

# A Parallel Compact Phasmatodea Population Evolution Algorithm for Energy Efficient Optimal Cluster in Wireless Sensor Networks

Jeng-Shyang Pan<sup>1,2,3</sup>, Meng-Fei Zhang<sup>2</sup>, Shu-Chuan Chu<sup>1,2\*</sup>, Xing-Wei Xu<sup>2</sup>, Geng-Geng Liu<sup>4</sup>

<sup>1</sup> School of Artificial Intelligence/School of Future Technology, Nanjing University of Information Science and Technology, China

<sup>2</sup> College of Computer Science and Engineering, Shandong University of Science and Technology, China

<sup>3</sup> Department of Information Management, Chaoyang University of Technology, Taiwan

<sup>4</sup> College of Computer and Data Science, Fuzhou University, China

jengshyangpan@gmail.com, 202282060064@sdust.edu.cn, scchu0803@gmail.com,

xuxingweijining@163.com, liugenggeng@fzu.edu.cn

## Abstract

In wireless sensor networks, the sensor nodes are powered by batteries and hence the energy is limited. This paper presents an algorithm -- Parallel Compact Phasmatodea Population Evolution (PCPPE), and then Differential Evolution (DE) is incorporated into the PCPPE algorithm, where Phasmatodea Population Evolution (PPE) is a meta-heuristic algorithm that incorporates the characteristics of the natural evolution of the Phasmatodea populations. We compare the proposed algorithm with PPE and five other algorithms on the test set of CEC2013, and the experimental results show that the PCPPE algorithm obtained better results. Finally, the proposed PCPPE algorithm is applied to select the best cluster head for wireless sensor networks (WSN) so as to reduce the energy consumption for WSN. The fitness function considers the average intra-cluster distance, the distance from the cluster head to the base station and the residual energy of the cluster head nodes. Subsequently, the proposed algorithm is compared with LEACH, PSO-C, HFAPSO, and PCFMO. The experimental results show that the proposed algorithm preserves the cluster head's energy consumption more significantly and effectively improves the network lifetime of WSN.

**Keywords:** Phasmatodea Population Evolution algorithm, Wireless sensor networks, Meta-heuristic algorithm, Cluster head selection

## 1 Introduction

As wireless sensor networks (WSN) technology becomes more and more mature, more and more attention is being paid to it by the outside world [1-2]. WSN is composed of a large number of randomly allocated wireless sensor nodes that gather the needed data from the surroundings, and transmit this information to the base station. Other than that, sensor nodes also have computing functions, data transmission functions and energy supply

functions. The energy of WSN is limited due to the small amount of energy that can be stored in the batteries and the difficulty of replenishing or replacing the batteries. Many techniques have also been proposed in recent years to improve the performance of WSN [3-4].

Clustering [5] is a technology that can improve the stability and network life of WSN. It divides the WSN into clusters, and designates one node in the cluster as the head of the cluster, called the cluster head, which manages the nodes in each cluster. The cluster head is used to manage the data transfer and data aggregation in each small cluster, then the data is sent to the remote base station. Therefore, cluster heads need to be nodes with more remaining energy. Otherwise, limited resources will greatly reduce the performance of WSN. Energy efficient optimal cluster is to select clusters that allow WSN to consume less energy as well as survive longer, the most critical of which is the selection of cluster heads. Since cluster head selection is also an NP-hard problem, meta-heuristic algorithms are highly necessary for the clustering process. In order to solve this problem, this paper aims to work out an effective cluster-based routing strategy to improve the routing efficiency and extend the network lifetime.

A meta-heuristic algorithm is an algorithm for optimization problems. They will be classified into four algorithm domains based on inspiration: swarm intelligence, evolutionary algorithms, physics-based algorithms, and human-based algorithms [6]. Algorithms based on group intelligence inspiration include Particle Swarm Optimization (PSO) [7-9], Ant Colony Optimization (ACO) [10-11], Grey Wolf Optimization (GWO) [12], Firefly Algorithm (FA) [13], Fish Migration Optimization (FMO) [14], Gannet Optimization Algorithm (GOA) [15], Harris Hawks Optimization (HHO) [16], Salp Swarm Algorithm (SSA) [17] and so on [18], which are derived from the behaviour of these creatures. Algorithms based on physics inspiration include Simulated Annealing (SA) [19], Thermal Exchange Optimization (TEO) [20], Black Hole Algorithm (BH) [21], Charged System Search (CSS) [22], Equilibrium Optimizer (EO) [23] and so on. These algorithms are generated by observing physical phenomena in nature. Evolutionary algorithms include

\*Corresponding Author: Shu-Chuan Chu; Email: scchu0803@gmail.com

DOI: <https://doi.org/10.70003/160792642025112606001>

Genetic Algorithm (GA) [24-25], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [26] and so on. Algorithms based on human inspiration are mainly inspired by phenomena in human society, such as Volleyball Premier League Algorithm (VPL) [27], Teaching-learning-based Optimization (TLBO) [28], League Championship Algorithm (LCA) [29-30] and so on.

In recent years, many excellent algorithms have improved on the classical meta-heuristic algorithms, and new meta-heuristic algorithms have appeared in our field of vision [31-33]. The Phasmatodea Population Evolution algorithm (PPE) [34] is an algorithm that originates from the observation of the characteristics of the evolution of the stick insect population found. After the algorithm was proposed, it was applied to many fields, such as using the PPE algorithm to optimize the downlink power allocation problem in 5G heterogeneous network [35], and to apply the optimization to cloud computing for task scheduling [36], etc. This shows that the PPE algorithm has some ability to achieve good results. The primary objective of this paper is to use the optimized PPE algorithm to select the optimal cluster head to extend the cycle life of WSN. The significant contributions of this paper are as follows:

1. A Parallel Compact Phasmatodea Population Evolution algorithm (PCPPE) algorithm that integrates compact technology and parallel technology into the PPE algorithm is proposed, which reduces memory utilization.
2. Hybridizing the PCPPE algorithm with the DE algorithm can increase the generality of solutions.
3. A fitness function is developed, which considers the cluster heads' remaining energy and the distance between inside and outside clusters.
4. The communication strategy used in this paper is inter-group communication: the optimal and suboptimal values are perturbed and then crossed to generate new particles to replace the optimal values in order to speed up the convergence.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 describes the basic algorithm and theory. Section 4 describes the proposed algorithm. Section 5 presents the simulation results, and section 6 presents a summary.

## 2 Related Work

The first and foremost task of clustering is the selection of cluster head, so that the energy efficient optimal cluster can be selected further. Low Energy Adaptive Clustering Hierarchy (LEACH) [37-38] is a traditional clustering protocol, which comprises two steps: the clustering stage and the data transmission stage. First, all nodes are divided into several small clusters, and subsequently, nodes are randomly selected to become cluster heads. Next is the transmission phase of the data, where the cluster head processes the data from each node before sending it to the base station. However, the residual energy of nodes is not fully considered, which can cause the survival time of WSN to drop drastically. So many more versions of routing protocols have been presented.

Low Energy Adaptive Clustering Hierarchy Centralized (LEACH-C) [39] is an optimized version of the LEACH protocol, which uses the idea of simulated annealing. First, the data of each sensor node is transmitted to the base station. Then the mean energy of all nodes is calculated, and the node with residual energy higher than the average is selected to become a cluster head, and then the cluster is formed. LEACH-C is more globally coordinated than LEACH. R-LEACH [40] is also an optimized version of LEACH. When selecting cluster heads, it also examines the two factors of residual energy and the numeral of optimal cluster heads, which makes it more likely that the cluster head is the node with more remaining energy. FIGWO forms clusters in the same way as R-LEACH, and when selecting the optimal cluster head, its fitness function considers the distance between the cluster head and the base station as well as the residual energy of the node, incorporating these two factors into the linear equation.

The HFAPSO algorithm [41] is a fusion of the PSO algorithm and the Firefly Algorithm, which proposes a reliable and energy-efficient cluster-based routing strategy for WSNs. Its fitness function takes into account two elements: the average distance and the total energy of the cluster head nodes. The PSO-C algorithm [42] thinks about the distance between ordinary nodes and cluster heads when selecting the best cluster head, so that the distance inside the cluster is minimized and also thinks about the remaining energy, which reduces the energy consumed by the WSN. The parallel fish migration optimization algorithm with compact technology (PCFMO) is proposed in [43], which designs a strategy based on the memory principle to improve the speed of finding the optimal cluster head when selecting the energy efficient optimal cluster.

## 3 Basic Algorithms and Theory

### 3.1 The Principle of the Basic Algorithm

The PPE algorithm is an algorithm obtained by observing the survival habits of the stick insect population in nature and their population development process. In this algorithm, the population has two properties: population quantity  $p$  and growth rate  $a$ . The algorithm is mainly designed to find the best global solution, which is the final result found by the algorithm.

In  $n$ -dimensional space, we set the expression of the solution to  $x = [x_1, x_2, x_3 \dots x_n]$ , where  $x_i (1 \leq i \leq n)$  represents a Phasmatodea population, each dimension represents a natural characteristic of that Phasmatodea populations. Generate  $N_p$  populations using Equation (1).

$$x_i = LB + (UB - LB) * rand, i = 1, 2, \dots, n \quad (1)$$

The attributes  $a$  and  $p$  best represent their outcomes as influenced by environmental and other factors. First,  $a$  takes values between 0 and 4. Second,  $p$  is calculated as follows.

$$p_i = 1 / N_p \quad (2)$$

The population evolution trend  $ev$  is mainly guided by the first  $L$  historical best solutions, which are stored in  $Ho$ , where the value of  $L$  is calculated using Equation (3).

$$L = \lfloor \log(N_p) + 1 \rfloor \quad (3)$$

The formula for location renewal of the stick insect population is as follows:

$$x^{t+1} = x^t + ev \quad (4)$$

The updating formula of population quantity is Equation (5).

$$p^{t+1} = a^{t+1} * p^t * (1 - p^t) \quad (5)$$

When the population is in a better condition after updating its position than the population before updating its position, firstly, the population converges to the most recent optimal historical solution. Secondly, the population changes  $ev$  through path dependence. Using Equation (6) to update  $ev$ .

$$ev^{t+1} = (1 - p^{t+1}) * (s(Ho, x^t) - x^t) * c + p^{t+1} * (ev^t + m) \quad (6)$$

where  $s(Ho, x^t)$  represents the solution stored in  $Ho$  nearest to  $x^t$ ,  $m$  represents the change in the population in some characteristic natural condition. When the situation of the population after updating the position is worse than the condition of the population before updating the position, then adopts the idea of simulated annealing. The update the  $ev$  equation as shown in Equation (7).

$$ev^{t+1} = rand * (s(Ho, x^t) - x^t) * c + st * B \quad (7)$$

where  $B$  represents the  $n$ -dimensional random vector and the value of  $st$  is  $(Ub - Lb) * 0.1$ .

Two populations compete when the distance a value between them is less than a distance threshold  $G$ .

$$G = (Ub - Lb) * 0.1 * \frac{Max_{gen} + 1 - t}{Max_{gen}} \quad (8)$$

The number of populations  $x_i$  is calculated using Equation (9).

$$p_i = p_i + a_i * p_i * \left( 1 - p_i - \frac{f(x_j)}{f(x_i)} p_j \right) \quad (9)$$

where  $x_j$  is a population different from  $x_i$ ,  $f(x_j)$  and  $f(x_i)$  are calculated as the values of fitness functions for populations  $x_j$  and  $x_i$ .  $ev$  is calculated using Equation (10).

$$ev^{t+1} = ev^{t+1} + \frac{f(x_j) - f(x_i)}{f(x_j)} * (x_j - x_i) \quad (10)$$

Finally, when the value of the growth rate  $a$  is not within the specified range or the amount of the population is abnormal, the current population will be eliminated.

### 3.2 Energy Consumption Model

In WSN, each node consumes some energy while sending and receiving data from other nodes. Energy plays an important role in the operation of the network as the network life cycle depends on the remaining energy of each sensor node. Therefore, improving energy efficiency and maximizing the network life cycle is the main problem of energy efficient optimal cluster and we need to use a suitable energy model to calculate the energy consumption of the nodes. When sending data, cluster heads require additional energy consumption to aggregate data information, so we use Equation (11) to calculate the energy required to send  $n$ -bit data information from cluster head.

$$E_{Txc} = (E_{elect} + E_{ar}) * n + E_{am} * n * d^2 \quad (11)$$

The energy loss of non-cluster head nodes sending  $n$  bits of data is calculated using Equation (12).

$$E_{Tx} = E_{elect} * n + E_{amp} * n * d^2 \quad (12)$$

where  $E_{Txc}$  represents the energy expended by the transmitter of the cluster head nodes when sending  $n$ -bit information, while  $E_{Tx}$  represents the energy expended by the non-cluster heads when sending messages, and  $E_{elect}$  is the energy consumption required for reception or transmission.  $n$  is the number of bits of the received or sent data information.  $d$  is the distance between the sensor nodes and the corresponding cluster heads. The energy consumption for receiving  $n$ -bit information by cluster head nodes and non-cluster heads is:

$$E_{Rx} = (E_{agr} + E_{re}) * n \quad (13)$$

where  $E_{Rx}$  indicates the energy required by non-cluster head nodes and cluster head nodes to receive  $n$ -bit data information.  $E_{agr}$  indicates the energy expended by the

cluster head node in aggregating individual data, and  $E_{re}$  indicates the energy consumed by the cluster head nodes when receiving individual data.

## 4 The Proposed Algorithms

The algorithm that integrates parallel technology [44-45] and compact technology [46-47] with the PPE algorithm is presented next.

### 4.1 Compact PPE Algorithm

Compact technology is an efficient way to reduce the memory usage of a solution. This is achieved by using a normal distribution to compress each dimension of these  $N$  solutions into two vectors, which are the mean  $\mu$  and variance  $\sigma$ , thus reducing memory usage and increasing the speed of operation. The mean and variance vectors are known as perturbation vectors ( $PV$ ), using Equation (14) to express  $PV$ , where the mean is initialised to 0 and the variance to 10. The compact technique is used to update the entire population by updating the  $PV$ .

$$PV = [\mu, \sigma] \quad (14)$$

$$PDF = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}} \sqrt{\frac{2}{\pi}}}{\left( \operatorname{erf}\left(\frac{\mu+1}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{\mu-1}{\sqrt{2}\sigma}\right) \right) * \sigma} \quad (15)$$

$$CDF = \int_{-1}^x PDF \, dx = \int_{-1}^x \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}} * \sqrt{\frac{2}{\pi}}}{\left( \operatorname{erf}\left(\frac{\mu+1}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{\mu-1}{\sqrt{2}\sigma}\right) \right) \sigma} dx \quad (16)$$

$$y = \sqrt{2}\sigma \operatorname{erf}^{-1} \left( \frac{-\operatorname{erf}\left(\frac{\mu+1}{\sqrt{2}\sigma}\right) - x * \operatorname{erf}\left(\frac{\mu-1}{\sqrt{2}\sigma}\right)}{+\operatorname{erf}\left(\frac{\mu+1}{\sqrt{2}\sigma}\right)} \right) + \mu \quad (17)$$

The Probability Density Function (PDF) is generated by the  $PV$ , where  $\operatorname{erf}$  is the error function. Firstly, the PDF needs to be constructed as a Cumulative Distribution Function (CDF) using the Chebyshev polynomial. Secondly randomly generated numbers between [0,1] are input to the inverse function of the CDF and the output is the solution we need. The PPE algorithm is based on the above steps to generate the required sample, and then the actual population is obtained by using Equation (18).

$$y_{actual} = \frac{(Ub - Lb) * y}{2} + \frac{(Ub - Lb)}{2} \quad (18)$$

where  $y$  is the virtual solution and  $y_{actual}$  is the actual solution. Then use the formula in the PPE algorithm to generate a new particle  $y_{new}$ , compare  $y_{new}$  with the old solution, and separate out the winner  $Wi$  and the loser  $Lo$ . The update equations for  $\mu$  and  $\sigma$  are as follows.

$$(\sigma_i^{t+1})^2 = (\sigma_i^t)^2 + \mu_i^t - \mu_i^{t+1} + \frac{1}{N_p} (Wi_i^2 - Lo_i^2) \quad (19)$$

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p} (Wi_i - Lo_i) \quad (20)$$

### 4.2 Parallel Compact PPE Algorithm

The parallel technique is to divide the original population into several small populations, which are then updated independently using the formula of the PPE algorithm, and then generate new small populations. However, using parallel techniques to generate new results requires us to continuously exchange the results between groups so that better solutions can be generated and better solutions will be searched for faster. This technique can speed up the convergence of the algorithm curve.

We divided the stick insect population into several small groups. All small populations use the compact technique. In other words, expressing the actual population in terms of  $\sigma$  and  $\mu$ .

The communication strategy used in this paper is: firstly, the optimal and suboptimal values of each group are perturbed, then the perturbed optimal and suboptimal values are cross-fetched, secondly, the optimal solution with a small fitness value is selected, and finally the previous optimal value will be replaced.

### 4.3 Add the Idea of DE Algorithm

The purpose of incorporating the DE algorithm into the PCPPE algorithm is to improve the extensiveness of the solution, the smaller fitness value of the solution, and the better jumping out of the local optimum solution. The DE algorithm is a random search algorithm, and there are three stages: variation stage, crossover stage and selection stage. The first stage is the mutation stage with the following equation.

$$V_{i,t} = X_{best,t} + F * (X_{r,t} - X_{i,t}) \quad (21)$$

where  $V_{i,t}$  denotes the  $i$ -th particle of the  $t$ -th generation,  $X_{best,t}$  is the global optimal solution in the  $t$ -th generation,  $F$  is the scaling factor,  $V_{r,t}$  is an individual chosen at random from the other  $N_p - 1$  individuals, and  $V_{i,t}$  is the current solution.

The second stage is the crossover stage, which crosses the solution of the parent and the mutated solution to get new offspring. The last stage is the selection stage, where the greedy selection strategy is used. The pseudocode of the PCPPE algorithm after adding the DE algorithm idea is shown in Algorithm 1.

**Algorithm 1.** Parallel compact PPE algorithm**Input:** The size of a population ( $N_p$ ), number of iterations ( $Max\_It$ ), number of dimensions ( $Dim$ ).**Output:** The optimal solution produced by each iteration ( $best\_fit$ ).

---

```

1: Initialize  $\mu$   $\sigma$  in  $g(j)$ .  $PV$ ,  $\mu = 0$ ,  $\sigma = 10$ 
2: Initialize the number of groups, scaling factor  $F$ , crossover probability  $CR$ 
3: for  $iter = 1: Max\_It$  do
4:   if ( $iter == 2 \parallel iter \% gap == 20$ ) then
5:     Initialize  $g(j).x$  with  $g(j)$ .  $PV$ 
6:     Calculating the fitness function ( $f(x)$ ) for each group, set  $Ho$  and  $g_{best} = g(j)_{best}$ 
7:   end if
8:   for  $j = 1: groups$  do
9:     Use  $g(j).ev_i$  generate new population ( $g(j).x_{new}$ )
10:    Compare  $g(j).x_{new}$  with  $g(j).x_{old}$  to generate  $Wi$  and  $Lo$ 
11:    for  $i = 1: Dim$  do
12:      Update  $\mu$   $\sigma$  via Eq.(19) and Eq.(20)
13:    end for
14:  end for
15:  for  $i = 1: groups$  do
16:    Update  $g(j).x_{new}$  via Eq.(21)
17:    Crossover is then performed to generate offspring, and the final selection of  $x$  with a small fitness value
18:  end for
19:  for  $i = 1: N_p$  do
20:    if  $f(g(j).x_{new}) \leq f(g(j).x_{old})$  then
21:       $g(j).x = g(j).x_{new}$ 
22:      Update  $ev_i$  via Eq.(6)
23:    else
24:      if  $rand < p$  then
25:         $x = x_{new}$ 
26:      end if
27:      Update  $ev_i$  via Eq.(7)
28:    end if
29:    if The distance between the two populations is less than  $G$  then
30:      Update  $ev_i$  via Eq.(10)
31:    end if
32:  end for
33:  for  $j = 1: groups$  do
34:    The optimal and suboptimal values are perturbed and then crossed to generate new particles to replace the optimal values
35:  end for
36: end for

```

---

#### 4.4 Numerical Analysis of the PCPPE Algorithm

This paper uses data tested with the CEC 2013 benchmark functions, of which there are 28 target functions. The experimental tests were conducted on a computer with a 12th Gen Intel (R) Core (TM) i7-12700H 2.30 GHz, Windows 11 operating system and MATLAB R2022a software.

We contrast the PCPPE algorithm with the PPE algorithm, SSA algorithm, HHO algorithm, GWO algorithm, PSO algorithm, and EO algorithm, and the initialization of the parameter values for the other comparison algorithms is shown in Table 1, where the population number used by PCPPE algorithm is 10, which is significantly smaller than that of other algorithms. Each algorithm runs the function 2000 iterations at a time, with a total of 30 independent runs of the function, and the solution space is in the range [-100,100].

We have run the tests for 10 dimensions (D=10), 30 dimensions (D=30), and 50 dimensions (D=50), each of which was run 30 times. Based on the results, we calculate their means and variances and record them in Table 2, Table 3 and Table 4. The bold data values in these tables represent the best values in each function.

The PCPPE algorithm is better than the other algorithms on 15 functions when the dimension is 50, in Table 2. However, the results of the PPE algorithm on F14, F15 and F16 are better than the PCPPE algorithm, and the results of the EO algorithm on F11, F12, F13, F17, F23, F24, F25 and F27 are better than PCPPE algorithm in these functions.

As shown in Table 3, when the dimension is 30, the PCPPE algorithm is better than the other algorithms on 15 functions, but the results of the PPE algorithm on F14, F15 and F16 are better than the PCPPE algorithm, and the results of EO algorithm in these functions F7, F9, F12, F13, F17, F23, F24, F25, F27 and F28 are better than the PCPPE algorithm. The outcomes of the EO algorithm are obviously better than the PCPPE algorithm for the composition functions.

As shown in Table 4, when the dimension is 10 the PCPPE algorithm is better than the other algorithms on 16 functions. But on F10, F16 and F17 the results of the PPE algorithm are better than the PCPPE algorithm and the results of the EO algorithm are better than the PCPPE algorithm on these functions F7, F9, F12, F18, F19, F25 and F28. The results show that the PCPPE algorithm still has some advantages in the case of small populations.

CEC 2013 contains test functions of various dimensions and complexities to test and evaluate the ability of optimization algorithms in solving real problems. From Table 2 to Table 4, it can be seen that the PCPPE algorithm has advantages in both low and high dimensions, showing strong competitiveness, and fully examining the search ability and robustness of the PCPPE algorithm. It can be seen from Table 5 that the  $\alpha$ -value of the Friedman test in all data is greater than 5%. Therefore, it can be concluded the data is considered credible.

Next is the convergence of the algorithm's function, and Figure 1, Figure 2 represents the image of the convergence function of the PCPPE algorithm with other algorithms at 30 dimensions. The  $x$ -axis and  $y$ -axis in Figure 1, Figure 2 represent the number of times the function is evaluated during the iterative process and the global optimum of the algorithm, respectively. It is clearly seen that the convergence of the PCPPE algorithm has been effectively improved. Therefore, the PCPPE algorithm can not only solve different types of function problems effectively, but also the parallel communication strategy realizes a faster convergence speed, and a better solution can be found in a short time. In addition, the compact technology is an effective way to reduce the memory footprint of metaheuristic algorithms. By updating the probabilistic model instead of updating the whole aggregate, the amount of computation is reduced and the algorithm runtime is shortened. The PCPPE algorithm is made to select the optimal cluster head in the next section, thus selecting the energy efficient optimal cluster.

**Table 1.** Parameter design of related algorithms

Algorithm	The initialization of the parameter values
PCPPE	$a = 1.1, N_p = 10, gap = 20, F = 0.5, CR = 0.3, max_{it} = 2000$
PPE	$N_p = 20, a = 1.1, max_{it} = 2000$
SSA	$N_p = 20, max_{it} = 2000$
HHO	$N_p = 20, E_0: [-1,1], max_{it} = 2000$
GWO	$N_p = 20, max_{it} = 2000$
PSO	$c1 = 2, N_p = 20, w = 0.9, c2 = 2, max_{it} = 2000$
EO	$N_p = 20, a1 = 2, a2 = 1, max_{it} = 2000$

**Table 2.** Results of the PCPPE algorithm on the CEC 2013 benchmark functions when D=50

Function	PCPPE		PPE		SSA		HHO		GWO		PSO		EO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	<b>-1381</b>	35.075	-795.9	116.66	70488.4	5565.52	20304	3329.4	28955	22453.2	197347	22453.24	7226.8	3688.3
F2	<b>2E+07</b>	6E+06	1E+08	3E+07	1.5E+09	3.9E+08	4E+08	1E+08	3E+08	2.6E+09	9.4E+09	2.55E+09	9E+07	3E+07
F3	<b>3E+09</b>	1E+09	5E+10	2E+10	1.05E+15	1.91E+15	1E+12	2E+12	2E+11	6.42E+23	1.47E+23	6.42E+23	4E+10	8E+09
F4	<b>69960</b>	4187.2	123292	18636	107501	17588.2	109972	15701	184443	1915816	1266628	1915816	128262	26251
F5	<b>-999.7</b>	0.0664	-489.2	104.89	16472.6	2801.38	3251.6	1466.8	5956.9	44614	129459	44614.01	-314.6	375.1
F6	<b>-812.2</b>	36.176	-542.7	43.156	7411.82	1415.94	565.55	356.19	1033.2	10502.8	43274.1	10502.77	-372.3	172.58
F7	<b>-687.3</b>	21.708	-592.7	67.383	8828.98	8710.87	3038.5	5528.1	-499.9	2.4E+08	9.9E+07	2.39E+08	-683.2	13.115
F8	<b>-678.8</b>	0.0371	-678.7	0.0471	-678.73	0.03827	<b>-678.7</b>	0.0487	-678.7	0.04507	-678.585	0.045075	<b>-678.7</b>	0.0553
F9	<b>-546.1</b>	4.9071	-535.5	5.0542	-521.83	1.475	-524.7	2.6628	-531.8	2.36322	-514.424	2.363221	-538.6	6.767
F10	<b>-495.4</b>	1.0698	117.57	118.28	10334.6	1431.24	2786.4	571.89	2982.5	6768.07	33275.4	6768.075	502.97	256.92
F11	67.384	52.665	197.1	89.902	756.082	47.9859	497.59	62.983	382.68	595.965	2982.84	595.965	<b>-5.978</b>	79.681
F12	323.09	72.582	458.39	124.39	958.197	71.9281	875.59	79.495	531.88	294.857	2454.69	294.8567	<b>210.38</b>	71.669
F13	425.88	75.148	658.6	122.95	1074.06	72.6982	1086.1	78.921	650.29	346.241	2781.61	346.2412	<b>387.43</b>	84.955
F14	8547.8	838.12	<b>8259</b>	753.91	14106.3	592.072	10908	1251.6	14109	489.41	15024.2	489.4103	9121.6	1291.8
F15	13874	1063.4	<b>11272</b>	804.03	15948.9	260.864	13375	957.6	15357	427.23	17393	427.2298	12646	968.22
F16	203.99	0.3678	<b>203.13</b>	0.6802	204.839	0.51009	203.87	0.6785	204.63	0.82028	206.246	0.82028	204.18	0.6195
F17	977.79	82.49	1213.8	117.34	1757.67	52.6894	1667.1	84.635	1433.5	587.612	6296.06	587.6117	<b>895.24</b>	89.996
F18	<b>1038.4</b>	50.183	1492.4	144.24	1855.36	75.7466	1808.8	90.549	1529.8	556.9	6383.33	556.9003	1137.3	99.155
F19	<b>543.64</b>	3.5408	797.4	219.6	406913	100244	17061	8906.1	222840	2E+07	5.1E+07	19862902	4360.5	5992.8
F20	<b>623.33</b>	0.7343	624.72	0.2544	624.992	0.03116	624.88	0.1995	624.75	0.0229	624.996	0.022895	<b>622.88</b>	0.9432
F21	<b>1700.8</b>	156.01	2266.2	657.47	5127.15	87.714	4184.6	129.18	4722.3	1500.07	14074.3	1500.073	3892.4	268.82
F22	<b>9995.5</b>	853.83	12014	1273.9	16673.8	374.699	14160	853.58	16286	433.265	18247.8	433.2653	11013	1525.4
F23	15798	1057	14218	854.56	17418.8	589.937	16430	710.76	16865	383.621	18676.1	383.6206	<b>14022</b>	962.85
F24	1370.3	15.732	1419.8	22.468	1438.28	11.3591	1554.6	77.221	1423.9	239.035	2145.88	239.0351	<b>1359.3</b>	14.279
F25	1508.6	23.309	1638	28.91	1552.05	8.89491	1644.8	29.885	1545.7	3.85561	1662.51	3.855613	<b>1483.8</b>	14.074
F26	<b>1543.9</b>	119.05	1594.4	123.91	1707.1	5.68147	1612.3	128.32	1624.9	463.327	2175	463.3266	1622.5	74.032
F27	3206.7	133.07	3541.3	143.33	3775.7	57.1854	4022.3	247.27	3458.1	968.481	5792.16	968.4808	<b>3134</b>	141.51
F28	<b>4125.9</b>	2353.8	8471.3	1564.2	11553.5	829.773	11694	810.33	8196.7	2755.5	20568.7	2755.5	4861.7	1130.7

**Table 3.** Results of the PCPPE algorithm on the CEC 2013 benchmark functions when D=30

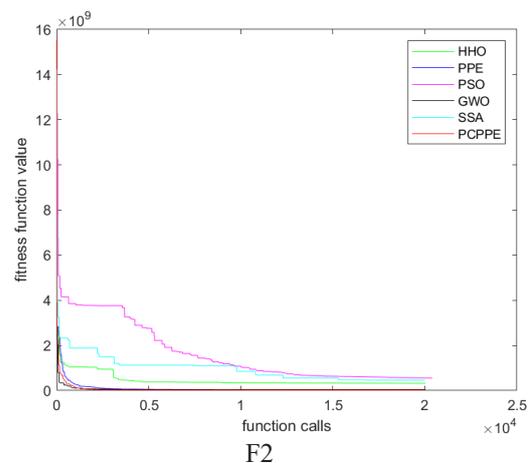
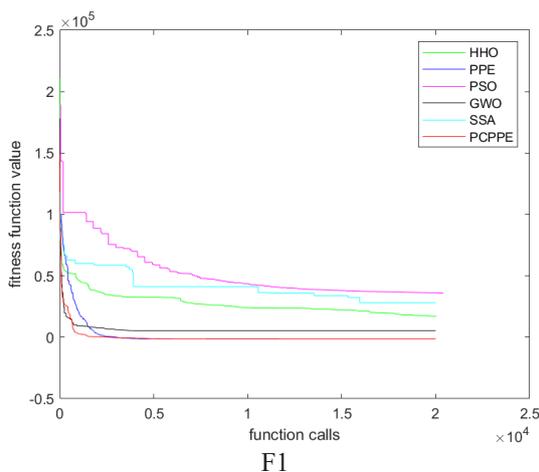
Function	PCPPE		PPE		SSA		HHO		GWO		PSO		EO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	<b>-1399</b>	5.4903	-1350	18.16	43864.2	5985.88	9504.7	2859.86	12243	5395	124647	19419.4	-654.6	901.41
F2	<b>8E+06</b>	3E+06	5E+07	2E+07	7.2E+08	2.1E+08	2.5E+08	1.3E+08	1E+08	6E+07	4.3E+09	1.6E+09	4E+07	1E+07
F3	<b>1E+09</b>	1E+09	2E+10	8E+09	6.63E+13	1.54E+14	1.79E+15	8.90E+15	3E+12	6E+12	1.22E+26	6.64E+26	1E+10	6E+09
F4	<b>46318</b>	5823.9	82788	17157	83603.6	16390.3	63691.9	3155.35	118078	28354	1044536	1390002	81561	21451
F5	<b>-999.9</b>	0.0165	-841	35.677	15634.7	4246.22	3276.4	1642.85	5045.4	4512.5	119125	59794.5	-883	58.78
F6	<b>-843.5</b>	26.001	-736.7	34.507	6331.59	1628.37	636.096	556.645	440.97	715.13	36324.5	10639.2	-738.7	45.237
F7	-671.6	28.329	-647.9	41.772	3828.5	6611.72	344837	712675	205.52	2084.8	4.6E+08	9.5E+08	<b>-695.8</b>	27.258
F8	<b>-679</b>	0.0578	-678.9	0.0595	-678.86	0.05371	-678.89	0.05479	-678.9	0.0613	-678.73	0.08088	-678.9	0.0585
F9	-571.2	2.6852	-565.7	2.8238	-556.46	1.45108	-558.33	3.00244	-563.8	4.5437	-553.54	1.56466	<b>-571.5</b>	3.3455
F10	<b>-498.6</b>	0.1813	-368.2	48.45	5821.65	1069.92	1679.64	716.983	1600.7	611.52	17954.9	4515.73	-210.6	150.82
F11	<b>-327.9</b>	18.117	-73.92	66.711	370.798	65.2323	284.354	71.0229	5.9326	78.498	1706.3	504.13	-213.6	48.268
F12	35.99	64.856	103.84	88.998	461.511	83.5725	522.781	102.635	104.46	70.569	1558.97	269.276	<b>-53.43</b>	54.472
F13	117.68	53.115	320.94	90.734	573.671	89.5349	599.898	96.7945	268.27	80.305	1630.19	362.067	<b>80.614</b>	56.36
F14	3857.4	559.88	<b>3830.7</b>	504.2	8127.83	306.411	5491.95	699.172	7604.7	1049.3	9342.89	387.129	4324.1	941.06
F15	6079.2	1056.6	<b>5565.2</b>	629.81	8528.69	407.281	6383.96	746.813	8037.1	912.47	9778.31	381.945	6181.1	662.27
F16	202.94	0.2706	<b>201.99</b>	0.4489	203.706	0.57471	202.915	0.62489	203.51	0.4752	205.436	0.80826	203.06	0.5416
F17	611.61	39.017	640.7	38.956	1146.1	67.1142	1166.65	79.8461	843.07	99.405	3557.13	400.253	<b>566.78</b>	82.065
F18	<b>716.24</b>	35.645	834.68	62.071	1273.1	61.8154	1293.18	86.5492	1005.4	91.833	3561.09	379.41	754.53	62.09
F19	<b>521.24</b>	2.9253	538.63	11.798	224523	66004.4	7351.13	5571.53	74697	76348	3.5E+07	1.9E+07	783.08	428.63
F20	<b>613.99</b>	0.7172	614.97	0.1008	614.991	0.0266	615	0.00274	615	0.0003	615	4.22E-14	614.13	1.0931
F21	<b>1097</b>	37.027	1332.5	433.3	3166.45	75.415	2416.64	164.579	2776.6	255.72	7875.47	688.037	2032.5	325.37
F22	<b>5149.4</b>	664.28	5927.9	922.88	10072	461.143	8233.1	701.968	9356	945.8	11133.1	477.795	5198.8	856.75
F23	7710.6	820.55	7546.7	720.8	10050.2	319.002	8859.1	721.631	9634.1	522.61	11392.8	356.143	<b>7458.6</b>	1037
F24	1293.6	13.402	1323.5	23.575	1332.33	5.27768	1420.48	113.599	1315.7	14.824	1860.48	234.892	<b>1276.2</b>	11.029
F25	1415.9	13.245	1456.7	16.602	1436.15	5.78393	1477.25	20.892	1431.5	7.0791	1501.41	8.70983	<b>1393.1</b>	9.9001
F26	<b>1406.4</b>	32.808	1518.7	91.751	1560.94	73.4503	1626.76	77.5751	1551.2	69.733	1845.64	366.335	1531.8	73.341
F27	2445	91.443	2549.3	119.56	2783.16	41.1467	2887.24	161.731	2555.5	114.2	4093.71	989.133	<b>2337.7</b>	106.15
F28	3665.6	1350	5138	779.69	7074.83	537.458	7271.22	611.236	5636.7	954.99	14207.2	5547.38	<b>3154.2</b>	688.67

**Table 4.** Results of the PCPPE algorithm on the CEC 2013 benchmark functions when D=10

Function	PCPPE		PPE		SSA		HHO		GWO		PSO		EO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	<b>-1400</b>	0.0001	-1400	0.0723	2325.8	1693.9	-1357	40.622	-293.7	1725.7	22261.49	7395.78	<b>-1400</b>	0.0032
F2	<b>153465</b>	173725	4E+06	2E+06	2E+07	6E+06	9E+06	6E+06	1E+07	6E+06	2.97E+08	3E+08	4E+06	2E+06
F3	<b>2E+08</b>	5E+08	3E+08	4E+08	9E+09	4E+09	5E+09	3E+09	5E+09	4E+09	8.30E+15	2.70E+16	3E+08	5E+08
F4	<b>4373.6</b>	1633.6	13106	6382.9	25211	13686	16928	2681.4	31990	16266	1672846	3188460	18266	8604.5
F5	-1000	0.0012	-999.4	0.3877	350.28	626.98	-681.8	237.01	-525.3	446.16	11562.34	9384.46	<b>-1000</b>	0.0283
F6	<b>-891</b>	19.995	-867.3	32.202	-591.8	126.04	-816.5	51.117	-837.4	36.243	1709.632	1446.56	-865.7	30.811
F7	-765.8	21.653	-745.8	21.756	-666.5	42.36	-499.1	587.94	-724.5	34.436	83136.99	232456	<b>-779.6</b>	13.844
F8	<b>-679.6</b>	0.0735	-679.4	0.0978	-679.4	0.1099	-679.5	0.1441	-679.4	0.1164	-679.078	0.11978	-679.4	0.1331
F9	-594.3	1.4307	-593.6	1.4851	-589.7	0.9329	-589.9	1.388	-592.2	1.5274	-586.693	1.48996	<b>-594.8</b>	1.571
F10	<b>-499.7</b>	0.1531	-496.6	2.1851	-117.9	137.54	-432.6	36.442	-292.6	163.62	2363.831	1107.04	-494.7	5.5302
F10	-398.6	1.1282	<b>-496.6</b>	2.1851	-117.9	137.54	-432.6	36.442	-292.6	163.62	2363.831	1107.04	-494.7	5.5302
F12	-264.8	13.998	-248.7	20.732	-169.9	22.869	-157.1	37.313	-240.9	25.017	53.08597	110.096	<b>-267.8</b>	15.819
F13	<b>-162.9</b>	13.795	-140.5	17.756	-68.44	25.394	-60.01	38.127	-122.4	28.382	136.6511	99.3664	-159.9	18.04
F14	<b>280.58</b>	158.98	504.38	171.69	1929.9	185.81	1020.8	327.48	1335.8	534.91	2648.73	270.035	432.24	254.26
F15	<b>767.63</b>	226.3	1067.1	358.86	1973.4	197.5	1140	276.75	1705.8	469.44	2822.561	177.474	1061.9	316.96
F16	201.29	0.227	<b>200.89</b>	0.3484	201.85	0.384	201.46	0.3634	201.8	0.4062	203.7465	0.92284	201.48	0.3792
F17	346.96	7.557	<b>328.32</b>	7.4243	432.32	21.782	426.41	25.624	371.39	22.865	814.6498	132.54	331.98	8.3934
F18	449.15	8.3532	447.14	14.332	534.35	16.101	529.39	25.465	483.97	23.866	1004.922	147.489	<b>438.06</b>	8.7586
F19	502.97	0.8664	501.65	0.8392	1768.3	1502.7	511.98	4.3676	907.68	1122.7	1166028	1331375	<b>501.42</b>	0.6302
F20	<b>603.16</b>	0.4735	603.97	0.5176	604.23	0.2611	604.43	0.3122	603.96	0.3995	604.9567	0.09699	603.26	0.5298
F21	<b>1090.7</b>	52.395	1100.2	0.0065	1217.9	39.882	1100.5	0.3713	1131.3	47.437	2449.836	529.272	1100.2	0.0004
F22	<b>1251.8</b>	154.02	1674	252.2	3102.5	210.78	2491.1	449.76	2640.3	536.29	3771.097	174.643	1409.5	305.58
F23	<b>2207</b>	289.64	2376.5	452.3	3156.8	211.21	2845.9	320.35	2964	354.86	3904.581	215.915	2172.8	303.05
F24	<b>1208.2</b>	32.725	1218.6	17.574	1230.2	2.6022	1238	13.778	1218.3	24.252	1302.96	61.4373	1215.3	15.446
F25	1318.2	5.7459	1325.1	6.1054	1329.4	2.367	1334.8	5.3274	1323.9	4.2136	1355.235	10.3091	<b>1318.1</b>	3.9209
F26	<b>1360.2</b>	29.958	1408.3	79.374	1404.7	5.0968	1515	51.601	1395.9	24.384	1760.018	403.044	1450.4	81.309
F27	<b>1690.8</b>	79.417	1702.2	43.893	1958.8	38.6	2007.8	166.19	1861.4	88.808	2630.03	287.057	1751.6	90.54
F28	1928	226.43	2000.5	210.78	2442.1	76.293	2375.5	181.54	2167.1	130.35	3648.436	374.404	<b>1883</b>	204.04

**Table 5.** Temperature and wildlife count in the three areas covered by the study

Dimension	Sum of squares	Degree of freedom	Mean squares	p-value
D=10	436784.5	5655	77.2386	0.6496
D=30	435185.2	5655	76.9558	0.4334
D=50	436392.1	5655	77.1692	0.5796



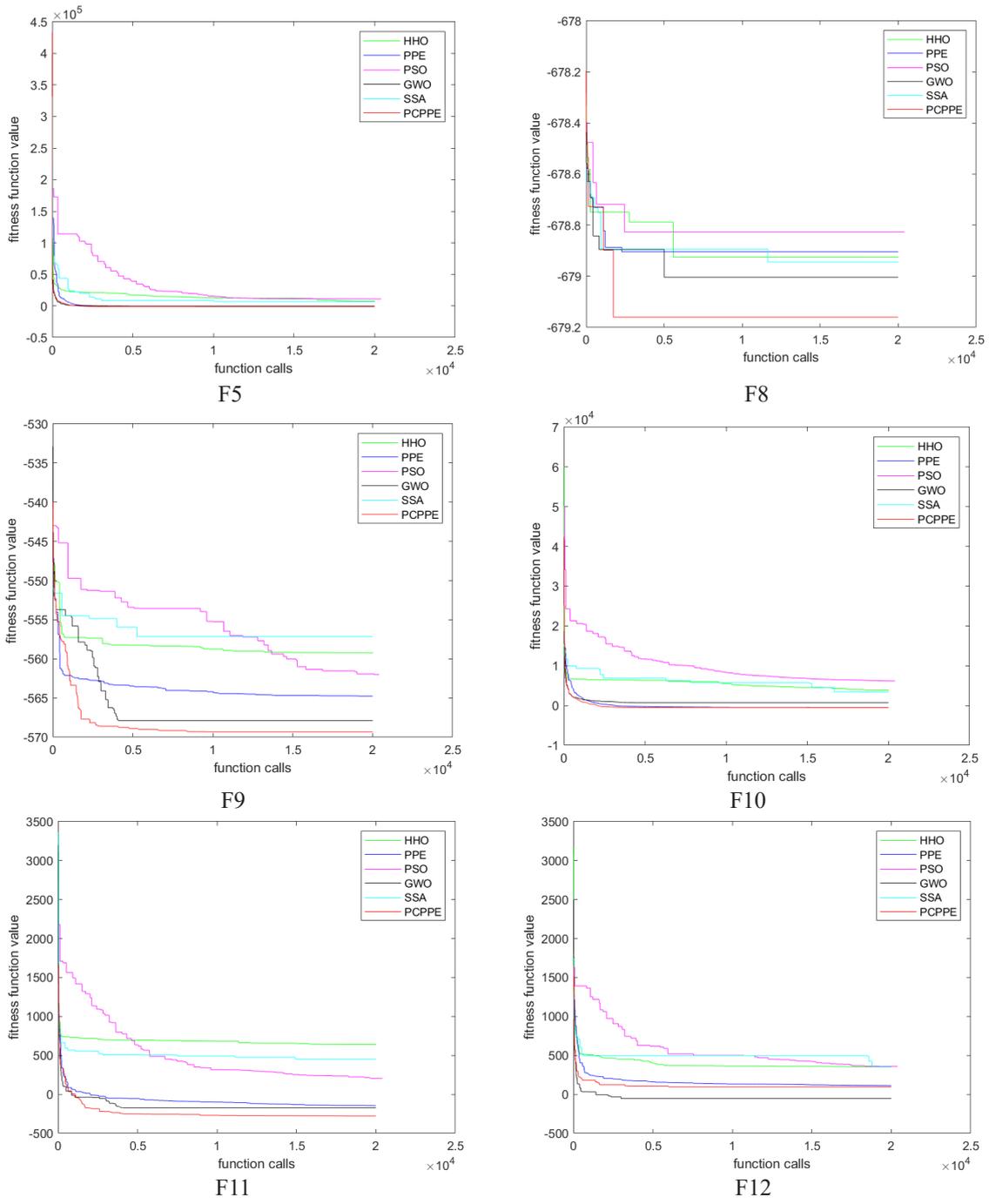
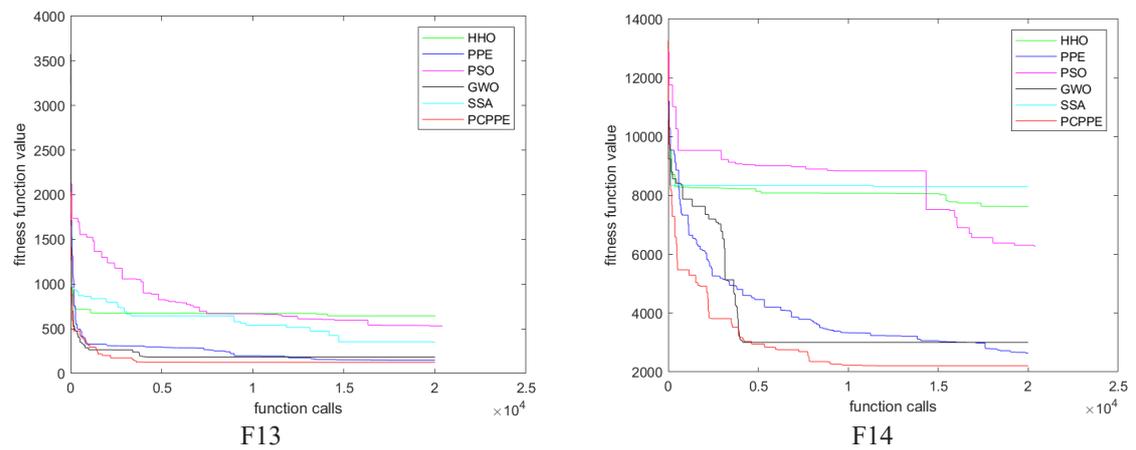


Figure 1. Convergence curve of the algorithm on CEC 2013



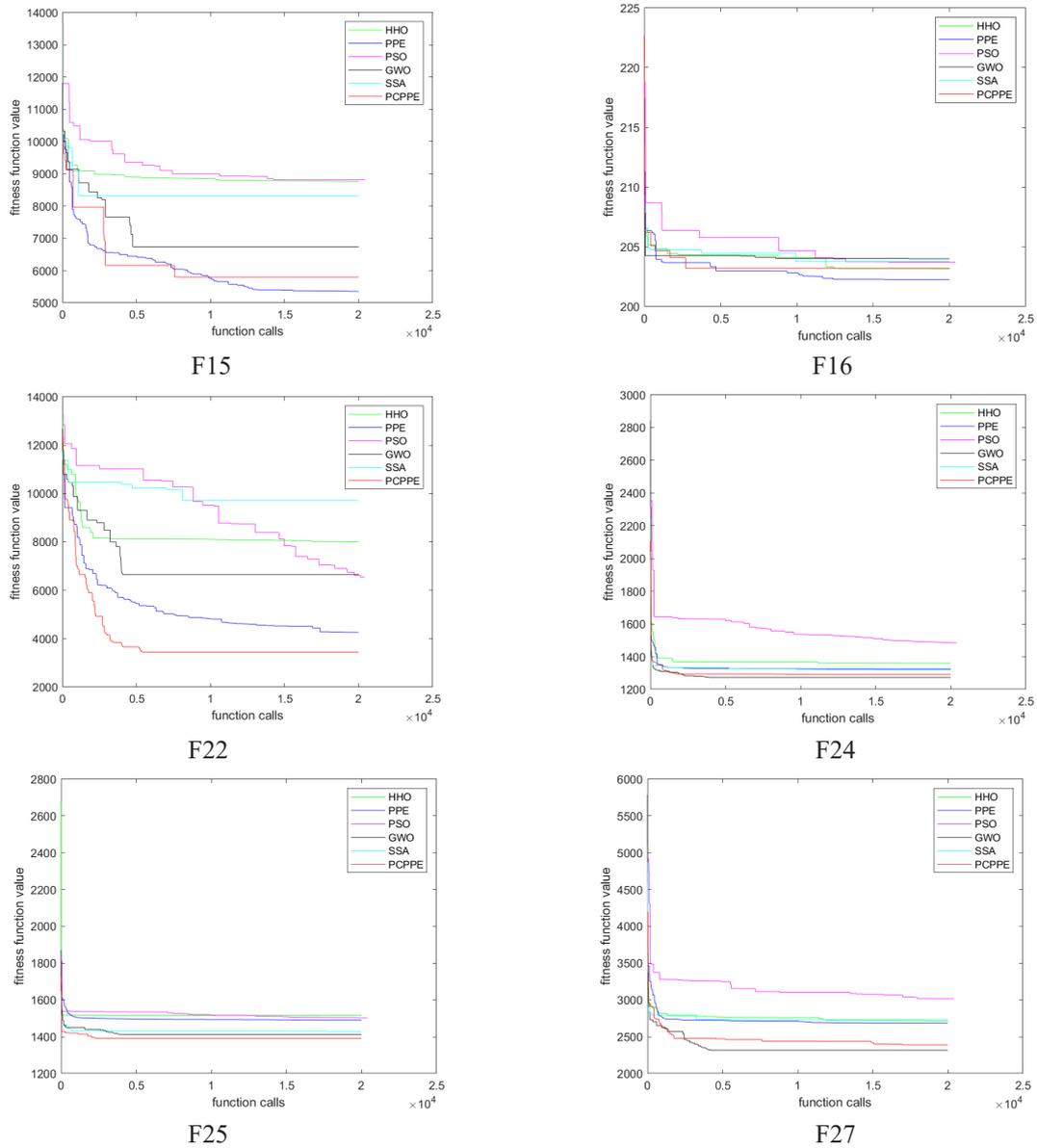


Figure 2. Convergence curve of the algorithm on CEC 2013

## 5 PCPPE Applications in WSN and Simulation Study

### 5.1 Fitness Function

The fitness is derived as follows, because both distance and node residual energy are related to the performance of WSN.

$$f_1 = \max \left\{ \frac{\sum d(nCH_i, CH_j)}{node\_n} \right\} \quad (22)$$

$$f_2 = \max \{ d(BS, CH_j) \} \quad (23)$$

where  $f_1$  represents the maximum value of the average distance between the cluster head nodes and the non-cluster head nodes, where  $\sum d(nCH_i, CH_j)$  is the total distance between the nodes in the cluster and the cluster head,  $node\_n$  is the number of nodes in that cluster.  $f_2$  represents the maximum distance from the base station to the cluster head nodes. Equation (24) is the final form of the fitness function.

$$f = \beta * (f1 + f2) + (1 - \beta) * (\sum_{j=1}^N En\_CH_j) \quad (24)$$

where  $\sum_{j=1}^N En\_CH_j$  represents the total energy of all cluster heads,  $\beta$  takes the value of 0.2.

### 5.2 Cluster Head Selection Process with PCPPE

#### Algorithm

The following describes the procedure for using the PCPPE algorithm in WSN.

Step 1: Set the location of sensor nodes and BS in the network scale, as well as the initial energy of sensor nodes.

Step 2: Form the sensor nodes into clusters using the LEACH-C algorithm and initialize the initial cluster heads of the clusters.

Step 3: The initial cluster head nodes are considered as the stick insect population in the PCPPE algorithm and the parameters of the PCPPE algorithm are assigned.

Then we start iterating with the PCPPE algorithm, and we use the proposed fitness function for the calculation of the fitness function values, and the cluster head with a small value represents the better cluster head until the end of the iteration.

Step 4: Tell each cluster the final result, then data aggregation through the cluster headers and transmit data to the BS. Repeat steps 2, 3, and 4 until all nodes are depleted of energy to end the process.

### 5.3 Simulation Environment

We compare the proposed algorithm with the LEACH algorithm, PSO-C algorithm, HFAPSO algorithm and PPE algorithm. We used two different models of WSNs with the parameters initialized in Table 6 for the simulation environment. The model parameters of the different WSNs are shown in Table 7.

**Table 6.** Parameter setting of WSN

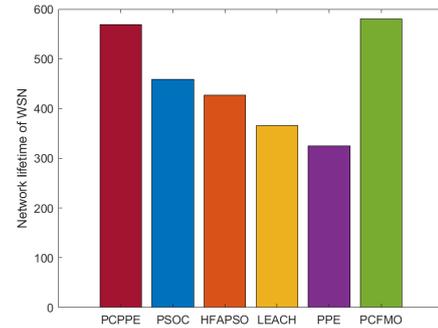
Parameter	Value
Initial energy	0.5J
Population size of PCPPE algorithm	10
Population size of other algorithms	30
Function calls	1200
$n$	6000
$E_{elect}$	$5 \times 10^{-8}$
$E_{re}$	$2 \times 10^{-8}$
$E_{agr}$	$5 \times 10^{-9}$
$E_{amp}$	$1 \times 10^{-10}$

**Table 7.** Parameters of the models of different WSNs

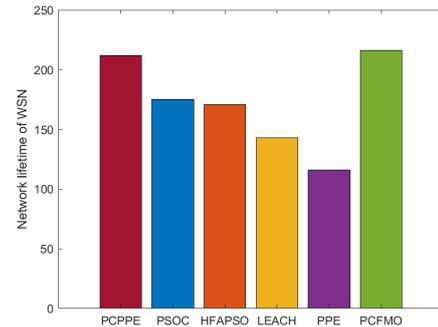
Parameter	Value
<b>WSN1</b>	
Network size	$100 \times 100 m^2$
Location of BS	(50,200)
Number of all nodes	150
Cluster head ratio	5%
<b>WSN2</b>	
Network size	$200 \times 200 m^2$
Location of BS	(100,400)
Number of all nodes	170
Cluster head ratio	5%

### 5.4 Network Life Cycle of WSN

The network life cycle in this paper is the number of rounds in which all the nodes in the network die. Figure 3 indicates the results of the life cycle of the HFAPSO algorithm, PSO-C algorithm, LEACH algorithm, PPE algorithm, PCFMO algorithm and PCPPE algorithm in the case of WSN1, while Figure 4 represents it in the case of WSN2. As shown in the figure, the network life cycle of the PCPPE algorithm is significantly higher than the other four compared algorithms, but slightly lower than the PCFMO algorithm.

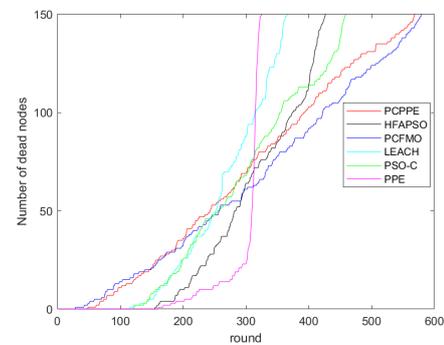


**Figure 3.** Network lifecycle in WSN1



**Figure 4.** Network lifecycle in WSN2

Figure 5, Figure 6 represent the death of nodes under WSN1 and WSN2, respectively. Although the first node dies early for the PCPPE algorithm, the WSN gradually stabilizes later and the last node dies significantly later compared to the other algorithms.



**Figure 5.** Death of a node under WSN1

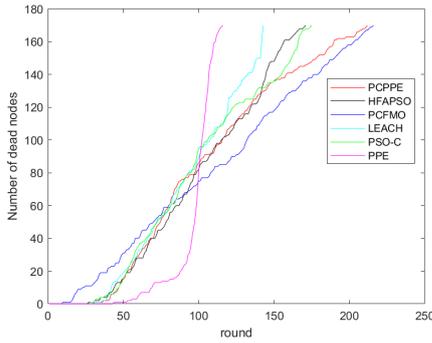


Figure 6. Death of a node under WSN2

### 5.5 WSN’s Residual Energy

The remaining energy is the overall remaining energy of all sensor nodes when the WSN is running. The more energy left, the better. Figure 7 represents the contrast of remaining energy of the PCPPE and HFAPSO, PSO-C, LEACH, PCFMO, PPE algorithms in the network area  $100 \times 100 m^2$ . In Figure 7, at the 300th round of the run, the HFAPSO algorithm has only  $8.84J$  remaining, but the PCPPE algorithm has  $17.45J$ . In Figure 8, at the 100th round of the run, the PCPPE algorithm has only  $18.92J$  remaining, but the PCFMO algorithm has  $28.08$  remaining. The residual energy of the PCPPE algorithm is higher than some algorithms because fitness function takes into account both distances, making communication less energy-consuming.

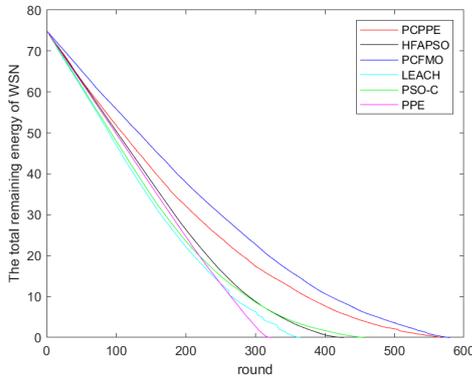


Figure 7. Remaining energy of the network under WSN1

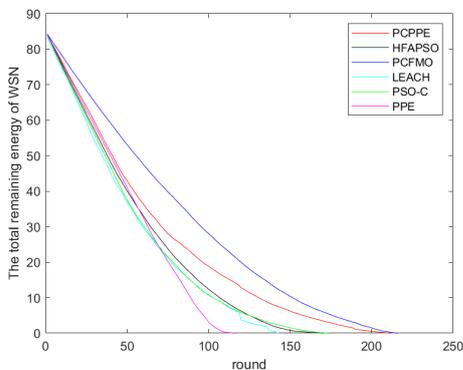


Figure 8. Remaining energy of the network under WSN2

### 5.6 The Information Received by BS

In the case of WSN1, the PCPPE, PSO-C, HFAPSO,

LEACH, PCFMO and PPE algorithms received 2658 bits, 2549 bits, 2631 bits, 2078 bits, 2818 bits, and 2429 bits, respectively. In the case of WSN2, the PCPPE, PSO-C, HFAPSO, LEACH, PCFMO and PPE algorithms received 1028 bits, 957 bits, 976 bits, 850 bits, 1068 bits, and 896 bits, respectively, and the PCPPE algorithm was slightly better than the other algorithms, as shown in Figure 9, Figure 10.

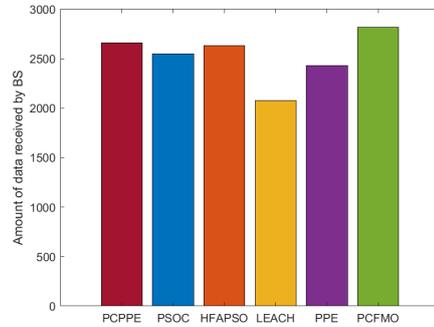


Figure 9. The amount of data received by BS under WSN1

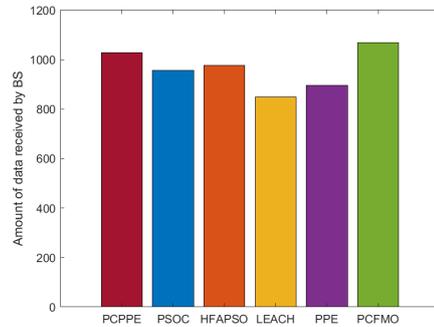


Figure 10. The amount of data received by BS under WSN2

## 6 Conclusion

In this paper, we present a Parallel Compact Phasmatodea Population Evolution (PCPPE) algorithm based on LEACH-C, a clustering mechanism that increases the survival time of the network, after an in-depth understanding of WSN. We also discuss the average distance within the cluster, residual energy and the distance of the cluster head from the base station, which enables the PCPPE algorithm to obtain better cluster heads in WSN. We calculated the PCPPE algorithm on the CEC2013 benchmark function. The experimental results show that the PCPPE algorithm has some advantages. It achieves good results even with small populations, and reduces memory usage as well as improves convergence. We then contrast the PCPPE algorithm with the HFAPSO, PSO-C, LEACH, PCFMO and PPE algorithms in four performance evaluations, and the experimental data showed that the PCPPE algorithm does reduce the energy loss and improve the network life cycle compared with other algorithms.

In the future, since the cluster radius has also been having a big influence on the performance of WSN, we can continue to optimize the algorithm and apply it to the WSN coverage problem. This paper does not consider

heterogeneous WSN, and in the future we can apply it to the localization of heterogeneous WSN. We can also consider the transmission problem over WSN, and further optimize the problem with an improved PPE algorithm.

## References

- [1] G. Y. Liu, X. K. Su, F. H. Hong, X. J. Zhong, Z. W. Liang, X. L. Wu, Z. Y. Huang, A Novel Epidemic Model Base on Pulse Charging in Wireless Rechargeable Sensor Networks, *Entropy*, Vol. 24, No. 2, Article No. 302, February, 2022. <https://doi.org/10.3390/e24020302>
- [2] J. N. Chen, Y. P. Zhou, Z. J. Huang, T. Y. Wu, F. M. Zou, R. Tso, An Efficient Aggregate Signature Scheme for Healthcare Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 6, No. 1, pp. 1-15, February, 2021.
- [3] J. Zhang, H. Nian, X. Ye, X. Ji, Y. He, A Spatial Correlation Based Partial Coverage Scheduling Scheme in Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 5, No. 2, pp. 34-43, May, 2020
- [4] C. H. Hsieh, J. Lin, C.-M. Yu, M.-H. Hung, F. Huang, A TSP-over-LEACH Protocol for Energy-Efficient Wireless Sensor Networks, *Journal of Network Intelligence*, Vol. 6, No. 4, pp. 835-846, November, 2021.
- [5] T. N. Tu, A Fuzzy Approach of Large Size Remote Sensing Image Clustering, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 11, No. 4, pp. 187-198, December, 2020.
- [6] S. Binitha, S. S. Sathya, A Survey of Bio inspired Optimization Algorithms, *International Journal of Soft Computing and Engineering*, Vol. 2, No. 2, pp. 137-151, May, 2012.
- [7] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization: An overview, *Swarm Intelligence*, Vol. 1, No. 1, pp. 33-57, June, 2007. <https://doi.org/10.1007/s11721-007-0002-0>
- [8] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, *Soft Computing*, Vol. 22, No. 2, pp. 387-408, January, 2018. <https://doi.org/10.1007/s00500-016-2474-6>
- [9] W. Li, B. Sun, Y. Huang, S. Mahmoodi, Adaptive Particle Swarm Optimization Using Scale-Free Network Topology, *Journal of Network Intelligence*, Vol. 6, No. 3, pp. 500-517, August, 2021.
- [10] R. S. Parpinelli, H. S. Lopes, A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, pp. 321-332, August, 2002. <https://doi.org/10.1109/TEVC.2002.802452>
- [11] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006. <https://doi.org/10.1109/MCI.2006.329691>
- [12] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [13] X.-S. Yang, X. He, Firefly Algorithm: Recent Advances and Applications, *International Journal of Swarm Intelligence*, Vol. 1, No. 1, pp. 36-50, August, 2013. <https://doi.org/10.1504/IJSI.2013.055801>
- [14] Y. Liu, W. M. Zheng, S. Liu, Q. W. Chai, Gaussian-Based Adaptive Fish Migration Optimization Applied to Optimization Localization Error of Mobile Sensor Networks, *Entropy*, Vol. 24, No. 8, Article No. 1109, August, 2022. <https://doi.org/10.3390/e24081109>
- [15] J. S. Pan, L. G. Zhang, R. B. Wang, V. Snášel, S. C. Chu, Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems, *Mathematics and Computers in Simulation*, Vol. 202, pp. 343-373, December, 2022.
- [16] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, Vol. 97, pp. 849-872, August, 2019. <https://doi.org/10.1016/j.future.2019.02.028>
- [17] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, Vol. 114, pp. 163-191, December, 2017. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- [18] S. Yin, J. Liu, L. Teng, A New Krill Herd Algorithm Based on SVM Method for Road Feature Extraction, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 4, pp. 997-1005, July, 2018.
- [19] S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies, *Journal of Statistical Physics*, Vol. 34, No. 5-6, pp. 975-986, March, 1984. <https://doi.org/10.1007/BF01009452>
- [20] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: Thermal exchange optimization, *Advances in Engineering Software*, Vol. 110, pp. 69-84, August, 2017. <https://doi.org/10.1016/j.advengsoft.2017.03.014>
- [21] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Information Sciences*, Vol. 222, pp. 175-184, February, 2013. <https://doi.org/10.1016/j.ins.2012.08.023>
- [22] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, *Acta Mechanica*, Vol. 213, No. 3-4, pp. 267-289, September, 2010. <https://doi.org/10.1007/s00707-009-0270-4>
- [23] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, *Knowledge-Based Systems*, Vol. 191, Article No. 105190, March, 2020. <https://doi.org/10.1016/j.knsys.2019.105190>
- [24] D. Whitley, A genetic algorithm tutorial, *Statistics and Computing*, Vol. 4, No. 2, pp. 65-85, June, 1994. <https://doi.org/10.1007/BF00175354>
- [25] J. A. Vasconcelos, J. A. Ramirez, R. H. C. Takahashi, R. R. Saldanha, Improvements in genetic algorithms, *IEEE Transactions on magnetics*, Vol. 37, No. 5, pp. 3414-3417, September, 2001. <https://doi.org/10.1109/20.952626>
- [26] M. W. Iruthayarajan, S. Baskar, Covariance matrix adaptation evolution strategy based design of centralized PID controller, *Expert Systems with Applications*, Vol. 37, No. 8, pp. 5775-5781, August, 2010. <https://doi.org/10.1016/j.eswa.2010.02.031>
- [27] R. Moghdani, K. Salimifard, E. Demir, A. Benyettou, Multi-Objective Volleyball Premier League algorithm, *Knowledge-Based Systems*, Vol. 196, Article No. 105781, May, 2020. <https://doi.org/10.1016/j.knsys.2020.105781>
- [28] R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-

- based optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Design*, Vol. 43, No. 3, pp. 303-315, March, 2011. <https://doi.org/10.1016/j.cad.2010.12.015>
- [29] A. H. Kashan, League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships, *Applied Soft Computing*, Vol. 16, pp. 171-200, March, 2014. <https://doi.org/10.1016/j.asoc.2013.12.005>
- [30] H. Bingol, B. Alatas, Chaotic League Championship Algorithms, *Arabian Journal for Science and Engineering*, Vol. 41, No. 12, pp. 5123-5147, December, 2016. <https://doi.org/10.1007/s13369-016-2200-9>
- [31] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, B. Yan, Digital watermarking with improved SMS applied for QR code, *Engineering Applications of Artificial Intelligence*, Vol. 97, Article No. 104049, January, 2021. <https://doi.org/10.1016/j.engappai.2020.104049>
- [32] T. T. Nguyen, T. Dong-Nguyen, V. T. Nguyen, An Optimizing Pulse Coupled Neural Network based on Golden Eagle Optimizer for Automatic Image Segmentation, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 13, No. 3, September, 2022.
- [33] J. Wu, M. Xu, F. F. Liu, M. Huang, L. H. Ma, Z. M. Lu, Solar Wireless Sensor Network Routing Algorithm Based on Multi-Objective Particle Swarm Optimization, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 12, No. 1, pp. 1-11, March, 2021.
- [34] P. C. Song, S. C. Chu, J. S. Pan, H. Yang, Simplified Phasmatodea population evolution algorithm for optimization, *Complex & Intelligent Systems*, Vol. 8, No. 4, pp. 2749-2767, August, 2022. <https://doi.org/10.1007/s40747-021-00402-0>
- [35] J. S. Pan, P. C. Song, C. A. Pan, A. Abraham, The Phasmatodea Population Evolution Algorithm and Its Application in 5G Heterogeneous Network Downlink Power Allocation Problem, *Journal of Internet Technology*, Vol. 22, No. 6, pp. 1199-1213, November, 2021. <https://doi.org/10.53106/160792642021112206001>
- [36] A. N. Zhang, S. C. Chu, P. C. Song, H. Wang, J. S. Pan, Task Scheduling in Cloud Computing Environment Using Advanced Phasmatodea Population Evolution Algorithms, *Electronics*, Vol. 11, No. 9, Article No. 1451, May, 2022. <https://doi.org/10.3390/electronics11091451>
- [37] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp. 660-670, October, 2002. <https://doi.org/10.1109/TWC.2002.804190>
- [38] S. K. Singh, P. Kumar, J. P. Singh, A Survey on Successors of LEACH Protocol, *IEEE Access*, Vol. 5, pp. 4298-4328, February, 2017. <https://doi.org/10.1109/ACCESS.2017.2666082>
- [39] G. Ran, H. Zhang, S. Gong, Improving on LEACH Protocol of Wireless Sensor Networks Using Fuzzy Logic, *Journal of Information & Computational Science*, Vol. 7, No. 3, pp. 767-775, March, 2010.
- [40] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, A. H. Gandomi, Residual Energy-Based Cluster-Head Selection in WSNs for IoT Application, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5132-5139, June, 2019. <https://doi.org/10.1109/JIOT.2019.2897119>
- [41] B. Pitchaimanickam, G. Murugaboopathi, A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks, *Neural Computing and Applications*, Vol. 32, No. 12, pp. 7709-7723, June, 2020. <https://doi.org/10.1007/s00521-019-04441-0>
- [42] N. M. A. Latiff, C. C. Tsimenidis, B. S. Sharif, Energy-Aware Clustering for Wireless Sensor Networks using Particle Swarm Optimization, *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Athens, Greece, 2007, pp. 1-5. <https://doi.org/10.1109/PIMRC.2007.4394521>
- [43] S. C. Chu, X. W. Xu, S. Y. Yang, J. S. Pan, Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks, *Knowledge-Based Systems*, Vol. 241, Article No. 108124, April, 2022. <https://doi.org/10.1016/j.knosys.2022.108124>
- [44] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, J. C. Bansal, A Survey on Parallel Particle Swarm Optimization Algorithms, *Arabian Journal for Science and Engineering*, Vol. 44, No. 4, pp. 2899-2923, April, 2019. <https://doi.org/10.1007/s13369-018-03713-6>
- [45] M. Pedemonte, S. Nesmachnow, H. Cancela, A survey on parallel ant colony optimization, *Applied Soft Computing*, Vol. 11, No. 8, pp. 5181-5197, December, 2011. <https://doi.org/10.1016/j.asoc.2011.05.042>
- [46] G. R. Harik, F. G. Lobo, D. E. Goldberg, The Compact Genetic Algorithm, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 287-297, November, 1999. <https://doi.org/10.1109/4235.797971>
- [47] X. Cheng, Y. Jiang, D. Li, Z. Zhu, N. Wu, Optimal Operation with Parallel Compact Bee Colony Algorithm for Cascade Hydropower Plants, *Journal of Network Intelligence*, Vol. 6, No. 3, pp. 440-452, August, 2021.

## Biographies



**Jeng-Shyang Pan** received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the

University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.



**Meng-Fei Zhang** received her B.S. degree from Qilu Normal University, China, in 2021. She is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests include swarm intelligence and wireless sensor networks.



**Shu-Chuan Chu** received a Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She has been a Research

Fellow and Associate Professor in the College of Science and Engineering of Flinders University, Australia since December 2009. Currently, she is a Research Fellow with a Ph.D. advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. She also serves as an editorial board member for *Engineering Applications of Artificial Intelligence (EAAI)*, *Journal of Internet Technology (JIT)* and *Research Reports on Computer Science (RRCS)*. Her research interests are mainly in swarm intelligence, intelligent computing, and wireless sensor networks.



**Xing-Wei Xu** received his B.S. degree from Shandong University of Science and Technology, China, in 2020. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include intelligent computing and wireless

sensor networks.



**Geng-Geng Liu** received the Ph.D. degree in Applied Mathematics from Fuzhou University in 2015. He is currently a professor at Fuzhou University. His current main research interests are EDA algorithms, computational intelligence and its applications.