02 | 4.457E+02 | 3.849E+02 |
| F11 | mean | 3.364E+04 | 1.215E+04 | 4.303E+03 | 2.379E+04 | 1.357E+05 | 7.326E+03 | 7.276E+03 |
| | std | 1.641E+04 | 6.481E+03 | 2.984E+03 | 1.113E+04 | 1.949E+05 | 1.148E+03 | 8.391E+02 |
| F12 | mean | 1.044E+11 | 3.934E+10 | 2.008E+10 | 1.348E+11 | 2.037E+11 | 5.043E+10 | 4.123E+10 |
| | std | 4.023E+10 | 3.696E+10 | 2.508E+10 | 2.384E+10 | 4.085E+10 | 1.805E+10 | 1.732E+10 |
| F13 | mean | 6.083E+10 | 4.114E+10 | 2.632E+10 | 1.598E+11 | 2.651E+11 | 3.136E+10 | 2.559E+10 |
| | std | 4.795E+10 | 5.091E+10 | 3.634E+10 | 5.398E+10 | 9.714E+10 | 2.164E+10 | 1.082E+10 |
| F14 | mean | 1.613E+07 | 4.453E+06 | 8.819E+05 | 1.496E+07 | 6.456E+07 | 2.450E+06 | 2.120E+06 |
| | std | 2.159E+07 | 7.711E+06 | 1.655E+06 | 6.813E+06 | 7.644E+07 | 1.289E+06 | 1.017E+06 |
| F15 | mean | 1.270E+10 | 2.704E+09 | 1.198E+09 | 2.631E+10 | 5.655E+10 | 1.329E+09 | 8.831E+08 |
| | std | 1.068E+10 | 4.268E+09 | 3.467E+09 | 2.216E+10 | 1.295E+10 | 6.873E+08 | 5.139E+08 |
| F16 | mean | 6.211E+03 | 4.724E+03 | 3.881E+03 | 5.767E+03 | 9.199E+03 | 4.838E+03 | 4.652E+03 |
| | std | 1.320E+03 | 9.566E+02 | 5.588E+02 | 1.296E+03 | 2.891E+03 | 4.765E+02 | 4.183E+02 |
| F17 | mean | 5.574E+03 | 3.376E+03 | 2.782E+03 | 8.033E+03 | 3.586E+04 | 3.318E+03 | 3.261E+03 |
| | std | 4.431E+03 | 1.701E+03 | 4.213E+02 | 9.972E+03 | 4.580E+04 | 3.943E+02 | 3.158E+02 |
| F18 | mean | 1.431E+08 | 5.218E+07 | 1.038E+07 | 1.929E+08 | 8.952E+08 | 2.622E+07 | 2.384E+07 |
| | std | 1.518E+08 | 7.617E+07 | 2.231E+07 | 1.262E+08 | 5.068E+08 | 1.939E+07 | 1.529E+07 |
| F19 | mean | 1.342E+10 | 4.243E+09 | 7.338E+08 | 3.241E+10 | 6.153E+10 | 1.758E+09 | 1.623E+09 |
| | std | 1.016E+10 | 7.039E+09 | 8.011E+08 | 1.616E+10 | 1.123E+10 | 1.469E+09 | 1.413E+09 |
| F20 | mean | 3.095E+03 | 3.093E+03 | 2.816E+03 | 3.380E+03 | 3.719E+03 | 2.829E+03 | 2.885E+03 |
| | std | 2.405E+02 | 2.838E+02 | 2.013E+02 | 1.948E+02 | 2.413E+02 | 1.422E+02 | 1.392E+02 |
| F21 | mean | 2.754E+03 | 2.683E+03 | 2.606E+03 | 2.751E+03 | 2.915E+03 | 2.694E+03 | 2.668E+03 |
| | std | 6.413E+01 | 8.245E+01 | 6.614E+01 | 4.111E+01 | 7.291E+01 | 3.479E+01 | 4.599E+01 |
| F22 | mean | 1.012E+04 | 9.590E+03 | 8.091E+03 | 1.060E+04 | 1.210E+04 | 7.315E+03 | 7.261E+03 |
| | std | 8.070E+02 | 1.257E+03 | 1.852E+03 | 6.367E+02 | 4.615E+02 | 1.141E+03 | 1.027E+03 |
| F23 | mean | 3.517E+03 | 3.415E+03 | 3.129E+03 | 3.322E+03 | 3.789E+03 | 3.266E+03 | 3.240E+03 |
| | std | 1.727E+02 | 2.424E+02 | 1.216E+02 | 1.244E+02 | 1.751E+02 | 7.141E+01 | 9.746E+01 |
| F24 | mean | 3.699E+03 | 3.567E+03 | 3.350E+03 | 3.644E+03 | 3.788E+03 | 3.393E+03 | 3.387E+03 |
| | std | 1.948E+02 | 2.354E+02 | 1.429E+02 | 7.656E+01 | 1.673E+02 | 1.078E+02 | 1.155E+02 |
| F25 | mean | 5.308E+03 | 4.842E+03 | 3.728E+03 | 7.155E+03 | 7.213E+03 | 4.324E+03 | 4.228E+03 |
| | std | 6.613E+02 | 1.374E+03 | 7.134E+02 | 1.500E+03 | 1.131E+03 | 2.782E+02 | 2.288E+02 |
| F26 | mean | 1.172E+04 | 1.010E+04 | 8.662E+03 | 1.124E+04 | 1.452E+04 | 9.800E+03 | 9.136E+03 |
| | std | 1.470E+03 | 1.189E+03 | 1.650E+03 | 1.126E+03 | 1.069E+03 | 7.493E+02 | 9.722E+02 |
| F27 | mean | 4.417E+03 | 3.913E+03 | 3.483E+03 | 4.246E+03 | 4.798E+03 | 3.708E+03 | 3.682E+03 |
| | std | 3.751E+02 | 5.082E+02 | 1.575E+02 | 1.658E+02 | 5.355E+02 | 1.589E+02 | 1.615E+02 |
| F28 | mean | 7.577E+03 | 6.122E+03 | 5.900E+03 | 7.524E+03 | 8.887E+03 | 5.915E+03 | 5.759E+03 |
| | std | 9.150E+02 | 1.331E+03 | 1.841E+03 | 9.280E+02 | 8.039E+02 | 5.067E+02 | 4.595E+02 |
| F29 | mean | 8.258E+03 | 6.478E+03 | 5.051E+03 | 8.547E+03 | 6.261E+04 | 6.099E+03 | 6.034E+03 |
| | std | 2.184E+03 | 1.691E+03 | 8.032E+02 | 2.772E+03 | 8.242E+04 | 7.246E+02 | 7.142E+02 |
| F30 | mean | 1.237E+10 | 2.510E+09 | 8.592E+08 | 2.696E+10 | 4.106E+10 | 2.764E+09 | 1.587E+09 |
| | std | 1.300E+10 | 3.718E+09 | 3.014E+09 | 1.100E+10 | 1.931E+10 | 1.633E+09 | 9.830E+08 |

Table 4. Results for the 50-dimensional test function in CEC2017.

| | | HHO(2019) | SPBO(2020) | PSO(1995) | AOA(2021) | KHA(2016) | BWO(2022) | tCBWO |
|-----|------|-----------|------------|------------------|------------------|------------------|------------------|------------------|
| F1 | mean | 1.123E+12 | 9.035E+11 | 5.182E+11 | 1.227E+12 | 1.357E+12 | 9.278E+11 | 8.941E+11 |
| | std | 8.043E+10 | 2.927E+11 | 2.397E+11 | 6.303E+10 | 1.522E+07 | 7.183E+10 | 7.425E+10 |
| F2 | mean | 6.202E+88 | 3.410E+84 | 3.099E+83 | 7.999E+84 | 1.858E+88 | 2.364E+79 | 9.905E+76 |
| | std | 2.750E+89 | 2.288E+85 | 2.164E+84 | 1.222E+85 | 1.170E+88 | 9.723E+79 | 3.839E+77 |
| F3 | mean | 4.473E+05 | 3.716E+05 | 3.733E+05 | 4.179E+05 | 3.585E+10 | 2.205E+05 | 2.484E+05 |
| | std | 1.030E+05 | 1.208E+05 | 9.209E+04 | 9.017E+04 | 1.611E+11 | 2.917E+04 | 2.519E+04 |
| F4 | mean | 4.037E+04 | 2.548E+04 | 1.135E+04 | 3.868E+04 | 4.480E+04 | 2.562E+04 | 2.558E+04 |
| | std | 7.754E+03 | 1.374E+04 | 8.443E+03 | 9.023E+03 | 8.206E+03 | 4.804E+03 | 5.019E+03 |
| F5 | mean | 1.210E+03 | 1.182E+03 | 1.177E+03 | 1.279E+03 | 1.307E+03 | 1.173E+03 | 1.168E+03 |
| | std | 5.166E+01 | 8.474E+01 | 8.984E+01 | 4.456E+01 | 4.317E+01 | 2.813E+01 | 3.221E+01 |
| F6 | mean | 7.365E+02 | 7.401E+02 | 7.066E+02 | 7.449E+02 | 7.716E+02 | 7.314E+02 | 7.331E+02 |
| | std | 1.347E+01 | 1.648E+01 | 1.879E+01 | 1.211E+01 | 1.131E+01 | 9.138E+00 | 8.709E+00 |
| F7 | mean | 1.982E+03 | 2.268E+03 | 2.079E+03 | 2.200E+03 | 2.142E+03 | 1.912E+03 | 1.889E+03 |
| | std | 8.097E+01 | 3.205E+02 | 3.566E+02 | 2.085E+01 | 5.723E+01 | 6.616E+01 | 7.841E+01 |
| F8 | mean | 1.520E+03 | 1.506E+03 | 1.463E+03 | 1.639E+03 | 1.654E+03 | 1.472E+03 | 1.486E+03 |
| | std | 6.921E+01 | 9.069E+01 | 8.560E+01 | 3.510E+01 | 3.428E+01 | 3.761E+01 | 3.205E+01 |
| F9 | mean | 3.553E+04 | 4.106E+04 | 3.287E+04 | 5.909E+04 | 7.180E+04 | 3.736E+04 | 3.795E+04 |
| | std | 9.772E+03 | 8.785E+03 | 8.971E+03 | 8.286E+03 | 6.727E+03 | 2.404E+03 | 3.732E+03 |
| F10 | mean | 1.533E+04 | 1.494E+04 | 1.397E+04 | 1.522E+04 | 1.786E+04 | 1.476E+04 | 1.479E+04 |
| | std | 8.138E+02 | 1.511E+03 | 1.083E+03 | 5.687E+02 | 6.802E+02 | 4.469E+02 | 5.192E+02 |
| F11 | mean | 3.144E+04 | 3.573E+04 | 1.482E+04 | 4.667E+04 | 2.559E+05 | 2.068E+04 | 2.225E+04 |
| | std | 1.121E+04 | 1.421E+04 | 7.602E+03 | 1.652E+04 | 3.448E+05 | 2.546E+03 | 3.345E+03 |
| F12 | mean | 7.265E+11 | 3.941E+11 | 1.913E+11 | 6.935E+11 | 1.167E+12 | 3.601E+11 | 2.652E+11 |
| | std | 1.837E+11 | 2.163E+11 | 1.310E+11 | 1.389E+11 | 1.443E+11 | 1.114E+11 | 1.163E+11 |
| F13 | mean | 4.802E+11 | 1.778E+11 | 1.090E+11 | 4.330E+11 | 8.753E+11 | 2.271E+11 | 1.731E+11 |
| | std | 1.524E+11 | 1.530E+11 | 9.643E+10 | 1.081E+11 | 2.042E+11 | 9.097E+10 | 6.586E+10 |
| F14 | mean | 1.083E+08 | 6.149E+07 | 1.095E+07 | 9.821E+07 | 3.542E+08 | 2.628E+07 | 1.814E+07 |
| | std | 8.905E+07 | 7.537E+07 | 1.295E+07 | 7.116E+07 | 1.699E+08 | 1.638E+07 | 1.069E+07 |
| F15 | mean | 9.486E+10 | 3.454E+10 | 6.574E+09 | 1.364E+11 | 2.308E+11 | 3.576E+10 | 2.553E+10 |
| | std | 4.008E+10 | 3.304E+10 | 1.133E+10 | 7.114E+10 | 1.988E+10 | 1.402E+10 | 1.127E+10 |
| F16 | mean | 1.047E+04 | 7.766E+03 | 6.143E+03 | 9.203E+03 | 1.471E+04 | 7.622E+03 | 7.086E+03 |
| | std | 2.444E+03 | 1.671E+03 | 1.044E+03 | 2.128E+03 | 2.440E+03 | 8.138E+02 | 5.073E+02 |
| F17 | mean | 4.020E+04 | 9.711E+03 | 6.540E+03 | 1.527E+05 | 8.671E+04 | 5.768E+03 | 5.001E+03 |
| | std | 6.263E+04 | 1.883E+04 | 5.423E+03 | 1.343E+05 | 6.233E+04 | 1.176E+03 | 4.341E+02 |
| F18 | mean | 3.473E+08 | 9.108E+07 | 6.389E+07 | 2.902E+08 | 1.255E+09 | 9.473E+07 | 5.800E+07 |
| | std | 3.387E+08 | 6.930E+07 | 8.311E+07 | 2.383E+08 | 4.893E+08 | 5.036E+07 | 3.082E+07 |
| F19 | mean | 4.015E+10 | 1.927E+10 | 8.521E+09 | 8.184E+10 | 1.255E+11 | 2.185E+10 | 1.312E+10 |
| | std | 1.584E+10 | 2.257E+10 | 1.145E+10 | 1.962E+10 | 2.039E+10 | 1.129E+10 | 6.943E+09 |
| F20 | mean | 4.329E+03 | 4.382E+03 | 4.059E+03 | 4.731E+03 | 5.112E+03 | 3.852E+03 | 3.837E+03 |
| | std | 2.922E+02 | 3.579E+02 | 2.808E+02 | 2.396E+02 | 2.316E+02 | 2.170E+02 | 1.450E+02 |
| F21 | mean | 3.244E+03 | 3.103E+03 | 2.978E+03 | 3.193E+03 | 3.554E+03 | 3.138E+03 | 3.124E+03 |
| | std | 1.027E+02 | 1.351E+02 | 9.843E+01 | 4.526E+01 | 1.188E+02 | 6.100E+01 | 6.807E+01 |
| F22 | mean | 1.713E+04 | 1.687E+04 | 1.518E+04 | 1.697E+04 | 1.944E+04 | 1.673E+04 | 1.669E+04 |
| | std | 8.425E+02 | 1.311E+03 | 1.034E+03 | 8.816E+02 | 6.732E+02 | 4.609E+02 | 4.413E+02 |
| F23 | mean | 4.498E+03 | 4.317E+03 | 3.855E+03 | 4.106E+03 | 4.963E+03 | 3.975E+03 | 3.970E+03 |
| | std | 2.551E+02 | 3.860E+02 | 2.002E+02 | 1.756E+02 | 2.649E+02 | 1.392E+02 | 1.420E+02 |
| F24 | mean | 4.709E+03 | 4.545E+03 | 3.974E+03 | 4.503E+03 | 4.841E+03 | 4.209E+03 | 4.162E+03 |
| | std | 2.351E+02 | 4.428E+02 | 2.139E+02 | 2.767E+02 | 2.879E+02 | 2.030E+02 | 2.162E+02 |
| F25 | mean | 1.605E+04 | 1.331E+04 | 8.859E+03 | 1.796E+04 | 1.707E+04 | 1.269E+04 | 1.252E+04 |
| | std | 1.751E+03 | 3.705E+03 | 3.456E+03 | 1.597E+03 | 1.848E+03 | 1.278E+03 | 1.363E+03 |
| F26 | mean | 1.901E+04 | 1.723E+04 | 1.611E+04 | 1.731E+04 | 1.886E+04 | 1.588E+04 | 1.577E+04 |
| | std | 2.407E+03 | 2.110E+03 | 2.704E+03 | 1.299E+03 | 9.186E+02 | 7.973E+02 | 8.490E+02 |
| F27 | mean | 7.338E+03 | 5.983E+03 | 4.590E+03 | 5.928E+03 | 8.136E+03 | 5.211E+03 | 4.771E+03 |
| | std | 1.249E+03 | 1.092E+03 | 4.555E+02 | 5.976E+02 | 1.294E+03 | 4.037E+02 | 3.120E+02 |
| F28 | mean | 1.464E+04 | 1.137E+04 | 1.114E+04 | 1.401E+04 | 1.719E+04 | 1.103E+04 | 1.139E+04 |
| | std | 1.833E+03 | 2.074E+03 | 3.330E+03 | 1.957E+03 | 2.182E+03 | 8.666E+02 | 8.297E+02 |
| F29 | mean | 2.201E+05 | 3.122E+04 | 9.464E+03 | 8.361E+04 | 2.104E+06 | 1.569E+04 | 1.382E+04 |
| | std | 3.647E+05 | 5.317E+04 | 3.856E+03 | 7.820E+04 | 1.640E+06 | 5.416E+03 | 5.001E+03 |
| F30 | mean | 7.343E+10 | 2.818E+10 | 1.097E+10 | 7.994E+10 | 1.763E+11 | 2.471E+10 | 2.172E+10 |
| | std | 3.076E+10 | 3.000E+10 | 1.364E+10 | 2.464E+10 | 5.115E+10 | 9.228E+09 | 9.856E+09 |



Continued on next page

Figure 5. Convergence curves of algorithm on CEC2017 (dim = 10).

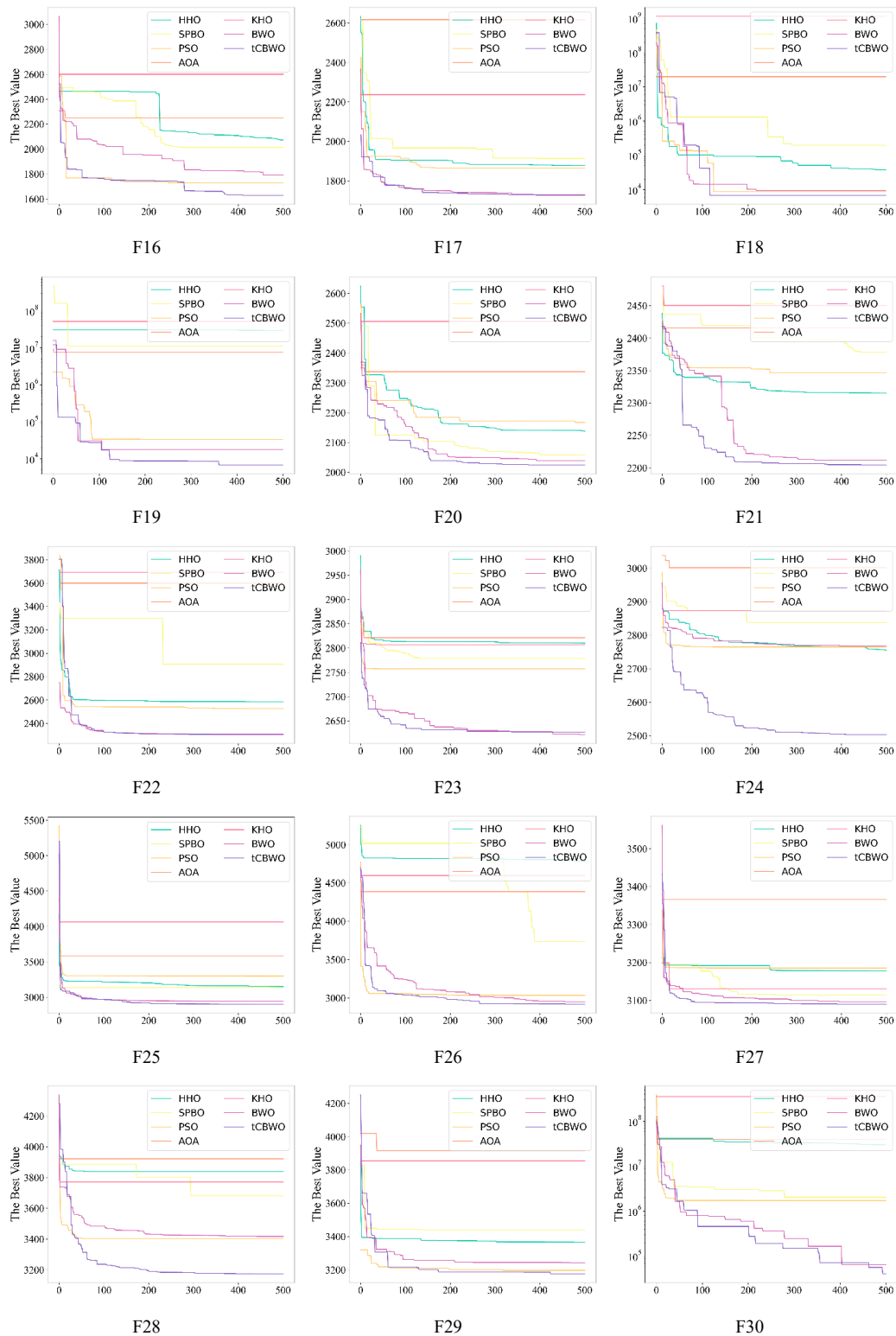
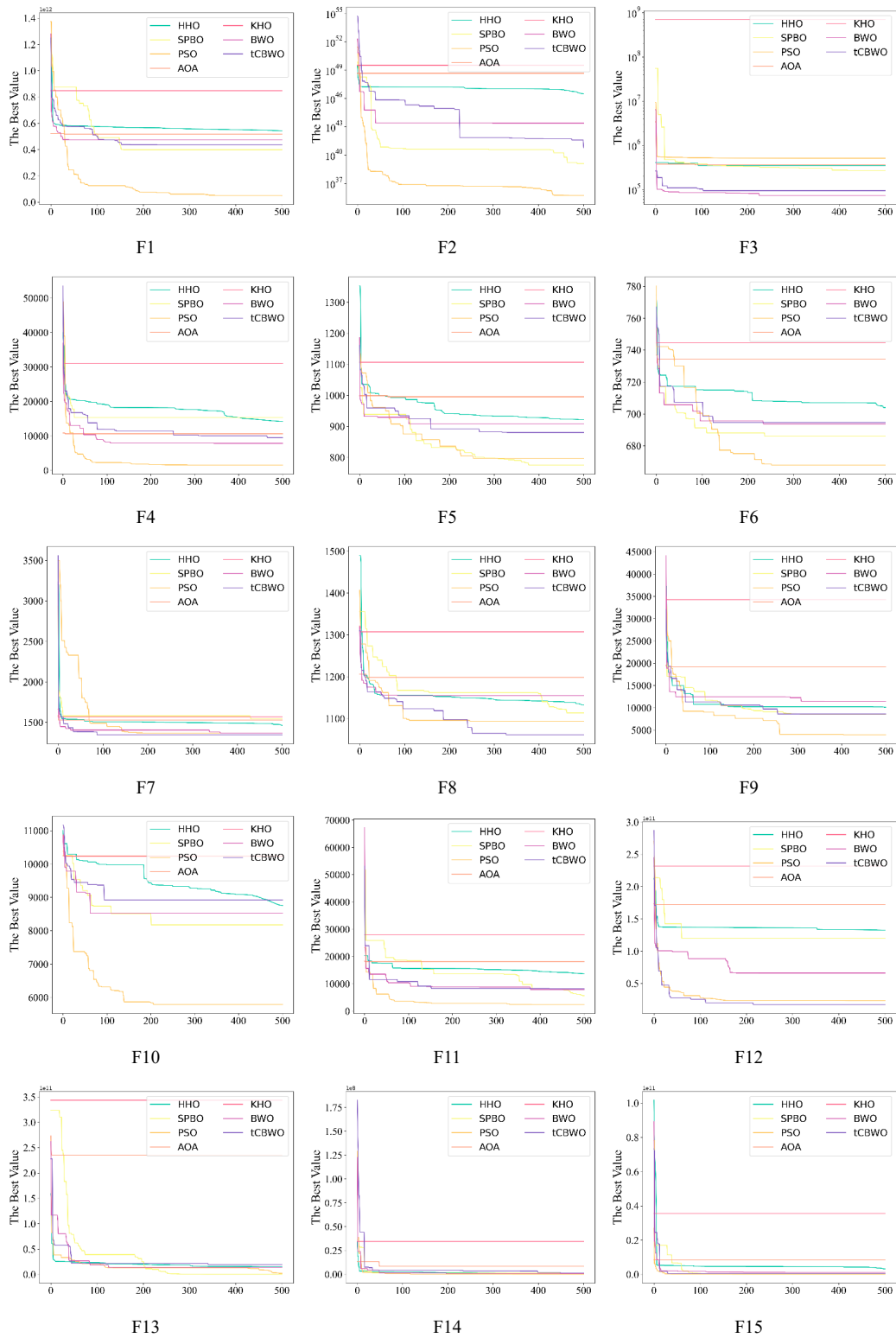


Figure 5. Convergence curves of algorithm on CEC2017 (dim = 10).



Continued on next page

Figure 6. Convergence curves of algorithm on CEC2017 (dim = 30).

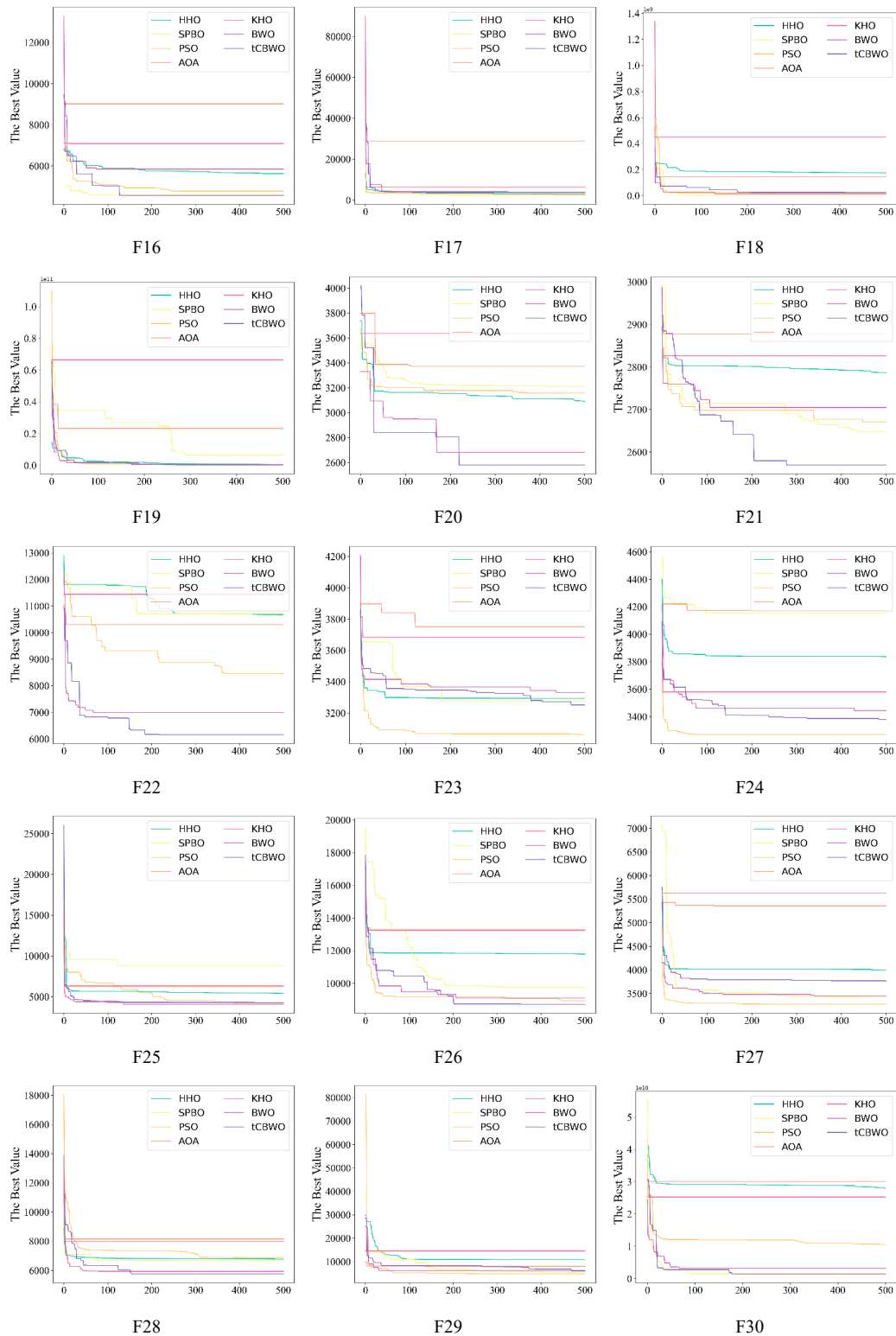
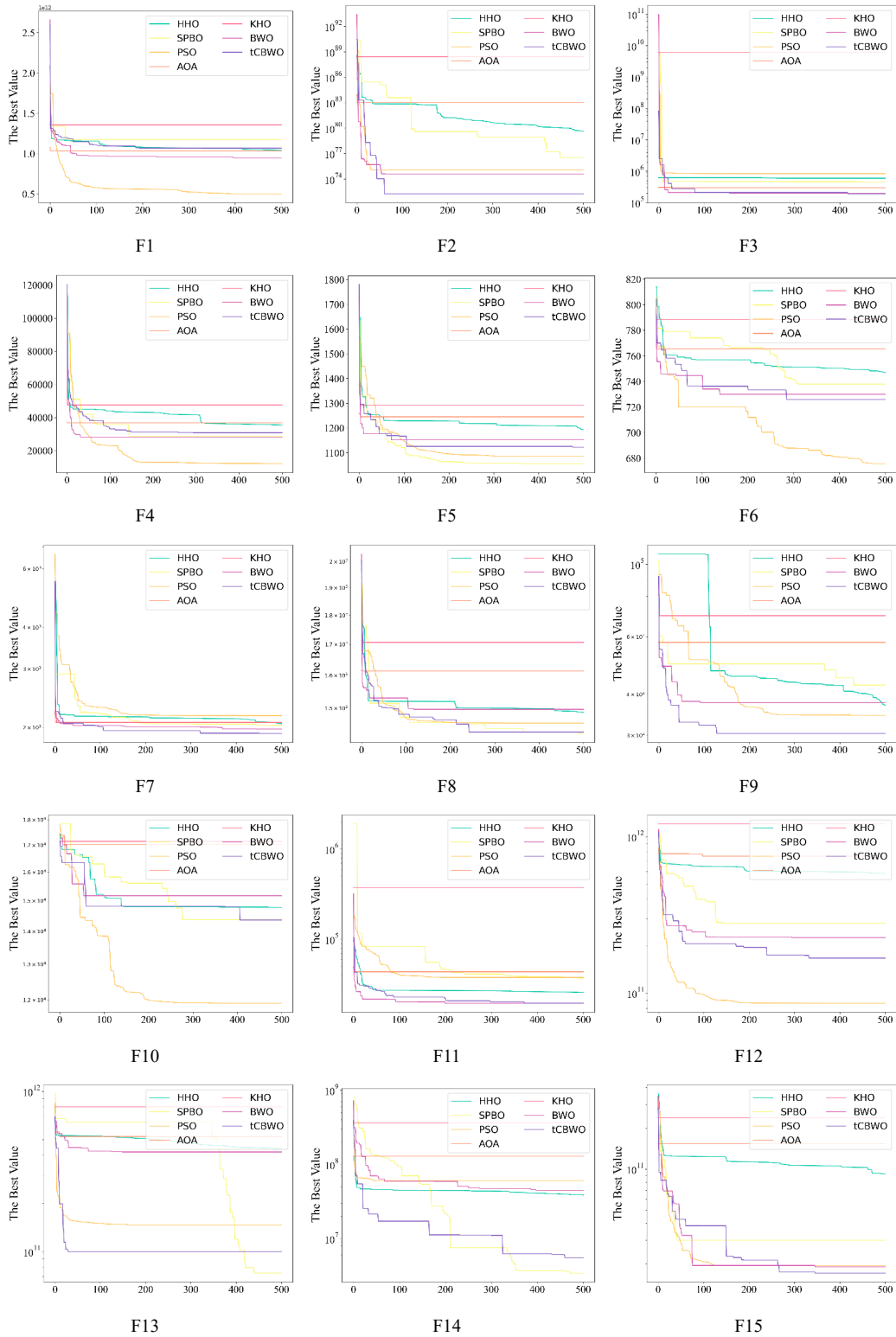


Figure 6. Convergence curves of algorithm on CEC2017 (dim = 30).



Continued on next page

Figure 7. Convergence curves of algorithm on CEC2017 (dim = 50).

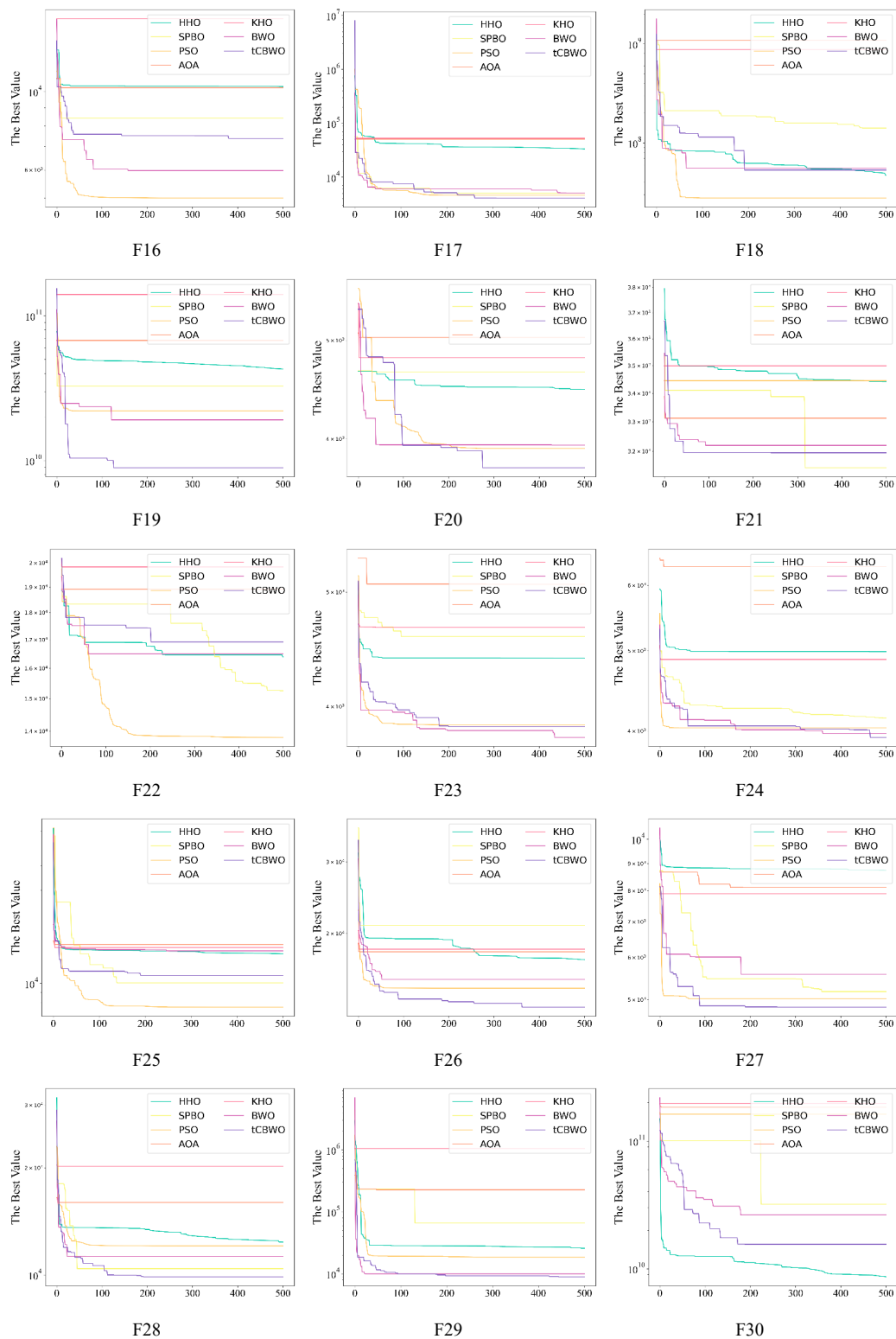


Figure 7. Convergence curves of algorithm on CEC2017 (dim = 50).

Table 5. Wilcoxon rank sum test results of CEC2017.

| Function | Dimension | P-value | | | | | |
|----------|-----------|----------|-----------------|-----------------|----------|----------|-----------------|
| | | HHO | SPBO | PSO | AOA | KHO | BWO |
| F1 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 |
| | 30 | 1.95E-13 | 9.16E-01 | 4.44E-09 | 1.78E-15 | 1.78E-15 | 9.87E-04 |
| | 50 | 1.24E-13 | 1.29E-10 | 2.43E-13 | 1.78E-15 | 1.78E-15 | 1.19E-02 |
| F2 | 10 | 1.78E-15 | 8.88E-15 | 1.24E-14 | 1.78E-15 | 1.78E-15 | 3.38E-04 |
| | 30 | 2.63E-11 | 5.85E-01 | 8.56E-01 | 1.78E-15 | 1.78E-15 | 3.29E-02 |
| | 50 | 1.24E-13 | 1.11E-03 | 3.21E-02 | 1.78E-15 | 1.78E-15 | 6.21E-02 |
| F3 | 10 | 1.78E-15 | 3.55E-15 | 5.31E-09 | 1.78E-15 | 1.78E-15 | 1.13E-07 |
| | 30 | 1.78E-15 | 8.88E-15 | 1.24E-13 | 1.78E-15 | 1.78E-15 | 1.76E-09 |
| | 50 | 1.62E-10 | 4.22E-12 | 4.22E-12 | 1.78E-15 | 1.78E-15 | 1.14E-04 |
| F4 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 2.97E-02 |
| | 30 | 1.35E-12 | 6.25E-01 | 6.22E-08 | 1.78E-15 | 1.78E-15 | 1.29E-02 |
| | 50 | 1.78E-15 | 1.33E-03 | 1.35E-11 | 1.78E-15 | 5.86E-14 | 5.72E-01 |
| F5 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 3.38E-04 |
| | 30 | 2.00E-07 | 2.61E-01 | 2.02E-11 | 1.78E-15 | 1.78E-15 | 8.09E-03 |
| | 50 | 9.67E-03 | 6.78E-02 | 5.94E-03 | 1.78E-15 | 1.78E-15 | 8.41E-01 |
| F6 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-14 | 1.78E-15 | 1.78E-15 | 6.71E-08 |
| | 30 | 3.09E-08 | 1.03E-02 | 4.88E-07 | 1.78E-15 | 1.78E-15 | 1.41E-02 |
| | 50 | 6.53E-03 | 9.12E-03 | 7.80E-08 | 1.78E-15 | 3.55E-15 | 5.92E-01 |
| F7 | 10 | 1.78E-15 | 1.78E-15 | 3.38E-14 | 1.78E-15 | 1.78E-15 | 1.98E-06 |
| | 30 | 3.40E-11 | 8.01E-10 | 3.42E-01 | 1.78E-15 | 1.78E-15 | 9.91E-02 |
| | 50 | 3.09E-12 | 3.38E-14 | 8.16E-04 | 1.78E-15 | 3.55E-15 | 5.72E-01 |
| F8 | 10 | 1.78E-15 | 5.86E-14 | 2.43E-13 | 1.78E-15 | 1.78E-15 | 1.05E-02 |
| | 30 | 2.43E-08 | 1.78E-03 | 1.66E-01 | 1.78E-15 | 1.78E-15 | 1.26E-02 |
| | 50 | 4.46E-03 | 1.03E-03 | 1.37E-02 | 1.78E-15 | 1.78E-15 | 2.44E-03 |
| F9 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.98E-03 |
| | 30 | 2.90E-02 | 1.69E-05 | 2.08E-01 | 1.78E-15 | 1.78E-15 | 6.74E-01 |
| | 50 | 1.56E-04 | 1.28E-01 | 6.13E-03 | 1.78E-15 | 1.78E-15 | 1.15E-02 |
| F10 | 10 | 1.78E-15 | 5.33E-15 | 7.94E-13 | 1.78E-15 | 1.78E-15 | 7.40E-02 |
| | 30 | 6.51E-05 | 1.69E-01 | 3.40E-11 | 1.78E-15 | 1.78E-15 | 2.55E-02 |
| | 50 | 8.63E-05 | 2.95E-01 | 2.83E-05 | 1.78E-15 | 3.55E-15 | 9.13E-02 |
| F11 | 10 | 1.78E-15 | 8.88E-15 | 7.64E-14 | 1.78E-15 | 1.78E-15 | 8.02E-02 |
| | 30 | 1.78E-15 | 2.21E-04 | 8.01E-10 | 1.78E-15 | 1.78E-15 | 2.11E-01 |
| | 50 | 1.78E-15 | 1.81E-10 | 1.26E-01 | 1.78E-15 | 1.78E-15 | 4.88E-07 |
| F12 | 10 | 1.78E-15 | 9.77E-14 | 1.02E-10 | 1.78E-15 | 1.78E-15 | 2.19E-05 |
| | 30 | 2.63E-12 | 2.01E-01 | 3.80E-06 | 1.78E-15 | 1.78E-15 | 6.53E-03 |
| | 50 | 8.88E-15 | 2.19E-01 | 1.45E-02 | 1.78E-15 | 1.78E-15 | 1.63E-04 |
| F13 | 10 | 6.35E-07 | 5.71E-12 | 2.99E-11 | 1.78E-15 | 1.78E-15 | 1.98E-03 |
| | 30 | 3.38E-06 | 7.16E-01 | 5.40E-01 | 1.78E-15 | 1.78E-15 | 3.99E-07 |
| | 50 | 8.84E-12 | 7.84E-03 | 1.96E-02 | 1.78E-15 | 1.78E-15 | 6.51E-05 |

Continued on next page

| Function | Dimension | P-value | | | | | |
|----------|-----------|-----------------|-----------------|-----------------|----------|----------|-----------------|
| | | HHO | SPBO | PSO | AOA | KHO | BWO |
| F14 | 10 | 1.91E-01 | 4.86E-09 | 1.19E-02 | 1.78E-15 | 1.78E-15 | 9.96E-03 |
| | 30 | 7.15E-11 | 8.11E-01 | 7.86E-05 | 1.78E-15 | 5.33E-15 | 7.16E-01 |
| | 50 | 4.22E-12 | 4.76E-03 | 4.46E-03 | 1.78E-15 | 1.78E-15 | 2.30E-02 |
| F15 | 10 | 3.71E-09 | 3.61E-12 | 9.08E-11 | 1.78E-15 | 1.78E-15 | 1.33E-02 |
| | 30 | 1.24E-13 | 2.01E-01 | 3.73E-07 | 1.78E-15 | 1.78E-15 | 1.62E-02 |
| | 50 | 3.00E-13 | 3.93E-01 | 2.21E-04 | 1.78E-15 | 1.78E-15 | 8.24E-07 |
| F16 | 10 | 1.78E-15 | 1.78E-15 | 1.56E-13 | 1.78E-15 | 1.78E-15 | 1.94E-04 |
| | 30 | 7.66E-12 | 2.95E-01 | 5.34E-08 | 1.78E-15 | 1.78E-15 | 3.88E-01 |
| | 50 | 1.78E-15 | 7.45E-01 | 4.22E-12 | 1.78E-15 | 1.78E-15 | 1.43E-03 |
| F17 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 4.61E-02 |
| | 30 | 2.25E-08 | 8.11E-01 | 2.07E-08 | 1.78E-15 | 1.78E-15 | 2.23E-02 |
| | 50 | 1.78E-15 | 3.19E-06 | 2.90E-03 | 1.78E-15 | 1.78E-15 | 1.98E-05 |
| F18 | 10 | 2.82E-01 | 2.81E-10 | 2.24E-12 | 1.78E-15 | 1.78E-15 | 2.75E-02 |
| | 30 | 7.24E-10 | 2.55E-02 | 6.35E-02 | 1.78E-15 | 1.78E-15 | 9.32E-02 |
| | 50 | 3.09E-12 | 9.96E-03 | 4.91E-03 | 1.78E-15 | 1.78E-15 | 2.20E-03 |
| F19 | 10 | 1.78E-15 | 8.88E-15 | 1.56E-13 | 1.78E-15 | 1.78E-15 | 1.86E-06 |
| | 30 | 3.55E-15 | 3.55E-15 | 7.84E-03 | 1.78E-15 | 1.78E-15 | 5.73E-04 |
| | 50 | 1.78E-15 | 2.65E-01 | 9.15E-04 | 1.78E-15 | 1.78E-15 | 6.73E-06 |
| F20 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 2.26E-10 |
| | 30 | 1.78E-05 | 4.27E-07 | 6.39E-01 | 1.78E-15 | 1.78E-15 | 9.77E-01 |
| | 50 | 1.78E-15 | 4.79E-06 | 2.31E-05 | 1.78E-15 | 1.78E-15 | 1.63E-01 |
| F21 | 10 | 1.78E-15 | 1.78E-15 | 1.35E-12 | 1.78E-15 | 1.78E-15 | 6.33E-03 |
| | 30 | 3.13E-10 | 2.41E-01 | 5.32E-10 | 1.78E-15 | 3.55E-15 | 3.31E-03 |
| | 50 | 1.95E-13 | 3.09E-01 | 4.96E-11 | 1.78E-15 | 1.78E-15 | 2.24E-02 |
| F22 | 10 | 1.78E-15 | 1.78E-15 | 1.61E-12 | 1.78E-15 | 1.78E-15 | 3.86E-11 |
| | 30 | 1.35E-12 | 8.97E-09 | 3.92E-08 | 1.78E-15 | 1.78E-15 | 8.11E-01 |
| | 50 | 1.33E-03 | 5.98E-01 | 1.19E-09 | 1.78E-15 | 1.78E-15 | 6.67E-01 |
| F23 | 10 | 1.78E-15 | 1.78E-15 | 8.88E-15 | 1.78E-15 | 1.78E-15 | 1.09E-04 |
| | 30 | 4.44E-14 | 3.09E-09 | 3.09E-01 | 1.78E-15 | 1.78E-15 | 1.37E-02 |
| | 50 | 1.78E-15 | 3.62E-08 | 9.09E-01 | 1.78E-15 | 1.78E-15 | 4.91E-03 |
| F24 | 10 | 1.78E-15 | 3.88E-10 | 1.45E-09 | 1.78E-15 | 1.78E-15 | 8.40E-02 |
| | 30 | 2.49E-14 | 5.07E-06 | 6.67E-01 | 1.78E-15 | 3.55E-15 | 7.08E-02 |
| | 50 | 1.24E-14 | 8.24E-07 | 1.01E-01 | 1.78E-15 | 3.55E-15 | 8.85E-03 |
| F25 | 10 | 1.78E-15 | 1.78E-15 | 1.78E-14 | 1.78E-15 | 1.78E-15 | 2.03E-04 |
| | 30 | 2.49E-14 | 5.07E-06 | 1.95E-13 | 1.78E-15 | 3.55E-15 | 1.84E-03 |
| | 50 | 2.24E-12 | 7.74E-01 | 1.24E-14 | 1.78E-15 | 3.55E-15 | 3.78E-01 |
| F26 | 10 | 1.78E-15 | 1.78E-15 | 5.33E-15 | 1.78E-15 | 1.78E-15 | 6.67E-01 |
| | 30 | 7.66E-12 | 1.09E-04 | 2.12E-02 | 1.78E-15 | 1.78E-15 | 3.29E-02 |
| | 50 | 1.78E-15 | 9.37E-06 | 3.00E-01 | 1.78E-15 | 1.78E-15 | 7.24E-02 |
| | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.90E-12 |

Continued on next page

| Function | Dimension | P-value | | | | | |
|----------|-----------|----------|-----------------|-----------------|----------|----------|----------|
| | | HHO | SPBO | PSO | AOA | KHO | BWO |
| F27 | 30 | 3.55E-15 | 6.46E-04 | 2.86E-04 | 1.78E-15 | 1.78E-15 | 3.05E-02 |
| | 50 | 1.78E-14 | 9.09E-01 | 3.72E-01 | 1.78E-15 | 1.78E-15 | 4.04E-03 |
| | 10 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 1.09E-02 |
| F28 | 30 | 1.24E-13 | 5.94E-02 | 4.43E-01 | 1.78E-15 | 1.78E-15 | 2.48E-02 |
| | 50 | 3.55E-15 | 7.86E-05 | 4.03E-06 | 1.78E-15 | 1.78E-15 | 1.30E-05 |
| | 10 | 1.78E-15 | 1.78E-15 | 1.61E-12 | 1.78E-15 | 1.78E-15 | 8.01E-10 |
| F29 | 30 | 2.63E-12 | 6.63E-02 | 1.37E-05 | 1.78E-15 | 1.78E-15 | 4.40E-02 |
| | 50 | 1.78E-15 | 8.59E-03 | 2.86E-04 | 1.78E-15 | 1.78E-15 | 8.59E-03 |
| | 10 | 1.78E-15 | 1.78E-15 | 2.63E-12 | 1.78E-15 | 1.78E-15 | 2.03E-04 |
| F30 | 30 | 4.49E-13 | 9.24E-01 | 1.78E-15 | 1.78E-15 | 1.78E-15 | 4.89E-04 |
| | 50 | 3.00E-13 | 9.71E-02 | 3.46E-05 | 1.78E-15 | 1.78E-15 | 3.31E-03 |

6.2. tCBWO-DPR testing

To evaluate the effectiveness of the algorithms in this paper, the article presents an experimental analysis of the tCBWO-DPR clustered routing algorithm. The study analyzes the individual impact of each improvement on the wireless sensor network and performs a comparative analysis with other algorithms.

In order to verify the performance of the algorithm, an ablation experiment and a comparison experiment were designed, respectively. In the ablation experiment, this paper compares and analyzes the newly proposed tCBWO-DPR with BWO-DPR and tCBWO-PR. Among them, BWO-DPR is composed of the original BWO and the improved prim algorithm DPR, and tCBWO-PR is a clustering routing algorithm composed of the improved tCBWO and the original Prim algorithm. In the comparison experiment, this paper combines the LEACH [7], HFAPSO [19], SCA-Levy [21], and HPR-LEACH [40] algorithms for simulation verification experiments.

6.2.1. Evaluating indicators of WSNs

In order to evaluate the performance of the tCBWO-DPR algorithm, this paper makes an experimental analysis of the KSAOA clustering routing algorithm and analyzes the influence of each contribution of the article on the WSN network and the influence of network parameters on the WSN. The meanings of the evaluation indicators are as follows:

1) FND refers to the number of turns in which the first node dies, which is an important indicator to evaluate the stability of wireless sensor networks. The death of a node will lead to structural changes in the monitoring system. In theory, the larger the FND, the stronger the stability of WSNs.

2) HND refers to half of nodes dead, which is an important parameters for evaluating network life and stability. When half of the nodes die, the monitoring network can still maintain a certain monitoring capability, so the larger the HND, the better.

3) LND refers to the number of rounds in which the last node dies. When the last node dies, the life cycle of the WSN ends and the monitoring ability is completely lost. Therefore, the larger the LND, the better, and the smaller the difference from the FND, the better, indicating that the network is more stable and reliable.

4) Node energy variance is an important parameter reflecting the load balance of wireless sensor networks, and its size directly reflects the difference in energy consumption of each node. The larger the node energy variance, it means that the network load is not balanced enough. It is given by

$$\sigma_e = \sum_{i=1}^n \frac{(E(i) - E)^2}{n}, \quad (6.1)$$

where n is the number of surviving nodes, E is the average energy of the nodes, and $E(i)$ is the remaining energy of the i^{th} node.

5) The amount of data received by the base station (BS) is the total amount of data received by the WSN network from the sensor nodes during the life cycle. This indicator reflects the data acquisition capability of the network. The larger the amount of received data, the more abundant data the base station can receive. Therefore, the larger the amount of data received, the stronger the data acquisition capability of the network.

6.2.2. Simulation parameter setting

The main evaluation indicators used in the study are explained as shown in Table 6.

Table 6. Simulation parameters.

| Parameters | Value |
|------------------------------------|------------------------------|
| Detection area | $300 \times 300 \text{ m}^2$ |
| Position of base station | (0, 0) |
| Percentage of cluster head | 10% |
| Number of cluster heads nodes | 200 |
| Initial energy of the node | 0.5 J |
| Packet size | 4000 bit |
| Control package size | 50 bit |
| Node Survival Threshold | 0.01 J |
| Electronics energy (E_{elec}) | 50 nJ/bit |
| Energy for free space (E_{fs}) | 10 pJ/bit/m ² |
| Energy for multi-path (E_{mp}) | 0.0013 pJ/bit/m ⁴ |

6.2.3. Distribution of node deaths

As shown in Figure 8(a), node survival in ablation experiments, the FND, HND, and LND distributions of the tCBWO algorithm are 94, 188 and 459 rounds, respectively. Compared with other algorithms, the stability of tCBWO has been greatly improved. Compared to the BWO-DPR and tCBWO-PR algorithms, FND improved by 34 and 31 rounds, respectively. Compared with the BWO-DPR and tCBWO-PR algorithms, HND improved by 16 and 23 rounds, respectively. Compared to the BWO-DPR and tCBWO-PR algorithms, LND improved by 120 and 145 rounds, respectively. Therefore, the algorithm has made a significant improvement in improving network performance.

Comparison of the experimental results in Figure 8(b) shows that the network stabilization period of the tCBWO algorithm is stable at 94 rounds. Compared to the other four algorithms, tCBWO has

better stability. Compared with the LEACH, HPR-LEACH, HFAPSO, and SCA-Lévy algorithms, FND improved by 90, 44, 48 and 36 rounds, respectively. Therefore, compared with the existing algorithms, the algorithm has made a significant improvement in improving the life of wireless sensor networks in terms of network survival.

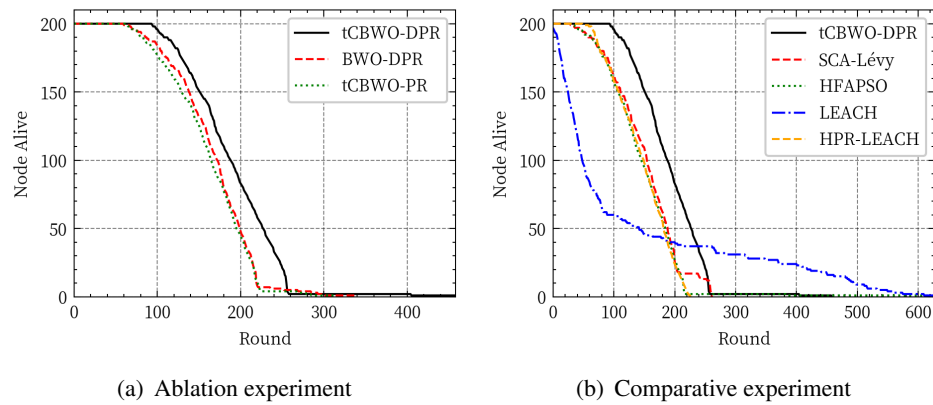


Figure 8. Node survival.

6.2.4. Average residual energy

As shown in Figure 9, by observing the average residual energy curve of nodes, the trend of energy change throughout the network can be understood, and then analyzed. Changes in average residual energy over time show different characteristics. In the ablation experiment, compared with the other two algorithms, the energy drop of the tCBWO-DPR algorithm is more gentle, and the remaining energy is the largest under the same number of turns. In the comparison experiment, the residual energy slope of the tCBWO algorithm in the initial stage is significantly lower than that of the other three algorithms, and in the case of no more than 200 rounds, the average residual energy of the tCBWO algorithm is improved compared with the other three algorithms. It should be noted that the decrease in energy of the tCBWO-DPR algorithm shows a linear trend, indicating that the energy consumption of the network is relatively balanced. On the contrary, LEACH in Figure 9(b) will show a nonlinear downward trend in the medium term, reflecting the instability of network load.

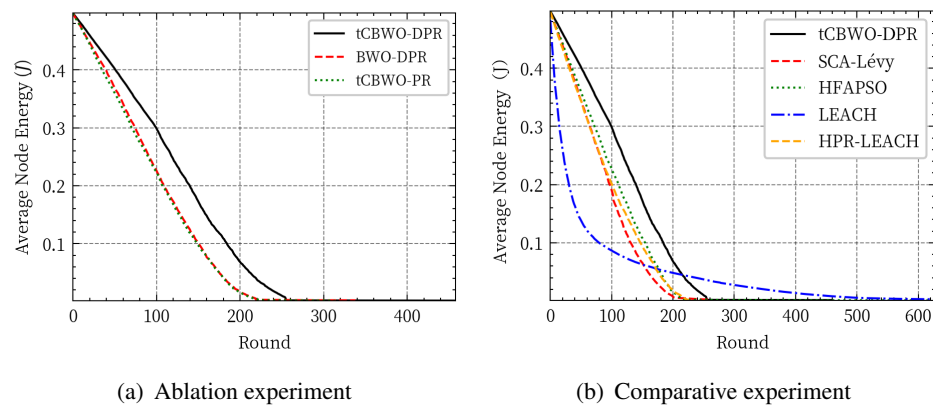


Figure 9. Average energy change of nodes.

6.2.5. Node energy variance

According to the results of the ablation experiment (see Figure 10(a) for details), comparison of the tCBWO algorithm with the other two algorithms shows that the maximum node energy variance of the tCBWO algorithm has increased and performs similarly to the previous two algorithms in the early stage. However, the value of maximum node energy variance is shifted to the right, which delays the death of the first network to some extent.

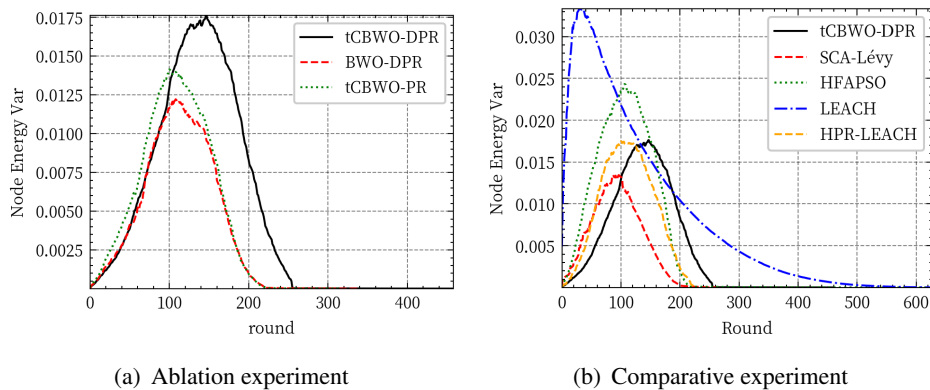


Figure 10. Energy variance variation of nodes.

On the other hand, it can be observed by comparing the Figure 10(b), among the five algorithms tested in the comparison, the tCBWO-DPR algorithm's node energy variance is close to the right side, with the maximum node energy variance of 0.0169. The maximum node energy variance of the LEACH algorithm is 0.0333, which indicates that the tCBWO-DPR algorithm is more effective in balancing the loads of the network nodes, and delays the premature death of some of the nodes, which in turn improves the overall stability of the network. This result further validates the effectiveness of the tCBWO-DPR algorithm in balancing the energy consumption of the network.

6.2.6. Data throughput of the base station

In terms of data transmission, this paper takes the received data from the receiving station as a reference. In the ablation experiment in Figure 11(a), the total amount of data received by the tCBWO-DPR algorithm is 10.55 and 13.14 MB higher than that of the BWO-DPR and tCBWO-PR algorithms, respectively. Both improvements to the algorithm are effective.

Combined with the comparison experiment in Figure 11(b), it can be seen that, in the entire network life cycle of tCBWO, when the data fusion rate is 0.6, the amount of data received by the base station is higher than that of the other three, and the total amount of data transmission reaches 89.34 MB. Compared with LEACH, HPR-LEACH, HFAPSO and SCA-Lévy, there are improvements of 40.49, 21.94, 18.81 and 18.41 MB, respectively, which effectively improves the efficiency of the network.

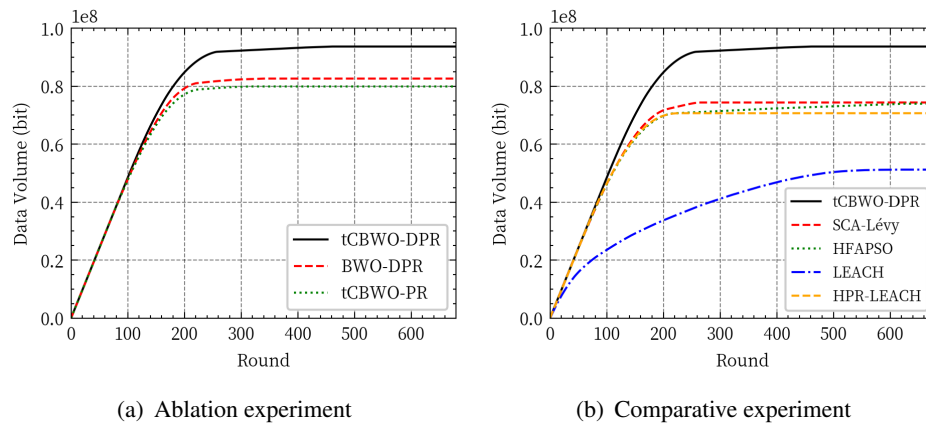


Figure 11. Base station data reception situation.

6.2.7. Average node hops

Observing the average number of node hops for data transmission in Figure 12, the tCBWO-DPR algorithm shows excellent stability in the first two hundred rounds, maintaining between 5 and 7 hops. This indicates that the tCBWO-DPR algorithm is able to effectively control the number of node hops and maintain a relatively short data transmission path during network data transmission. The lower hop count helps to reduce energy consumption and improve network lifetime.

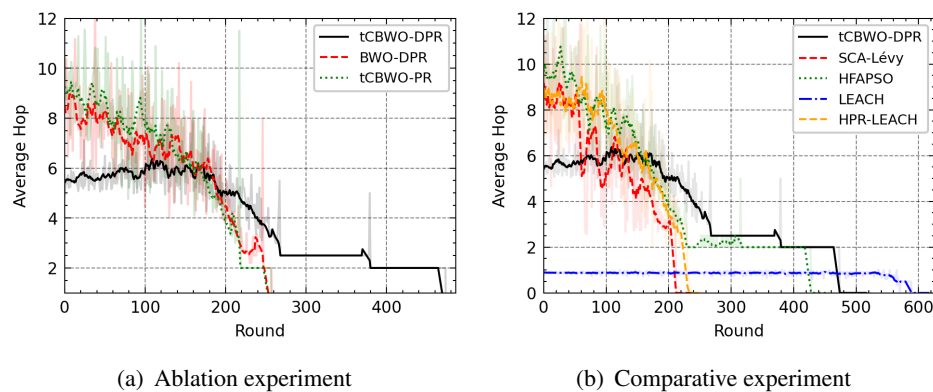


Figure 12. Base station data reception situation.

In the ablation experiment in Figure 12(a), the average hop counts of the tCBWO-PR and BWO-DPR algorithms show a decreasing trend, and the maximum average hop count reaches 12 hops. This trend may reflect that these two algorithms have unreasonable inter-cluster routing establishment during data transmission, which leads to longer paths. An excessive number of hops may trigger energy wastage, which directly affects the lifetime of the network.

In the comparison experiment in Figure 12(b), the tCBWO-DPR algorithm exhibits relatively more stable and uniform node hop counts in the first 200 rounds, and the nodes have relatively fewer hops. This may indicate that the tCBWO-DPR algorithm is able to maintain a relatively low number of node hops under different network conditions, which in turn reduces the delay of data transmission. Compared with the other four algorithms, the HFAPSO algorithm has the most unstable performance in terms of

node hop count, which can reach up to 12 hops, while, in comparison, the tCBWO algorithm has a node hop count of only 7 hops. This means that the tCBWO algorithm is able to better control the number of node hops while reducing the data transmission path, effectively reducing the delay of data transmission.

7. Conclusions

In this study, we propose an improved WSN clustered routing algorithm based on the beluga whales optimization (BWO) algorithm to improve the energy usage efficiency of sensor networks.

First, we introduce tCBWO, a novel optimization algorithm derived from BWO. tCBWO enhances the global exploration capability of BWO by introducing the cosine dynamic boundary strategy, which can achieve a wider exploration of the search space. In addition, SAOA enhances the local exploitation capability of BWO by a dynamic t-distribution factor, which helps the algorithm converge to the optimal solution faster and more accurately. The improved algorithm is tested with CEC2017 in different dimensions, and the test results show that SAOA has improved in convergence and stability in different functions and dimensions compared to other algorithms.

In addition, this paper presents a novel clustering routing algorithm for WSNs called the tCBWO-DPR algorithm. In the process of cluster head election, we introduce a fitness function based on node capacity and positional relationship and use the tCBWO algorithm to select cluster heads. In addition, we also consider the energy consumption characteristics of WSNs and introduce a distance threshold to improve the performance of the Prim algorithm. We verify the performance of the algorithm through ablation and comparative experiments and show that the tCBWO-DPR algorithm outperforms other algorithms in the stability and convergence ability of the cluster head election process. This algorithm has important practical application value and can provide new ideas and methods for the research and application of WSNs.

The heuristic algorithm using tCBWO-DPR is used to construct the evaluation cluster head evaluation function based on the node energy information and node location relationship. Inter-cluster transmission paths are built by heuristic algorithms to enhance the stability and lifetime of the network. However, the complexity of this strategy increases with the expansion of the network scale, which poses a greater challenge for sensors. The planning of node data transmission paths remains a challenging task in the later stages of network planning, especially in larger networks. In future research, we plan to apply heuristic algorithms such as tCBWO to data preprocessing of neural networks, such as with graph neural networks (GNNs) to construct node classification and data transmission path models [41]. This approach may cause high computational cost in the initial stage, but, by pre-training the network model, it can improve the operation efficiency in the real environment, reduce the resource consumption, and automatically adjust the routing scheme according to the network size. In addition, by improving the allocation of spectrum resources for wireless sensor networks can also be able to improve the lifetime and efficiency of the network and reduce resource consumption [42, 43].

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work was supported by National Key R & D Program of China (2021YFB3901400), Natural Science Foundation of Chongqing municipality (2022NSCQ-MSX4084), Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJZD-M202201204, KJZD-M202301203, KJQN202201239, KJQN202101233), Open Fund of Chongqing Key Laboratory of Geo-environment Monitoring and Disaster Early Warning of Three Gorges Reservoir Area (MP2020B0202), Science and Technology Innovation Smart Agriculture Project of Wanzhou District Science and Technology Bureau (2022-17), Chongqing Social Science Planning Project (2019PY52).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Madakam, V. Lake, Internet of Things (IoT): A literature review, *J. Comput. Commun.*, **3** (2015), 164. <https://doi.org/10.4236/jcc.2015.35021>
2. K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, G. Saravanan, A review paper on wireless sensor network techniques in Internet of Things (IoT), *Mater. Today Proc.*, **51** (2022), 161–165. <https://doi.org/10.1016/j.matpr.2021.05.067>
3. V. J. Hodge, S. O'Keefe, M. Weeks, A. Moulds, Wireless sensor networks for condition monitoring in the railway industry: A survey, *IEEE Trans. Intell. Transp. Syst.*, **16** (2014), 1088–1106. <https://doi.org/10.1109/TITS.2014.2366512>
4. M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, et al., Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review, *Sensors*, **22** (2022), 2087. <https://doi.org/10.3390/s22062087>
5. A. Watt, M. R. Phillips, C. E. A. Campbell, I. Wells, S. Hole, Wireless sensor networks for monitoring underwater sediment transport, *Sci. Total Environ.*, **667** (2019), 160–165. <https://doi.org/10.1016/j.scitotenv.2019.02.369>
6. A. R. Basha, A review on wireless sensor networks: routing, *Wireless Pers. Commun.*, **125** (2022), 897–937. <https://doi.org/10.1007/s11277-022-09583-4>
7. M. Handy, M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, in *4th international workshop on mobile and wireless communications network*, Stockholm, Sweden, (2002), 368–372. <https://doi.org/10.1109/MWCN.2002.1045790>
8. X. Kui, J. Wang, S. Zhang, Energy-balanced clustering protocol for data gathering in wireless sensor networks with unbalanced traffic load, *J. Cent. South Univ.*, **19** (2012), 3180–3187. <https://doi.org/10.1007/s11771-012-1393-7>
9. W. Xiang, N. Wang, Y. Zhou, An energy-efficient routing algorithm for software-defined wireless sensor networks, *IEEE Sens. J.*, **16** (2016), 7393–7400. <https://doi.org/10.1109/JSEN.2016.2585019>
10. F. Lu, W. Chen, W. Feng, H. Bi, 4pl routing problem using hybrid beetle swarm optimization, *Soft Comput.*, (2023), 1–14. <https://doi.org/10.1007/s00500-023-08378-4>

11. T. Yan, F. Lu, S. Wang, L. Wang, H. Bi, A hybrid metaheuristic algorithm for the multi-objective location-routing problem in the early post-disaster stage, *J. Ind. Manage. Optim.*, **19** (2023), 4663–4691. <https://doi.org/10.3934/jimo.2022145>
12. I. Daanoune, B. Abdennaceur, A. Ballouk, A comprehensive survey on leach-based clustering routing protocols in wireless sensor networks, *Ad Hoc Networks*, **114** (2021), 102409. <https://doi.org/10.1016/j.adhoc.2020.102409>
13. D. W. Sambo, B. O. Yenke, A. Förster, P. Dayang, Optimized clustering algorithms for large wireless sensor networks: A review, *Sensors*, **19** (2019), 322. <https://doi.org/10.3390/s19020322>
14. C. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowledge-Based Syst.*, **251** (2022), 109215. <https://doi.org/10.1016/j.knosys.2022.109215>
15. M. Kaedi, A. Bohlooli, R. Pakrooh, Simultaneous optimization of cluster head selection and inter-cluster routing in wireless sensor networks using a 2-level genetic algorithm, *Appl. Soft Comput.*, **128** (2022), 109444. <https://doi.org/10.1016/j.asoc.2022.109444>
16. W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, Maui, HI, USA, (2000), 1–10. <https://doi.org/10.1109/HICSS.2000.926982>
17. W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Trans. Wireless Commun.*, **1** (2002), 660–670. <https://doi.org/10.1109/TWC.2002.804190>
18. A. Shahraki, M. K. Rafsanjani, A. B. Saeid, Hierarchical distributed management clustering protocol for wireless sensor networks, *Telecommun. Syst.*, **65** (2017), 193–214. <https://doi.org/10.1007/s11235-016-0218-7>
19. B. Pitchaimanickam, G. Murugaboopathi, A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks, *Neural Comput. Appl.*, **32** (2020), 7709–7723. <https://doi.org/10.1007/s00521-019-04441-0>
20. X. W. Yu, Y. Li, Y. Liu, H. Yu, Wsn clustering routing algorithm based on hybrid genetic tabu search, *Wireless Pers. Commun.*, **124** (2022), 3485–3506. <https://doi.org/10.1007/s11277-022-09522-3>
21. X. Guo, Y. Ye, L. Li, R. Wu, X. Sun, WSN clustering routing algorithm combining sine cosine algorithm and Lévy mutation, *IEEE Access*, **11** (2023), 22654–22663. <https://doi.org/10.1109/ACCESS.2023.3252027>
22. L. Chang, F. Li, X. Niu, J. Zhu, On an improved clustering algorithm based on node density for wsn routing protocol, *Cluster Comput.*, **25** (2022), 3005–3017. <https://doi.org/10.1007/s10586-022-03544-z>
23. H. Esmaeili, B. M. Bidgoli, V. Hakami, CMML: Combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks, *Appl. Soft Comput.*, **118** (2022), 108477. <https://doi.org/10.1016/j.asoc.2022.108477>
24. K. Debasis, L. D. Sharma, V. Bohat, R. S. Bhadoria, An energy-efficient clustering algorithm for maximizing lifetime of wireless sensor networks using machine learning, *Mobile Networks Appl.*, **28** (2023), 853–867. <https://doi.org/10.1007/s11036-023-02109-7>

25. Z. M. Zahedi, R. Akbari, M. Shokouhifar, F. Safaei, A. Jalali, Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks, *Expert Syst. Appl.*, **55** (2016), 313–328. <https://doi.org/10.1016/j.eswa.2016.02.016>
26. S. Radhika, P. Rangarajan, On improving the lifespan of wireless sensor networks with fuzzy based clustering and machine learning based data reduction, *Appl. Soft Comput.*, **83** (2019), 105610. <https://doi.org/10.1016/j.asoc.2019.105610>
27. P. S. Mehra, M. N. Doja, B. Alam, Fuzzy based enhanced cluster head selection (FBECS) for WSN, *J. King Saud Univ. Sci.*, **32** (2020), 390–401. <https://doi.org/10.1016/j.jksus.2018.04.031>
28. Y. Jin, L. Wang, Y. Kim, X. Yang, EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks, *Comput. Networks*, **52** (2008), 542–562. <https://doi.org/10.1016/j.comnet.2007.10.005>
29. M. Ahsanullah, B. G. Kibria, M. Shakil, *Normal and Student's t Distributions and Their Applications*, Atlantis Press, Paris, **4** (2014). <https://doi.org/10.2991/978-94-6239-061-4>
30. K. T. Lan, C. H. Lan, Notes on the distinction of gaussian and cauchy mutations, in *2008 Eighth International Conference on Intelligent Systems Design and Applications*, Kaohsiung, Taiwan, (2023), 272–277. <https://doi.org/10.1109/ISDA.2008.237>
31. A. Singh, A. Nagaraju, Low latency and energy efficient routing-aware network codingbased data transmission in multi-hop and multi-sink WSN, *Ad Hoc Networks*, **107** (2020), 102182. <https://doi.org/10.1016/j.adhoc.2020.102182>
32. Z. Ramadhan, A. P. U. Siahaan, M. Mesran, Prim and floyd-warshall comparative algorithms in shortest path problem, in *Proceedings of the Joint Workshop KO2PI and The 1st International Conference on Advance and Scientific Innovation*, (2018), 47–58.
33. A. E. Fawzy, M. Shokair, W. Saad, Balanced and energy-efficient multi-hop techniques for routing in wireless sensor networks, *IET Networks*, **7** (2018), 33–43. <https://doi.org/10.1049/iet-net.2017.0063>
34. G. Wu, R. Mallipeddi, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization, *Nanyang Technol. Univ., Singapore, Tech. Rep.*, **10** (2016), 1–18.
35. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. <https://doi.org/10.1016/j.future.2019.02.028>
36. B. Das, V. Mukherjee, D. Das, Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems, *Adv. Eng. Software*, **146** (2020), 102804. <https://doi.org/10.1016/j.advengsoft.2020.102804>
37. Y. Shi, R. C. Eberhart, Empirical study of particle swarm optimization, in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, Washington, DC, USA, **3** (1999), 1945–1950. <https://doi.org/10.1109/CEC.1999.785511>
38. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. <https://doi.org/10.1016/j.cma.2020.113609>

39. J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. <https://doi.org/10.1080/21642583.2019.1708830>
40. H. Li, Leach-HPR: An energy efficient routing algorithm for heterogeneous WSN, in *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Xiamen, China, (2010), 507–511. <https://doi.org/10.1109/ICICISYS.2010.5658274>
41. P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case, *Comput. Commun.*, **196** (2022), 184–194,. <https://doi.org/10.1016/j.comcom.2022.09.029>
42. Q. Zheng, P. Zhao, Y. Li, H. Wang, Y. Yang, Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification, *Neural Comput. Appl.*, **33** (2021), 7723–7745. <https://doi.org/10.1007/s00521-020-05514-1>
43. Q. Zheng, X. Tian, Z. Yu, H. Wang, A. Elhanashi, S. Saponara, DL-PR: Generalized automatic modulation classification method based on deep learning with priori regularization, *Eng. Appl. Artif. Intell.*, **122** (2023), 106082. <https://doi.org/10.1016/j.engappai.2023.106082>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)