



The Pine Cone Optimization Algorithm (PCOA)

Mahdi Valikhan Anaraki and Saeed Farzin *

Department of Water Engineering and Hydraulics Structures, Faculty of Civil Engineering, Semnan University, Semnan 35131-19111, Iran; mvalikhan@semnan.ac.ir

* Correspondence: saeed.farzin@semnan.ac.ir

Abstract: The present study introduces a novel nature-inspired optimizer called the Pine Cone Optimization algorithm (PCOA) for solving science and engineering problems. PCOA is designed based on the different mechanisms of pine tree reproduction, including pollination and pine cone dispersal by gravity and animals. It employs new and powerful operators to simulate the mentioned mechanisms. The performance of PCOA is analyzed using classic benchmark functions, CEC017 and CEC2019 as mathematical problems and CEC2006 and CEC2011 as engineering design problems. In terms of accuracy, the results show the superiority of PCOA to well-known algorithms (PSO, DE, and WOA) and new algorithms (AVOA, RW_GWO, HHO, and GBO). The results of PCOA are competitive with state-of-the-art algorithms (LSHADE and EBOwithCMAR). In terms of convergence speed and time complexity, the results of PCOA are reasonable. According to the Friedman test, PCOA's rank is 1.68 and 9.42 percent better than EBOwithCMAR (second-best algorithm) and LSHADE (third-best algorithm), respectively. The authors recommend PCOA for science, engineering, and industrial societies for solving complex optimization problems.

Keywords: optimization; nature-inspired; pine tree; pine cone; mathematical benchmark functions; engineering problems; swarm intelligence



Citation: Valikhan Anaraki, M.; Farzin, S. The Pine Cone Optimization Algorithm (PCOA). *Biomimetics* **2024**, *9*, 91. <https://doi.org/10.3390/biomimetics9020091>

Academic Editor: Heming Jia

Received: 21 December 2023

Revised: 26 January 2024

Accepted: 29 January 2024

Published: 1 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nature-inspired optimization algorithms refer to the new optimization methods designed based on natural phenomena. These algorithms can solve optimization problems without the need to calculate the gradient of the objective function. Hence, nature-inspired optimization algorithms can solve non-derivative optimization problems. Nature-inspired optimization algorithms are flexible and can handle many constraints and decision variables. In contrast, the computation times of numeric optimization methods increase exponentially with the number of constraints and decision variables. Thanks to the design and development of nature-inspired algorithms, it is possible to solve complex problems that have not been solved so far. Although many nature-based algorithms have been proposed, new algorithms are still being developed due to the “no free lunch in search and optimization” theory [1]. According to this theory, each optimization algorithm can solve some optimization problems. Some problems still need to be solved, and new nature-inspired optimization algorithms can solve them with better accuracy. Researchers divide these algorithms into six categories based on the type of idea that inspires them: evolutionary, swarm-based, mathematical-based, physics and chemistry-based, sport-based, and human-based. Evolutionary algorithms are based on the principle of natural evolution. They begin by creating a random population, which then evolves through selection, crossover, and mutation operations. The most popular evolutionary algorithms are Genetic Algorithm (GA) (developed in 1975) [2]; Differential Evolution (DE) (developed in 1997) [3]; Harmony Search (HS) (developed in 2001) [4]; Black Widow Optimization Algorithm (BWO) (developed in 2020) [5]; Learner Performance Based Behavior algorithm (LPB) (developed in 2021) [6]; Evolutionary Mating Algorithm, based on the adoption of the random mating concept from the Hardy–Weinberg equilibrium and crossover index [7]; and One-Dimensional

Subspaces Optimization Algorithm (1D-SOA) [8]. Although evolutionary algorithms have seen successful application in some problems, these algorithms slowly converge to the optimal solutions and become stuck in local optima. Swarm-inspired algorithms are another type of nature-inspired algorithms. These algorithms are designed based on the moving methods of swarms. The most popular swarm-based algorithm is the Particle Swarm Optimization algorithm (PSO) (developed in 1995) [9]. Moving swarms of birds or fish inspire the search processing in PSO. Researchers introduced the Artificial Bee Colony (ABC) in 2007 [10], Firefly Algorithm (FA) [11] in 2008, Bat Optimization Algorithm (BA) in 2012 [12], Krill-Herd Algorithm (KHA) in 2012 [13], Gray Wolf Optimizer (GWO) in 2014 [14], Whale Optimization Algorithm (WOA) in 2016 [15], Harris Hawks Optimization Algorithm [16] in 2019, Coati Optimization Algorithm (COA) in 2023 [17], Dung Beetle Optimizer (DBO) in 2023 [18], Kookaburra Optimization Algorithm (KOA) in 2023 [19], Giant Armadillo Optimization (GAO) in 2023 [20], Lyrebird Optimization Algorithm (LOA) in 2023 [21], and Humboldt Squid Optimization Algorithm (HSOA) in 2023 [22]. In this category, algorithms such as PSO have poor local optimization ability. The bat algorithm has yet to establish a good balance between global and local search ability. KH has two main limitations: stagnation in local optima and a slow convergence speed. GWO and WOA use adaptive weights to control the search scope and balance global and local search capabilities. However, the global search ability of these algorithms is weak. The other types of nature-inspired optimization algorithms are physics and chemistry-based algorithms. These algorithms are designed based on the laws of physics or chemistry. Some of the well-known algorithms in this category include Central Force Optimization (CFO) (developed in 2008) [23], Charged System Search (CSS) Optimization Algorithm (developed in 2010) [24], Artificial Chemical Reaction Optimization Algorithm (ACROA) (developed in 2011) [25], Multi-Verse Optimizer (MVO) (developed in 2016) [26], Flow Direction Algorithm (FDA) (developed in 2021) [27], and Rime Optimization Algorithm (RIME) (developed in 2023) [28]. Certain optimization problems may pose challenges for physics and chemistry-based algorithms. In particular, ACROA may suffer from slow convergence as a result of unbalanced local and global searches, while MVO may lack diversity, resulting in trapping and premature convergence. Human-based optimization algorithms are another category of nature-inspired optimization algorithms. These algorithms work based on human-based phenomena. In 1989, Tabu Search (TS) was developed, using the mechanics of human memory [29]. The other well-known algorithms in this category consist of the Poor and Rich Optimization (PRO) algorithm (developed in 2019) [30], Coronavirus Herd Immunity Optimizer (CHIO) (developed in 2019) [31], Incomprehensible but Intelligible-in-Time Logics: Theory and Optimization Algorithm (ILA) (developed in 2023) [32], Technical and Vocational Education and Training-Based Optimizer (TVETBO) (developed in 2023) [33], and Human Memory Optimization (HMO) (developed in 2024) [34]. The category of sport-based algorithms includes optimization algorithms that are motivated by sports methods. The best known of these algorithms are Tug of War Optimization (TWO) (developed in 2017) [35], Football Game Based Optimization (FGBO) (developed in 2020) [36], and Athletic Run Based Optimization (ARBO) (developed in 2021) [37]. Efforts to develop and create new optimization algorithms are still ongoing. In this regard, researchers developed the Polar Bear Optimization Algorithm [38]; a modified Whale Optimization Algorithm, based on different searching paths and perceptual disturbance [39]; an Improved Butterfly Optimization Algorithm, based on the cross-entropy method [40]; and an improved Chaotic Particle Swarm Optimization Algorithm, with a more symmetric distribution [41]. Figure 1 shows nature-inspired optimization algorithms in the different categories.

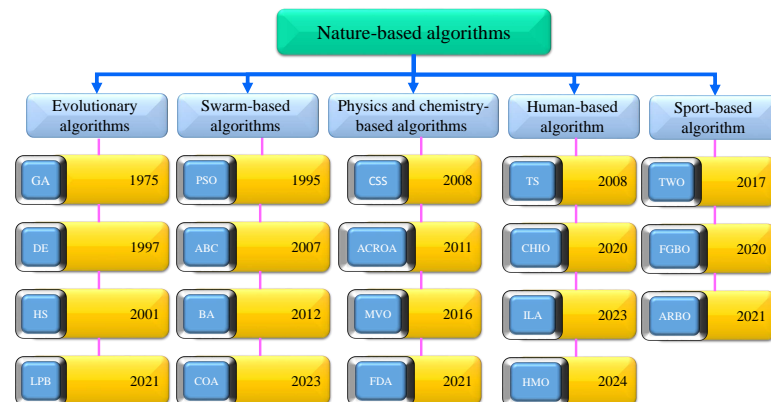


Figure 1. Nature-based optimization algorithms in various categories.

Although researchers have studied various natural phenomena to design nature-based algorithms, phenomena related to the life of plants, despite their high potential for designing nature-based algorithms, have received less attention. The studies in this field are reviewed in the following section. Researchers developed the Invasive Weed Optimization (IWO) algorithm in 2006, based on colonizing weeds [42]. IWO selects better fitness seeds and distributes them in a search space. It uses normal-distribution random numbers with a large standard deviation in the exploration phase and decreases the standard deviation in the exploitation phase. IWO uses adaptive standard deviation but does not share information between seeds. The Flower Pollination Algorithm was introduced in 2012. The pollination mechanism of flowers inspires this algorithm [43]. FPA uses Levy flight to balance exploration and exploitation. It updates the position of pollen based on the best position, the current position, and Levy distribution. When exploiting, it updates the position by randomly choosing two pollen grains and generating a random number. It does not use dispersing seeds and assumes an equal chance for exploration and exploitation. In 2015, the Tree Seed Algorithm (TSA) was motivated by the relationship between trees and their seeds for continuous optimization [44]. TSA uses the best tree and seed positions as selection operators. TSA generates multiple seeds around the current and a random tree during exploration. During exploitation, TSA generates numerous seeds around the tree based on the best position of all trees, a random tree, and the current position of the tree. TSA uses the wind impact to disperse seeds but does not involve animals, pollination, or Levy flight. Researchers in 2018 developed the Tree Growth Algorithm (TGA) based on the competition among trees for acquiring light and food [45]. TGA categorizes trees into four fitness groups and uses different methods for exploration and exploitation. It balances exploration and exploitation but does not use dispersing seed strategies, a pollination process, or Levy flight. In 2022, the Trees Social Relations Optimization Algorithm was inspired by the hierarchical and collective life of trees in the jungle [46]. This algorithm has advantages, such as dividing the search domain into sub-clusters. However, it does not incorporate pollination mechanisms or the roles of animals in seed dispersal. In 2023, researchers proposed the Orchard Algorithm (OA), based on the fruit gardening process [47]. This algorithm used a neighborhood search. However, it does not use clustering search space, pollination mechanisms, or dispersing seeds. The Tree Optimization Algorithm (TOA) was created by [48] in 2023. The growth of trees inspired this algorithm. However, TOA does not consider reproduction methods for one tree.

To the best of the authors' knowledge, previous research on nature-inspired optimization algorithms has given less attention to the excellent life potential of plants, particularly their reproductive methods. While there have been some studies on developing nature-based algorithms, little work has been performed on exploring the different aspects of tree reproduction. As a result, this study introduces a new algorithm named the Pine Cone Optimization Algorithm (PCOA), which is based on the unique reproductive strategies

of pine trees. These strategies have enabled the pine species to survive for millions of years, and simulating them in developing PCOA could offer solutions to scientific and engineering issues. The new contributions of this study are as follows:

- Considering different pollination mechanisms for simulating the impact of wind on pine tree pollination.
- Considering the impacts of animals and gravity on pine cone dispersal using new operators.
- Clustering the search domain to sub-cluster and search in each of them in parallel.
- Shrinking the search domain over course generations.
- Using both swarm-based and evolutionary operators in updating solutions to the PCOA.

PCOA is set apart from other existing algorithms due to these contributions. The remainder of the paper is structured as follows: Section 2 describes the pine tree's reproduction methods, introduces the PCOA theory and its structure and flowchart, and discusses the mathematical and engineering problems. Section 3 presents the accuracy, convergence efficacy, and time complexity of PCOA in classic and new mathematical benchmark functions and engineering problems. Lastly, Section 4 presents the conclusions and future scope.

2. Materials and Methods

2.1. Inspiration: The Life Cycle of the Pine Tree

Pines are coniferous trees in the *Pinus* family of the Pinaceae, which grow in a wide range of ecosystems. They grow to 50 m in height with dark green needles and 5–10 cm cones with rounded scales [49]. Some pines, like Scotch pine, are monoecious because they have female and male flowers on the same tree. Tree shoots have male flowers at their base. The wind pollinates female flowers in the early summer. The female flower develops into small cones that quickly grow and reach full size by early summer after more than a year. Then, gravity causes the cones to break away and fall to the ground, or animals eat them and disperse the seeds in spring. Eventually, the seeds turn into saplings, and new trees arise. Plants like the pine tree are pollinated only by the wind and, unlike other plants, insects do not affect their pollination [50,51]. Figure 2 shows the life cycle of pine trees and pine tree swarms in the forest.



Figure 2. Pine tree life cycle and swarms in the forest. (a) Pine tree life cycle. (b) Pine tree forest.

2.2. Mathematical Modeling of Pine Reproduction

The pollination and dispersal of cone seeds is a critical phase of the life cycle of pine trees. Therefore, the present study focuses only on considering the pollination and dispersion of cones. In PCOA, gravity and animals scatter the cones, just like in nature. Continuing with the study, we mathematically express the PCOA, considering the pollination and scattering of cones through gravity and animals. According to previous studies, pine tree reproduction is significantly dependent on pollination [52], and dispersion of

cone seeds by gravity [53] and animals [54]. Pollination in a pine tree is similar to solving an optimization problem in two aspects. In one aspect, the movement of pollen particles between cones of pine trees can be simulated by a swarm-based operator. In this case, pollen particles move driven by the wind. The atmosphere of the forest is the search space for problems, and the pollen particles are the search agents. Each particle has a position and objective function in the search space of the optimization problem. The movement of pollen particles from one pine tree cone to another is similar to searching for the optimal solution in the problem's search space. In another aspect, pollen particles enter the female flower of the pine tree and cause the female flower to develop and create small cones. Therefore, this process can be simulated by evolutionary operators. The evolution of the pine cone from one state to another is similar to improving the problem solution and finding its optimal solution. The scattering of pine cone seeds by gravity and animals causes the creation and growth of pine seedlings in new situations. If the position of the seedling is suitable, it will become a pine tree. Similarly, in the PCOA, the answers are updated if a better position is found. The pine tree forest is equivalent to the problem search space, and the pine cone seeds are the search agents. Population-based operators can be used to simulate the dispersion of pine cone seeds by gravity or animals. Hence, population-based and evolutionary-based operators can simulate the mentioned biological principles and solve optimization problems. In this regard, the position of the pollen, seeds of cones, and pine trees are the search agents, and the best position for growing the pine tree is the global optimal problem.

2.3. Generating the Initial Population in PCOA

The PCOA has two populations: pine trees and their cones. The problem domain is divided into equal segments. In the algorithm initialization, there is a tree in the center of each segment, and multiple cones are generated around each tree. For this purpose, the boundary conditions for each part are given by

$$LbS = lb + (i - 1) \times \frac{ub - lb}{N_{tree}} \quad (1)$$

$$UbS = ub + (i - 1) \times \frac{ub - lb}{N_{tree}} \quad (2)$$

In Equations (1) and (2), lb and ub are the lower and upper bounds of the problem's domain, and LbS and UbS are each segment's lower and upper bounds. The pine cone population is generated as follows:

$$CX_{j,i} = LbS_i + \overrightarrow{rand}_{1 \times dim} \times (rand \times UbS_i - rand \times LbS_i) \quad (3)$$

In Equation (3), CX is the pine cone position, $\overrightarrow{rand}_{1 \times dim}$ is a random vector with a normal distribution between 0 and 1, and $rand$ is a random number with a normal distribution between 0 and 1. In operators of PCOA, i indicates the pine tree index and j shows the cone index. After initializing PCOA, the position of each tree is equal to the position of the best cone in that segment.

2.4. Pine Cone Dispersal by Gravity (Exploitation)

When the pine cones mature and their weight increases, gravity causes the pine cones to fall from the tree. The issue causes the cones to disperse in a limited area. The search space problem is divided into multiple super-cubes to simulate this process. Each of these super-cubes contains a tree inside. In order to accurately simulate the impact of gravitational forces on the dispersion of cones in the vicinity of trees, it is necessary to generate multiple solutions around each tree. In order to accomplish this objective, the following operator is utilized.

$$CX_{j,i}^{new} = \begin{cases} TX_i + w_1 \times R_1 \times (R_2 \times (Ub_s^i - Lb_s^i - TX_i)), & \text{if } Control_{parameter} = 0 \\ CX_{j,i} + w_1 \times R_1 \times (R_2 \times (Ub_s^i - Lb_s^i - Tpop_{all,r1}) - Tbest_{x,i}), & \text{otherwise.} \end{cases} \quad (4)$$

In Equation (4), $CX_{j,i}^{new}$ is the new position of the pine cone, $CX_{j,i}$ is the updated position of the pine cone, TX_i is the position of the tree, Ub_s^i is the upper bound of the super-cube, Lb_s^i is the lower bound of the super-cube, $Tpop_{all,r1}$ is a randomly selected solution in the memory of PCOA, $Tbest_{x,i}$ is the i th of the top solutions of PCOA, w_1 is an adaptive weight, $r1$ is a random integer number between 1 and the size of the PCOA memory, and R_1 and R_2 are normal random numbers between 0 and 1. In the operators of PCOA, i shows the pine tree index, and j indicates the cone index. The boundaries of the problem (lb and ub), LbS and UbS , shrink over the generations. In this approach, the research space is restricted to the closing area in the best surrounding position over the generations. This approach enables the PCOA to balance exploration and exploitation. To achieve this, we define the following equations:

$$LbS = Lb + (i - 1) \times \frac{Ub - Lb}{N_{tree}} \quad (5)$$

$$UbS = Ub + (i - 1) \times \frac{Ub - Lb}{N_{tree}} \quad (6)$$

$$Lb = lb + Radius_{lb} \times W \quad (7)$$

$$Ub = ub - Radius_{ub} \times W \quad (8)$$

$$W = \min FES / FES_{max}, 0.5 \quad (9)$$

$$Radius_{lb} = X_{best} - lb \quad (10)$$

$$Radius_{ub} = ub - X_{best} \quad (11)$$

In Equations (5)–(11), the algorithm updates the boundaries of the problem, Lb and Ub . It also determines the shrinking radius with $Radius_{lb}$ and $Radius_{ub}$. The adaptive weight W decreases from 1 to 0.5. X_{best} represents the current best position. FES signifies the current generation, with FES_{max} denoting the maximum number of generations.

2.5. Pollination (Exploration)

In pollinating pine trees, the agents are produced from male cones and drift to female cones. This movement can occur from one tree’s cones to those of another or within the same tree. As this movement is very complicated, it is possible to define different mechanisms to simulate it. In PCOA, two different methods are defined to mimic this motion. In the first method, Algorithm 1 is used to simulate wind pollination. In the second method, wind pollination is simulated according to the study by [55]. This method is formulated as follows:

$$CX_{j,i}^{new} = CX_{j,i} + 0.5 \times \phi_{r_1} \times (Tbest_j - CX_{r_1,i}) + 0.5 \times \phi_{r_3} \times (Tbest_j - CX_{r_3,i}) \quad (12)$$

In Equation (12), ϕ is the chance of successful pollination of pine cones. ϕ is calculated as follows:

$$\phi_i = 1 - e^{-\gamma \sum_{j=1}^n a_{i,j}} \quad (13)$$

$$a_{i,j} = \frac{\frac{\beta}{d_{i,j}^\alpha} - \frac{\alpha}{d_{i,j}^\beta}}{\beta - \alpha} \quad (14)$$

In Equations (13) and (14), $a_{i,j}$ is the effect of the i th cone on the j th cone, and $d_{i,j}$ is the Euclidian distance between the i th cone and j th cone. In addition, α , β , and γ are constant

parameters that define the pollination process mathematically. According to the conducted study by [55], the values of α , β , and γ are set to 3, 40, and 0.62, respectively.

Algorithm 1 Pollination

Input: Pollination cycle (P_N), cones position (CX), cones fitness (CF)

Output: Pine Cone's position

Wind's pollination :

```

1: while  $N < P_N$  do
2:    $N \leftarrow N + 1$ 
3:    $Memory_{Index,rand} \leftarrow \lfloor dim * rnd \rfloor$ 
4:    $\mu_{W_1} = Memory_{W_1}(Memory_{Index,rand})$ 
5:    $\mu_{W_2} = Memory_{W_2}(Memory_{Index,rand})$ 
6:    $\mu_{W_3} = Memory_{W_3}(Memory_{Index,rand})$ 
7:    $\mu_{cr} = Memory_{cr}(Memory_{Index,rand})$ 
8:    $W_1 \leftarrow \mu_{W_1} + \overrightarrow{Rnd_1} \times \tan(\pi \times (\overrightarrow{Rnd_2} - 0.5))$ 
9:    $W_2 \leftarrow \mu_{W_2} + \overrightarrow{Rnd_3} \times \tan(\pi \times (\overrightarrow{Rnd_4} - 0.5))$ 
10:   $W_3 \leftarrow \mu_{W_3} + 0.1 \times \tan(\pi \times (\overrightarrow{Rnd_5} - 0.5))$ 
11:   $cr \leftarrow \mu_{cr} + 0.1 \times \overrightarrow{Rnd_6}$ 
12:  if  $W_1 < 0$  then
13:     $W_1 \leftarrow \mu_{W_1} + 0.1 \times \tan(\pi \times (\overrightarrow{Rnd_7} - 0.5))$ 
14:  end if
15:  if  $W_2 < 0$  then
16:     $W_2 \leftarrow \mu_{W_2} + 0.1 \times \tan(\pi \times (\overrightarrow{Rnd_8} - 0.5))$ 
17:  end if
18:  if  $W_3 < 0$  then
19:     $W_3 \leftarrow \mu_{W_2} + 0.1 \times \tan(\pi \times (\overrightarrow{Rnd_9} - 0.5))$ 
20:  end if
21:   $W_1 \leftarrow \min(W_1, 1), W_2 \leftarrow \min(W_2, 1), W_3 \leftarrow \min(W_3, 1), cr \leftarrow \min(cr, 1), cr \leftarrow \max(cr, 0)$ 
22:  Generate three random integer number ( $r_1, r_2, r_3$ )
23:  Choose  $Tbest_{rate}\%$  of best solutions and save them on  $Tbest$ ;
24:  Choose  $Tworst_{rate}\%$  of worst solutions and save them on  $Tworst$ ;
25:  if  $rand < 0.5$  then
26:     $CX^{new} \leftarrow XC + W_1 \times (Tbest - CX_{r_1}) + W_2 \times (CX_{r_1} - Tpop_{all,r_2})$ 
27:  else
28:     $W = \max(W_1 \times W_2, (1 - W_1) \times W_2, 1 - W_2)$ 
29:     $CX^{new} \leftarrow W_2 \times (W_1 \times CX + (1 - W_1) \times Tbest) + (1 - W_2) \times CX_{r_1} + W \times (CX_{r_3} - Tpop_{all,r_2})$ 
30:  end if
31:  Evaluate fitness function
32:  if  $CF^{new} < CF \ || \ rand < cr$  then
33:     $XC \leftarrow XC^{new}, XF \leftarrow XF^{new}$ 
34:  end if
35:  Update  $Tpop_{all}$  (Add new solution to archive and remove duplicate or randomly remove some solutions to maintain the archive size);
36:  Update  $\mu_{W_1}, \mu_{W_2}, \mu_{cr}$ 
37: end while
38: return Pine Cone's position

```

2.6. Pine Cone Dispersal by Animals (Exploitation)

Animals eat pine cones and disperse them in the environment. For example, in the fall, squirrels bury pine cones to eat later on [56]. Birds and other animals like bears, deer, rats, and mice eat pine cones. In the PCOA, four operators are defined to simulate these behaviors. In the first operator, it is assumed that animals like the squirrel carry

pine cones from a random position to a position with a better condition for burial. This operator employs the quadratic programming optimization method to find the best location. The initial point in this method is defined as follows:

$$X_{initial} = X_{best} + rand \times (\overline{CX} - X_{best}) \tag{15}$$

In Equation (15), $X_{initial}$ is the initial position in quadratic programming, and \overline{CX} is an average of all the pine cones' positions. It should be noted that the quadratic programmatic search is performed under the updated boundary conditions of the problem. This approach allows the PCOA to approximate the global optimum better. The roles of animals in scattering pine cones are the basis for driving the second to fourth operators. The second operator is designed as follows:

$$X_{animal} = \frac{X_{best} + X_{animal}}{2} + levy \times (levy \times (lb + ub - \frac{X_{best} + X_{animal}}{2}) - \frac{X_{best} + X_{animal}}{2}) \tag{16}$$

In Equation (16), X_{animal} is known as the position of the pine cones carried by animals, a *Levy* random number generated by the Levy distribution. In the third operator, it is assumed that animals carry pine cones from their position to near a tree's position. The position of the pine cone is updated by the third operator as follows:

$$X_{animal} = CX + (1 - w_d) \times \overline{TX} + w_d \times levy \times (levy \times (lb + ub - \overline{TX}) - \overline{Tree_x}) \tag{17}$$

In Equation (17), \overline{TX} is the average position of the pine trees, w_d is an adaptive weight that is given by [57]:

$$w_d = exp-20 \times \frac{FES}{FES_{max}} \tag{18}$$

w_d limits the search domain over increasing function evaluations and leads to converting exploration into exploitation. The fourth operator assumes that an animal carries a pine cone in a nearer position:

$$X_{animal} = CX + w_d \times levy \times (levy \times (lb + ub - CX) - CX) \tag{19}$$

In PCOA, exploitation falls under the responsibility of all four defined operators. Algorithm 2 illustrates how animals in PCOA disperse pine cones.

Algorithm 2 Dispersing pine cone by animals

Input:

Input: P_1, P_2, CX, TX, FES and FES_{max}

Output: Pine Cone's position

- Dispersing pine cone by animals :*
- 1: **if** *and*($FES > P_2 \times FES_{max}, rand < 0.9$) || *and*($FES < P_1 \times FES_{max}, rand < 0.9$) **then**
 - 2: Calculate $X_{initial}$ using Equation (12)
 - 3: Run quadratic programming optimization method based on the $X_{initial}$
 - 4: Update X_{best} if solution of quadratic programming was better
 - 5: Calculate X_{animal} using Equation (13)
 - 6: **else**
 - 7: Compute w_d using Equation (15)
 - 8: **if** $rand < 0.5$ **then**
 - 9: Calculate $X_{animals}$ using Equation (14)
 - 10: **else**
 - 11: Calculate $X_{animals}$ using Equation (16)
 - 12: **end if**
 - 13: **end if**
 - 14: **return** Pine Cone's position
-

2.7. Assigning Pine Cones to Each Tree

Each tree in PCOA has cones that are updated exclusively near it. This approach is used to simulate the scattering of pine cones by gravity. In the first generation, every tree has an equal number of cones. However, in later generations, each tree receives the cones that are closest to it. Since most solutions converge towards the best solution through increasing generations, pines with a better position have more cones. This issue balances exploration and exploitation and estimates the absolute optimum more accurately.

2.8. Flowchart of PCOA

In the PCOA, first, the algorithm generates the initial population. Then, it updates this population using the operators of pine cone dispersal by gravity, wind pollination, and pine cone dispersal by animals. Gravity dispersal of the pine cones and animal dispersal of the pine cones handle exploitation, and wind pollination is associated with exploration. In addition, the adaptive weight and reduced boundary conditions help the PCOA to achieve a balance between exploration and exploitation. Figure 3 shows the flowchart of PCOA.

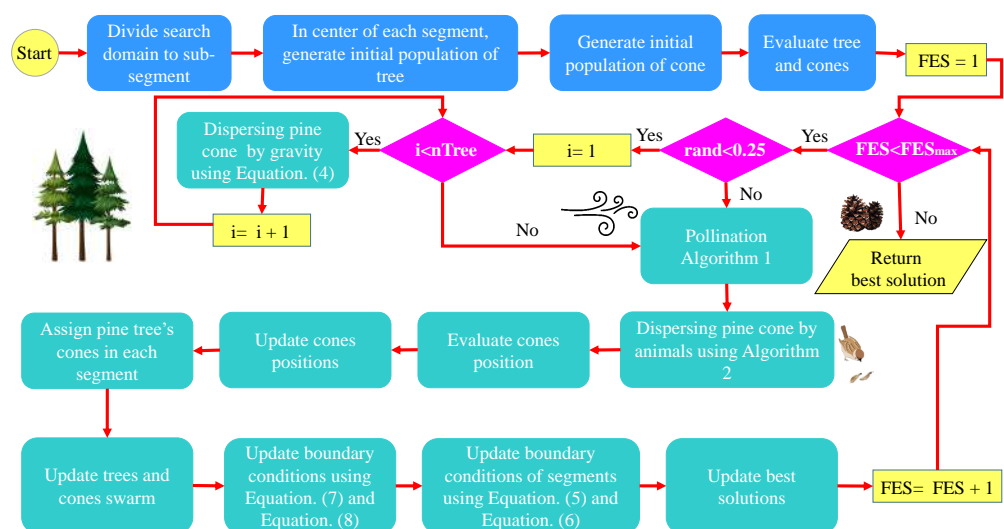


Figure 3. Pine Cone Optimization Algorithm (PCOA) flowchart.

3. Experimental Research Settings

PCOA was evaluated using mathematical benchmark functions and engineering problems. The mathematical benchmark functions comprise 23 classic benchmark functions, 29 CEC-BC-2017 benchmark functions in 10, 30, 50, and 100 dimensions [58], and 10 CEC-C06 2019 benchmark functions [58]. The engineering problems include 6 CEC2006 and 17 CEC2011 problems [59]. PCOA and other algorithms were implemented using Matlab R2020b, executed on Windows 10, Intel Xeon 2.2 GHz CPU, and 40 G RAM. This study uses the following assumptions to evaluate PCOA and other competitors:

- In order to achieve a fair comparison of optimization algorithms, we set the maximum number of function evaluations (FES_{max}) to the same value. FES_{max} for the classic benchmark functions, CEC2017, CEC2006, and CEC2011 is equal to 150,000, 10,000, 15,000, and 150,000 [59], respectively. For CEC2019, the FES_{max} for first to tenth benchmark functions are equal to 90,000, 320,000, 1,800,000, 1,000,000, 100,000, 100,000, 1,000,000, 1,000,000, 1,000,000, and 600,000, respectively.
- The researchers set the independent run of algorithms for the classic benchmark functions, CEC2017, CEC2019, and CEC2006, to 51. For CEC2011, this value was 25. These values are based on [22,59,60]. In the following subsections, more information about the experimental settings is presented.

3.1. Mathematical Benchmark Functions

Sixty-two benchmark functions were employed to evaluate the performance of PCOA and the other investigated algorithms. The classic benchmark functions comprise seven unimodal (1 to 7), six multi-modal (8 to 13), and ten multi-modal fixed-dimension (14 to 23) functions. The CEC2017 test suite includes two unimodal (1 and 3), seven multi-modal (4–10), ten hybrid (11–20), and ten composite (21–30) benchmark functions. Because of the unstable behavior, the second CEC2017 benchmark function was removed. Algorithms with better performance in solving unimodal and multi-modal benchmark functions have more ability in exploration and exploitation. An algorithm with more accuracy in optimizing hybrid and composite functions has more ability to escape from local optima. The global optimum for CEC2017_01 and CEC2017_03–30 are 100, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400, 2500, 2600, 2700, 2800, 2900, and 3000, respectively. The global optima of the classic benchmark functions and CEC2019 and their characteristics are listed in Tables A1 and A2 in the Appendix A.

3.2. Engineering Design Optimization Problems

The analysis of PCOA's and its competitors' performance in optimizing engineering design problems involved using twenty-three problems. CEC2006 comprises a tension/compression spring [61], gear train [27], speed reducer [62], and three models of a 25-bar truss [15]. CEC2011 contains constrained problems, and several linear and non-linear constraints increase their complexity. Therefore, using these problems shows the ability of PCOA to solve engineering problems. These problems comprise parameter estimation for frequency-modulated (FM) sound waves (CEC11_01); the Lennard–Jones potential problem (CEC11_02); optimal control of a non-linear stirred tank reactor (CEC11_03); the transmission network expansion planning (TNEP) problem (CEC11_04); the large-scale transmission pricing problem (CEC11_05); the circular antenna array design problem (CEC11_06); the static economic load dispatch (ELD) problem (CEC11_07); different models of dynamic economic dispatch (DED) problems (CEC11_08–12); different models of hydrothermal scheduling problems (CEC11_13–15); the messenger: spacecraft trajectory optimization problem (CEC11_16); and the Cassini 2: spacecraft trajectory optimization problem (CEC11_17). This study uses the Lagrangian method to handle the constrained CEC2006 and CEC2011 problems.

3.3. PCOA Competitors

In this study, a range of well-known, new, and state-of-the-art algorithms are chosen for comparative reference to demonstrate PCOA's performance in relation to various algorithms. Among the well-known algorithms are PSO, DE, and WOA; AVOA, RW_GWO, HHO, and gradient-based optimizer (GBO) are new algorithms; and linear population size reduction (LSHADE) and Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase (EBOwithCMAR) represent state-of-the-art optimization algorithms.

3.4. PCOA Competitors' Parameter Settings

Table 1 tabulates the parameter settings of PCOA's competitors. The population size and maximum number of iterations for the investigated algorithms were selected so that FES_{max} was the same for them. In this table, the parameters of AVOA, RW_GWO, WOA, HHO, and GBO are fixed; no sensitivity analysis is needed. The parameters for PSO and DE are determined based on sensitivity analyses from previous studies [22,63], which had experimental settings similar to the present study. The parameters of LSHADE and EBOwithCMAR are determined by the sensitivity analyses in the studies of [63,64].

Table 1. Parameter settings of PCOA’s competitors.

Algorithm	Parameters	Parameter Values
AVOA	$L_1, L_2, K, P_1, P_2, P_3$	0.8, 0.2, 2.5, 0.6, 0.4, and 0.6 [65]
PSO	C_1, C_2 , inertia weight	2, 2, linearly decreasing from 0.9 to 0.1 [66]
DE	Lower bound of scaling factor (β_{min}), upper bound of scaling factor (β_{max}), crossover probability	0.2, 0.8, 0.2
RW_GWO	Convergence parameter (a)	Linearly decreasing from 2 to 0
WOA	Convergence parameter (a)	Linearly decreasing from 2 to 0
HHO	β	1.5
LSHADE	Archive size rate, initial population size, top individuals present, memory size	2, 20, 0.1, 5
GBO	B_{min}, B_{max}, Pr	0.2, 1.2, 0.5
EBOwithCMAR	Maximum primary population size, minimum primary population size, maximum secondary population size, minimum secondary population size, memory size, third population size, local search probability	$18.dim, 4, 46.8.dim, 10, 6, 4 + 3.log(dim), 0.1$

3.5. Analyzing the Influence of the PCOA Parameters

In this study, we analyzed the influence of the parameters of the proposed algorithm using a sensitivity analysis. The method for analyzing the influence of the PCOA parameters was as follows:

- To perform a sensitivity analysis on the PCOA, select one parameter (either N_Tree, N_Cone, N_Cycle, P1, or P2).
- Determine one value for the selected parameter.
- Execute the PCOA algorithm multiple times and compute the fitness function.
- If there is another value for the investigated parameters, go to step 3; otherwise, go to step 5.
- Normalize the obtained fitness function using the map min–max method as follows:

$$fitness_{normalized,j} = \frac{\overline{fitness_j} - \min(\overline{fitness_j})}{\max(\overline{fitness_j}) - \min(\overline{fitness_j})} \tag{20}$$

In Equation (20), $fitness_{normalized,j}$ is the normalized fitness function for the j th parameter, $\overline{fitness_j}$ is the average fitness function for the j th parameter in the n th random run. The value of n equals the number of random runs for each benchmark function or engineering problem set.

- Consider the parameter setting with a minimum $fitness_{normalized,j}$ as the desired value.
- If there is another parameter for sensitivity analysis, go to step 2. For each parameter, plot the sensitivity analysis results. The plot displays the minimum value of the fitness function as closest to zero (green color), whereas values closer to one (red color) indicate the maximum values.

3.6. Statistical Tests

Wilcoxon signed-rank and Friedman tests, as non-parametric tests, were employed to compare PCOA with the other investigated algorithms. The study conducted by [67] recommended the Wilcoxon signed-rank test for pair-wise comparisons. The Wilcoxon test returns the number of victories and defeats of one algorithm against the others. The Friedman test determines the algorithm’s mean rank and is used in many studies, such as [66,67].

4. Results

In this section, a sensitivity analysis of PCOA is carried out, and the best parameters of PCOA are reported. Then, the robustness, convergence efficiency, and time complexity of PCOA are evaluated on 62 benchmark functions and 21 engineering design problems.

The benchmark functions test suite comprises 23 classical benchmark functions, 29 CEC2017 in 10, 30, 50, and 100 dimensions, and 10 CEC2019. The engineering design problems include 6 CEC2006 and 18 CEC2011.

4.1. Sensitivity Analysis

In this study, the parameters of PCOA are obtained using a sensitivity analysis. The sensitivity analysis method presented by [22] is employed. This method uses the min–max normalization method to scale the objective function between 0 and 1. The results are clearly shown using this method. Figures 4 and A1–A7 show the results of the PCOA sensitivity analysis. These figures clearly show the best results (represented by a green color) and the worst results (represented by a red color) which are obtained by employing the method proposed by [22] and using the min–max normalization method to scale the objective function between 0 and 1. As seen for the classic benchmark functions, CEC2019 and CEC2006, the best N_Tree was equal to 2. In contrast, the best N_Tree for CEC2017 was equal to 2 and 30. About CEC2006 and CEC2011, the best value of N_Tree was 20 and 10, respectively. About N_Cone, the best values for the classic benchmark functions, CEC2017, CEC2019, CEC2006, and CEC2011, were equal to 20, 5, 5, 20, and 5, respectively. The best values of N_Cycle for the classic benchmark functions, CEC2017, CEC2019, CEC2006, and CEC2011, were 5, 50, 100, 5, and 10, respectively. Considering P1, the best value was 0.05. The best value for P2 was equal to 0.8. Regarding all results, the default values for N_Tree, N_Cone, N_Cycle, P1, and P2 were 2, 5, 50, 0.05, and 0.8, respectively.

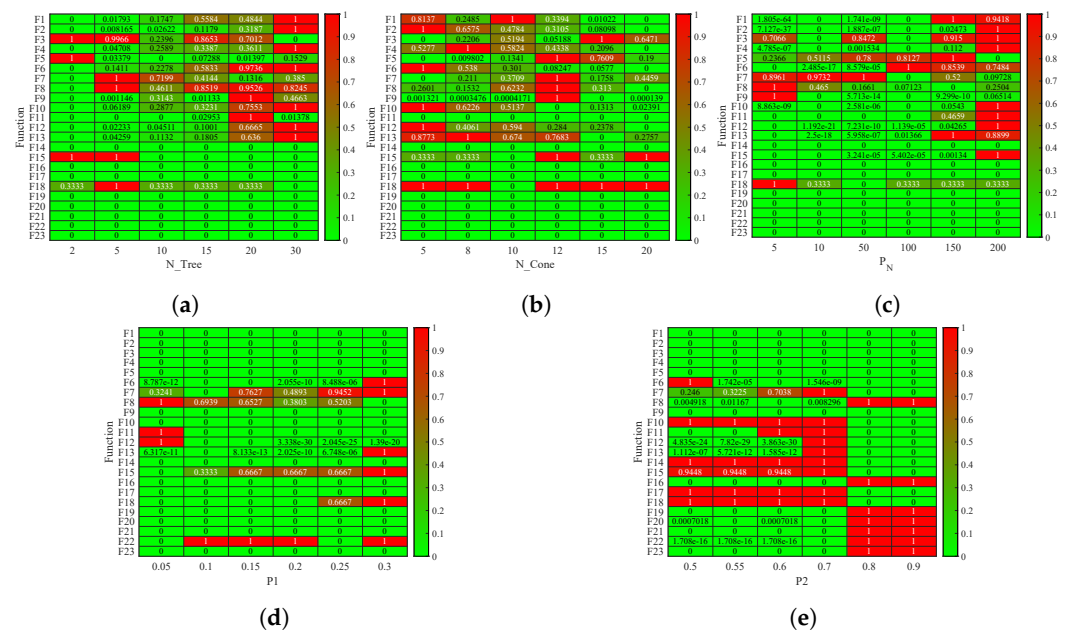


Figure 4. PCOA sensitivity analysis using classic benchmark functions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

4.2. The Benchmark Set and Compared Algorithms

A comparison was made between PCOA and AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR to assess their ability to solve the classic benchmark functions, CEC2017 and CEC2019. Table A3 presents the minimum (min), mean, median, maximum (max), and standard deviation (std) results of PCOA and other approaches for solving the classical benchmark functions. PCOA reached the global optimum for F1–F6, F9, F11, F17–F18, F20, and F21. Regarding the remaining classical benchmark functions, PCOA approached the global optimum with high accuracy. In Figure 5, the Wilcoxon test determines the number of victories, equalities, and defeats of PCOA compared to other competitors. According to the results, the numbers of PCOA’s victories against AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR were equal to 14,

19, 18, 19, 20, 19, 8, 15, and 6, respectively. Based on the mentioned results, the superiority of PCOA compared to AVOA, PSO, DE, RW_GWO, WOA, HHO, and GBO is clear in optimizing the investigated classical benchmark functions. The equality results in this figure show that PCOA had 13 and 14 equalities versus LSHADE and EBOwithCMAR. Hence, in the classical benchmark functions, PCOA's results were comparable to those for LSHADE and EBOwithCMAR. In Figure 6, the Friedman ranking test is employed to show the rank of each algorithm. The Friedman ranking analysis shows that PCOA was ranked first. Meanwhile, EBOwithCMAR and LSHADE rank second and third, respectively. The PCOA's rank (2.90) was 28.92% and 26.77% less than LSHADE and EBOwithCMAR.

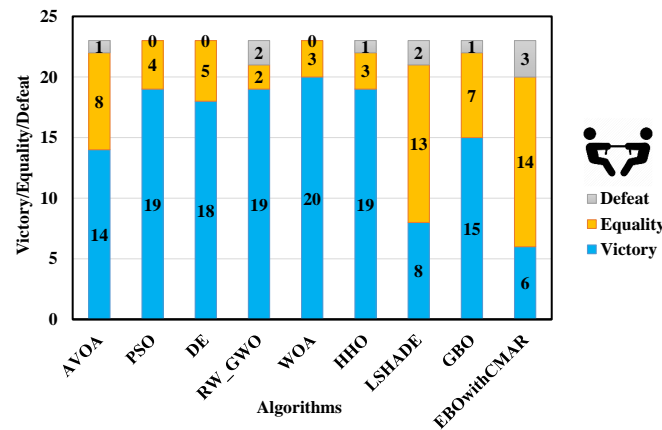


Figure 5. Comparison of Wilcoxon test for PCOA and other algorithms using the classic benchmark functions over 51 independent runs and 150,000 NFES.

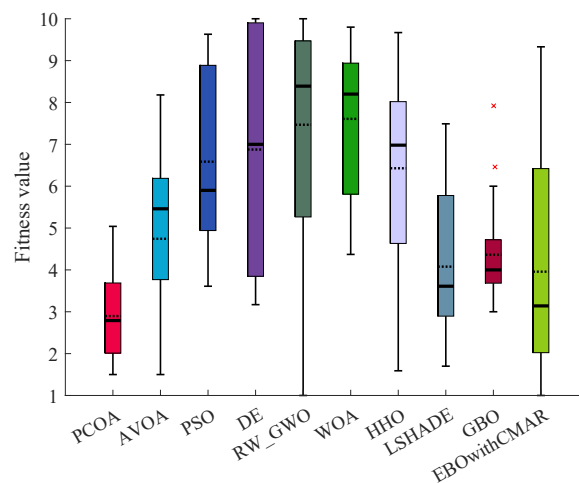


Figure 6. Comparison of Friedman ranking for PCOA and other algorithms using the classic benchmark functions over 51 independent runs and 150,000 NFES.

Table A4 lists the statistical criteria of PCOA and other approaches for optimizing CEC2017 over 10 dimensions. Regarding unimodal benchmark functions (CEC17_01 and CEC17_02), PCOA could consistently find a global optimum solution. Regarding multi-modal benchmark functions (CEC17_04–10), PCOA could consistently find global optimum solutions for CEC17_04–06 and CEC17_08–09. PCOA approached the global optimum solutions with reasonable accuracy in the remaining multi-modal functions. Regarding hybrid benchmark functions (CEC17_11–20), PCOA could compute the global optimum of CEC17_11 consistently. In the remaining hybrid functions, the solutions of PCOA were close to the global optimum, and sometimes PCOA approached the global optimum solutions with 100% accuracy. About the composite benchmark functions (CEC17_21–30),

PCOA became trapped in local optimum solutions in solving CEC17_21, CEC17_23–25, CEC17_27, and CEC17_29–30. This is an issue because of the many local optimums and complex structures of the mentioned functions. However, PCOA’s results were close to the optimum solutions in the remaining composite functions. PCOA could find the global optima of CEC17_22, CEC17_26, and CEC17_28 in some runs. A comparison of the results of PCOA and its competitors based on the Wilcoxon test (Figure 7) showed that PCOA had 28 victories over AVOA, 25 victories over PSO and DE, 27 victories over RW_GWO, 29 victories over WOA and HHO, and 28 victories over GBO. Compared to LSHADE, PCOA had 11 victories and 4 equalities. The number of victories and equalities of PCOA against EBOwithCMAR were equal to 4 and 5, respectively. The Friedman test results (Figure 8) showed that PCOA had a lower rank than other approaches, followed by LSHADE and EBOwithCMAR.

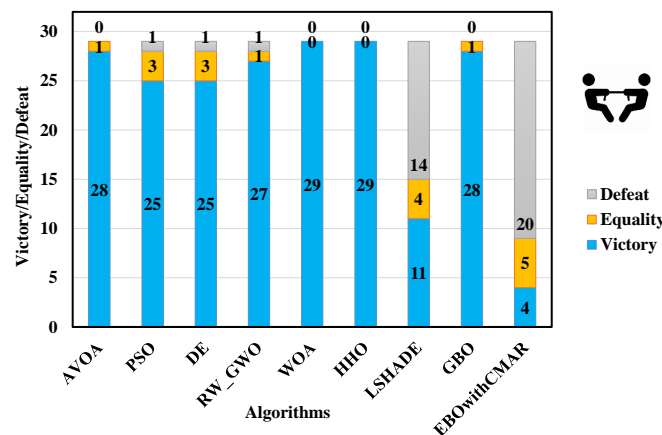


Figure 7. Comparison of Wilcoxon test for PCOA and other algorithms using the 10D CEC2017 benchmark functions over 51 independent runs and 100,000 NFES.

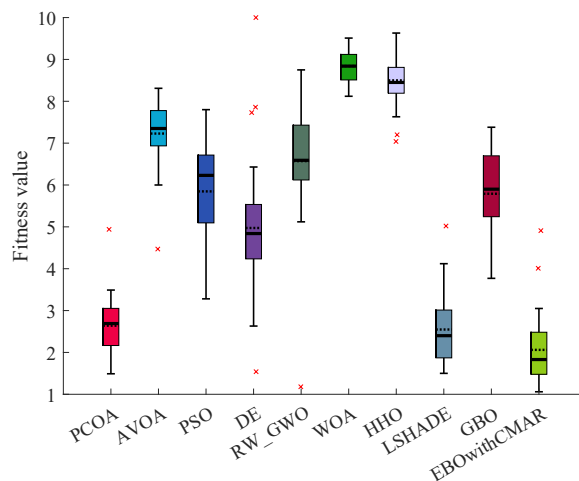


Figure 8. Comparison of Friedman ranking for PCOA and other algorithms using the 10D CEC2017 benchmark functions over 51 independent runs and 100,000 NFES.

Table A5 tabulates the results of solving CEC2017 over 30 dimensions using PCOA, AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. On the unimodal benchmark functions, PCOA could find global optimum solutions for them in all runs. Regarding the multi-modal benchmark functions, PCOA consistently obtained the global optimum solutions of CEC17_04, CEC17_06, and CEC17_09. In the remaining unimodal functions, PCOA estimated optimum solutions with good precision. Regarding the hybrid benchmark functions, PCOA could find global optimum solutions of CEC17_11 and,

in the remaining functions, approached the optimum solutions with reasonable accuracy. Regarding the composite benchmark functions, PCOA was trapped in a local optimum because of the large number of local optimum solutions and the challenging structure of these functions. However, the accuracy of PCOA in solving the composite functions was comparable with other algorithms. According to the Wilcoxon test results for CEC2017 over 30 dimensions (Figure A8), PCOA won in 29, 27, 28, 28, 29, 29, 13, 28, and 9 functions, respectively. According to the Friedman test results (Figure A9), PCOA achieved the highest rank, followed by EBOwithCMAR. Table A6 displays the results of solving CEC2017 over 50 dimensions using PCOA, AVOA, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. Based on the results, for unimodal CEC2017, PCOA achieved the global optimum solutions consistently. Regarding multi-modal CEC2017, PCOA reached the global optimum of CEC17_04, CEC17_06, and CEC17_09 with an accuracy of 100%. PCOA's precision in approximating global optimum solutions in the remaining multi-modal CEC2017 was reasonable compared to other competitors. Regarding hybrid and composite CEC2017, PCOA had a challenge finding global optimum solutions. However, the results of PCOA competed with the other investigated algorithms. The Wilcoxon test (Figure A10) and the Friedman test (Figure A11) showed the superiority of PCOA over the other competitors. PCOA was the winner in 19 and 18 functions over LSHADE and EBOwithCMAR. In comparison with the state-of-the-art algorithms, the Friedman's ranking of PCOA was 31.91% and 20.62% better than LSHADE and EBOwithCMAR. Table A7 compares PCOA and its competitors using CEC2017 over 100 dimensions. PCOA obtained global optimum solutions (CEC17_01 and CEC17_02) and some multi-modal CEC2017 (CEC17_04, CEC17_06, and CEC17_09). In the remaining CEC2017, PCOA performed better than the other competitors in most conditions. The Wilcoxon test results (Figure A12) showed that PCOA had 29 victories over AVOA, PSO, WOA, HHO, and GBO, 28 victories over DE and RW_GWO, 22 victories over LSHADE, and 17 victories over EBOwithCMAR. Figure A13 shows the Friedman ranking results in solving CEC2017 with 100 dimensions. As seen, PCOA took the first rank (1.68), which was 39.37% and 18.43% less than the third- (LSHADE) and second- (EBOwithCMAR) best algorithms. Considering the CEC2017 benchmark functions in all dimensions, PCOA had superiority over AVOA, PSO, DE, RW_GWO, WOA, HHO, and GBO. In 10 and 30 dimensions, the results of PCOA were close to the winners of CEC2017, LSHADE and EBOwithCMAR, while in the remaining dimensions, PCOA had more accuracy. Regarding the quality of PCOA's results, it can be recognized that PCOA's results were comparable to those of the other investigated algorithms. Table A8 presents the min, mean, median, max, and std results of solving the CEC2019 benchmark functions using PCOA and the other approaches. Based on the findings of this table, the proposed algorithm performed better than AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. PCOA could find a global optimum for CEC19_01 and CEC19_03–CEC19_06. For other benchmark functions, the PCOA results were comparable with other investigated approaches. The Wilcoxon test in Figure 9 revealed that PCOA had more victories over AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, and GBO. In comparison with EBOwithCMAR, PCOA had three victories and one equality. According to the Friedman ranking, PCOA achieved the first rank. The Friedman ranking of PCOA (Figure 10) was 24.36% and 21.69%, better than LSHADE (the third-best algorithm) and EBOwithCMAR (the second-best algorithm).

According to the obtained results for the benchmark functions, in most benchmark functions PCOA had better performance than AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. Comparing statistical criteria over 51 random runs (min, mean, median, max, and std), the Wilcoxon and Friedman ranking tests confirm this.

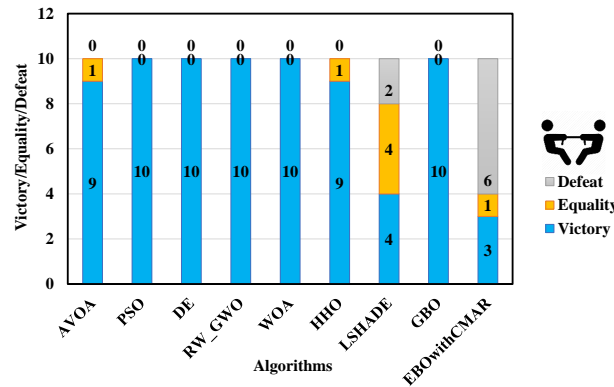


Figure 9. Comparison of Wilcoxon test for PCOA and other algorithms using the CEC2019 benchmark functions over 51 independent runs.

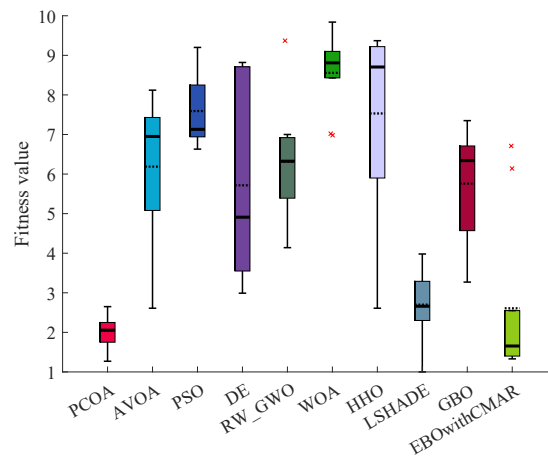


Figure 10. Comparison of Friedman ranking for PCOA and other algorithms using the CEC2019 benchmark functions over 51 independent runs.

4.3. The Engineering Problems and the Compared Algorithms

A summary of the statistical results (min, mean, median, max, and std) of the PCOA and other approaches to solving the CEC2006 engineering problems is presented in Table A9. The results of this table are for 51 random runs. PCOA performed better than AVOA, PSO, DE, RW_GWO, WOA, HHO, and GBO in solving CEC06_01–02 and CEC06_04–06. For CEC06_03, the results of PCOA were better than AVOA, RW_GWO, WOA, and HHO. According to the Wilcoxon test (Figure 11) and Friedman test (Figure 12), PCOA was better than AVOA, PSO, DE, RW_GWO, WOA, HHO, and GBO.

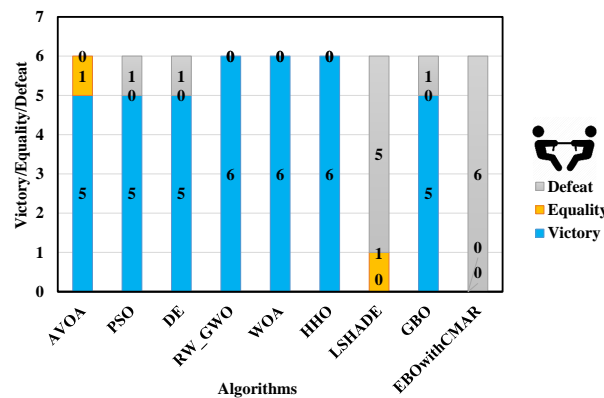


Figure 11. Comparison of Wilcoxon test for PCOA and other algorithms using the CEC2006 benchmark functions over 51 independent runs.

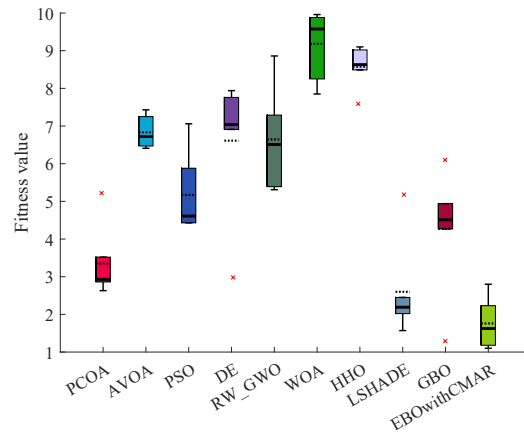


Figure 12. Comparison of Friedman ranking for PCOA and other algorithms using the CEC2006 benchmark functions over 51 independent runs.

Table A10 displays the optimization results for CEC2011 by PCOA and its competitors. PCOA outperformed its competitors in solving CEC11_03, CEC11_05, CEC11_07, CEC11_11–13, and CEC11_15–17. Considering the Wilcoxon test results (Figure 13), PCOA had 11, 14, 15, 13, 15, 15, 11, 13, and 10 victories over AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. Regarding the Friedman test (Figure 14), the proposed algorithm was ranked first (rank = 1.94). LSHADE achieved the second-ranking (rank = 2.60) while EBOwithCMAR obtained the third-ranking (rank = 3.25). Therefore, the results show that PCOA performed competitively compared to the other investigated approaches in optimizing engineering problems.

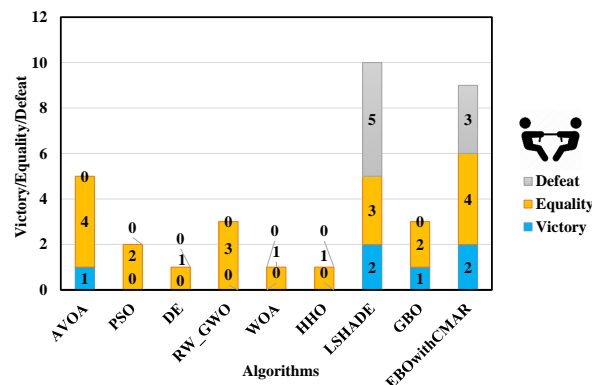


Figure 13. Comparison of Wilcoxon test for PCOA and other algorithms using the CEC2011 benchmark functions over 25 independent runs.

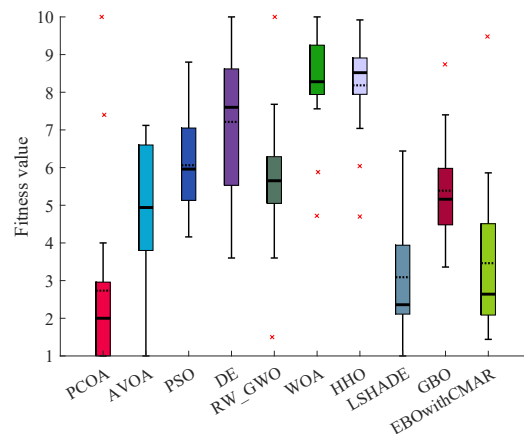


Figure 14. Comparison of Friedman ranking for PCOA and other algorithms using the CEC2011 benchmark functions over 25 independent runs.

4.4. PCOA Convergence Analysis

This subsection analyzes the PCOA’s and competitors’ convergence efficacies using violin plots (Figure 15) and convergence curves (Figure 16). The violin plot shows the distribution of the algorithm’s results. A broader violin shows more uncertainty and less quality in this plot, and vice versa. Based on the violin plots shown in Figure 15, PCOA converged to optimum values in almost all conditions. The difference between the maximum and minimum solutions of PCOA was significantly less than other algorithms. This issue shows that the results of PCOA are consistent. Considering Figure 16, PCOA had a reasonable convergence speed compared to the competitors. The following equations are used to ensure that algorithms converge.

$$Conv_R = \left| \frac{Obj_{opt} - \overrightarrow{Obj}}{Obj_{opt} - Obj_0} \right| \tag{21}$$

In Equation (20), Obj_{opt} is globally optimal, the \overrightarrow{Obj} is the objective function in the last generation, and Obj_0 is the objective function in the initial generation.

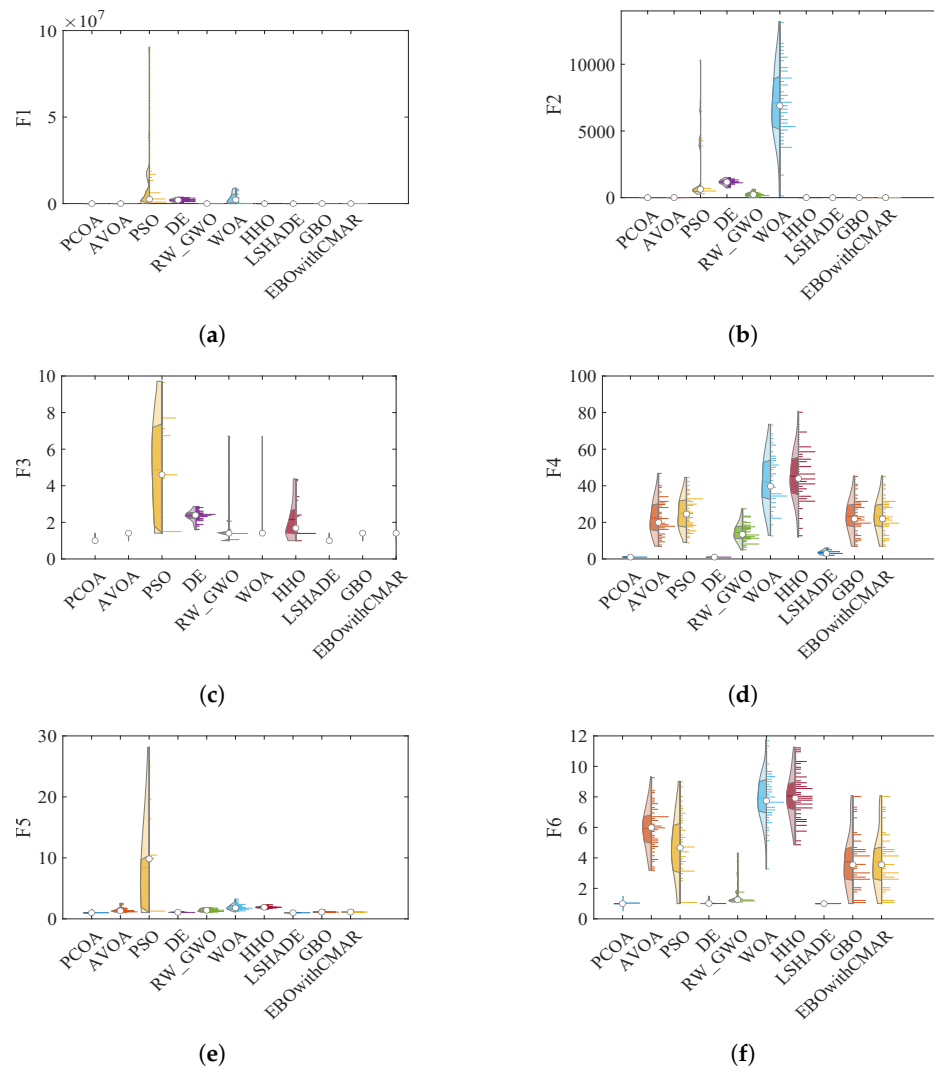


Figure 15. Cont.

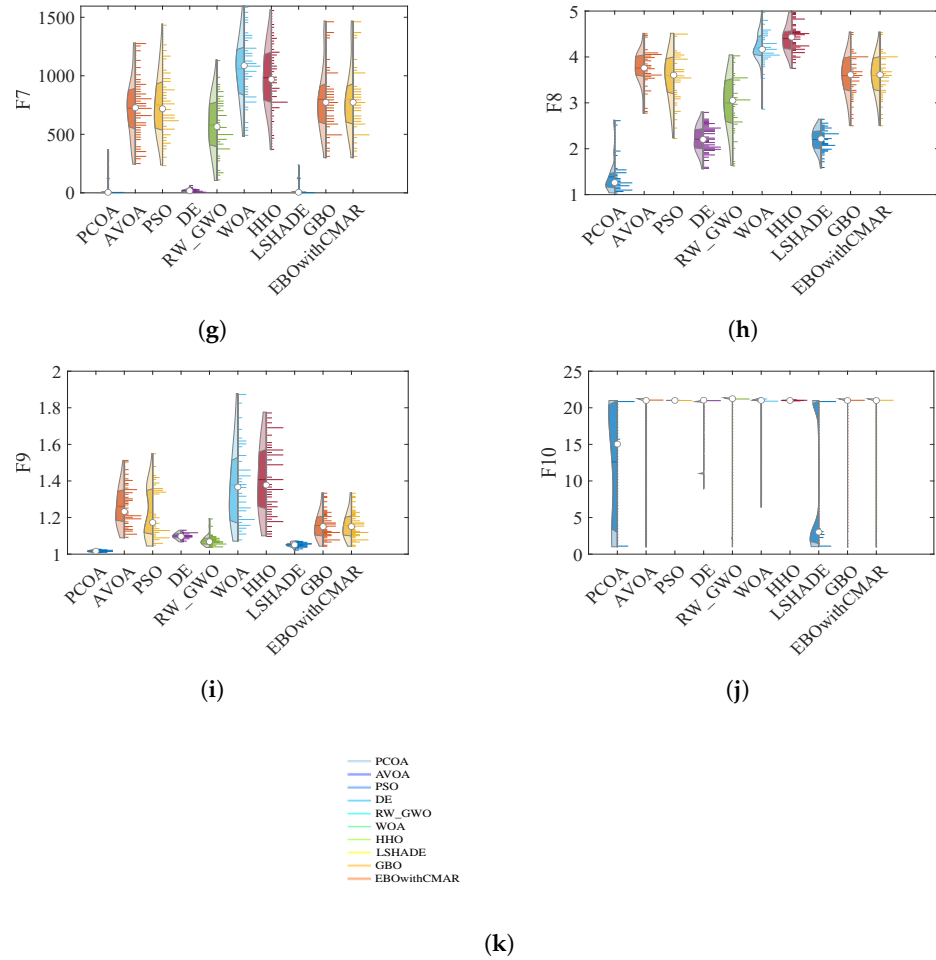


Figure 15. Violin plots of PCOA and competitors in optimizing CEC2019 benchmark functions. (a) CEC19_01; (b) CEC19_02; (c) CEC19_03; (d) CEC19_04; (e) CEC19_05; (f) CEC19_06; (g) CEC19_07; (h) CEC19_08; (i) CEC19_09; (j) CEC19_10; (k) Legend.

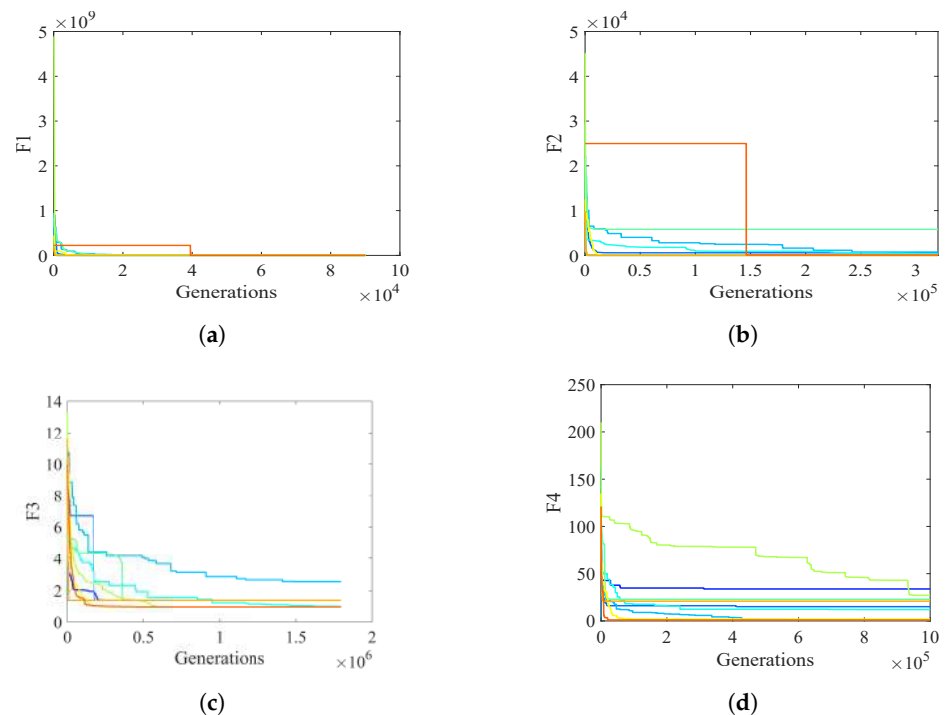


Figure 16. Cont.

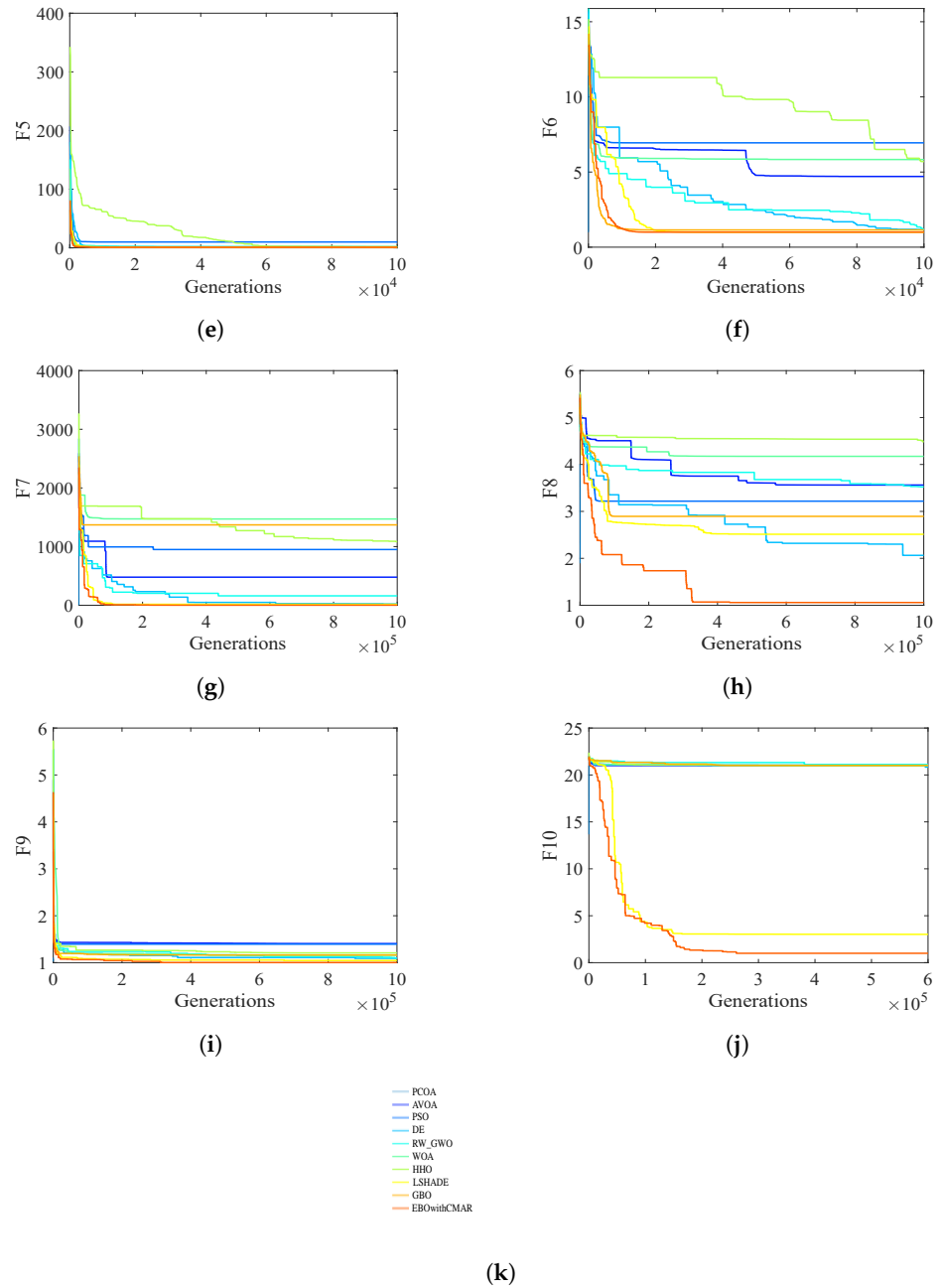


Figure 16. Convergence plots of PCOA and competitors in optimizing CEC2019 benchmark functions. (a) CEC19_01; (b) CEC19_02; (c) CEC19_03; (d) CEC19_04; (e) CEC19_05; (f) CEC19_06; (g) CEC19_07; (h) CEC19_08; (i) CEC19_09; (j) CEC19_10; (k) Legend.

Table 2 shows the $Conv_R$ results for PCOA, AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR on distinct problems. According to the results of solving the classic, CEC2017, CEC2019, CEC2006, and CEC2011 functions, convergence is guaranteed in almost all conditions for PCOA. PCOA's convergence rate ($Conv_R$) competes with other algorithms.

Table 2. Convergence rate analysis of PCOA and competitors using the classic, CEC2017, CEC2019, CEC2006, and CEC2011 benchmark functions.

Function	PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
F1	1.00	1.00	0.99	0.97	1.00	1.00	1.00	0.99	1.00	0.99
F2	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.99
F3	1.00	0.99	0.98	0.55	0.99	0.81	1.00	0.99	1.00	0.99
F4	1.00	0.96	0.93	0.75	0.99	0.79	1.00	0.97	1.00	0.98
F5	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	0.99
F6	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F7	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00
F8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F9	1.94	0.56	0.43	0.45	0.53	0.59	0.58	0.49	0.46	1.22
F10	1.00	0.96	0.89	0.89	0.99	0.99	1.00	0.97	1.00	-
F11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F13	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00
F14	1.00	1.00	0.99	1.00	1.00	1.00	1.00	0.99	0.99	1.00
F15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F17	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F21	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
CEC17_01	0.99	1.00	1.00	0.99	0.99	0.99	0.96	1.00	1.00	1.00
CEC17_03	0.77	0.99	0.95	0.96	0.97	0.98	0.94	0.99	0.99	0.99
CEC17_04	0.98	0.99	1.00	0.99	0.99	0.99	0.98	1.00	1.00	1.00
CEC17_05	0.95	0.80	0.87	0.74	0.75	0.76	0.64	0.91	0.82	0.95
CEC17_06	0.96	0.79	0.92	0.98	0.85	0.66	0.56	0.99	0.89	1.00
CEC17_07	0.94	0.88	0.96	0.91	0.90	0.86	0.82	0.96	0.86	0.95
CEC17_08	0.94	0.82	0.88	0.73	0.74	0.76	0.70	0.92	0.84	0.95
CEC17_09	0.98	0.91	0.98	0.96	0.96	0.89	0.89	1.00	0.95	1.00
CEC17_10	0.75	0.69	0.77	0.40	0.48	0.63	0.60	0.73	0.78	0.77
CEC17_11	0.97	0.99	1.00	0.98	0.98	0.99	0.97	0.99	1.00	0.99
CEC17_12	0.90	1.00	1.00	0.99	0.99	1.00	0.99	1.00	1.00	1.00
CEC17_13	0.97	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
CEC17_14	0.88	0.99	0.99	0.98	1.00	1.00	0.99	0.99	1.00	1.00
CEC17_15	0.89	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
CEC17_16	0.90	0.89	0.94	0.85	0.88	0.87	0.76	0.94	0.93	0.95
CEC17_17	0.88	0.98	0.94	0.96	0.96	0.95	0.94	0.97	0.96	0.99
CEC17_18	0.88	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00
CEC17_19	0.97	0.98	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
CEC17_20	0.82	0.82	0.89	0.77	0.78	0.73	0.71	0.87	0.89	0.90
CEC17_21	0.47	0.65	0.65	0.59	0.60	0.54	0.44	0.74	0.69	0.74
CEC17_22	0.98	0.86	0.84	0.56	0.63	0.75	0.68	0.88	0.90	0.96
CEC17_23	0.75	0.66	0.65	0.64	0.61	0.60	0.46	0.68	0.60	0.71
CEC17_24	0.48	0.78	0.77	0.64	0.73	0.72	0.55	0.80	0.70	0.81
CEC17_25	0.65	0.97	0.92	0.86	0.91	0.97	0.93	0.93	0.90	0.89
CEC17_26	0.86	0.94	0.99	0.83	0.90	0.86	0.79	0.88	0.98	0.93
CEC17_27	0.62	0.73	0.76	0.72	0.78	0.74	0.66	0.74	0.70	0.72
CEC17_28	0.96	0.91	0.91	0.87	0.88	0.90	0.87	0.90	0.90	0.90
CEC17_29	0.82	0.92	0.87	0.87	0.90	0.88	0.83	0.91	0.90	0.89
CEC17_30	0.90	1.00	1.00	0.99	1.00	1.00	0.97	1.00	1.00	1.00
CEC19_01	1.00	1.00	0.99	0.97	1.00	0.97	1.00	0.99	1.00	0.56
CEC19_02	0.79	1.00	0.97	0.89	0.95	0.80	1.00	0.99	1.00	0.54
CEC19_03	0.94	0.96	0.90	0.78	0.93	0.91	0.97	0.98	0.96	0.99
CEC19_04	0.98	0.79	0.84	0.98	0.89	0.83	0.68	0.98	0.80	1.00
CEC19_05	0.84	0.99	0.95	0.99	0.99	0.99	0.93	0.99	0.99	1.00
CEC19_06	0.93	0.68	0.58	0.82	0.86	0.63	0.39	0.95	0.97	0.97
CEC19_07	0.94	0.79	0.62	0.94	0.90	0.31	0.59	0.98	0.46	0.99
CEC19_08	0.38	0.37	0.45	0.58	0.36	0.25	0.22	0.61	0.55	0.91
CEC19_09	0.67	0.89	0.90	0.97	0.96	0.94	0.95	0.98	0.93	0.99
CEC19_10	0.13	0.05	0.04	0.04	0.03	0.04	0.06	0.82	0.04	0.91
CEC06_01	0.99	0.99	0.90	0.94	0.98	0.99	0.98	0.98	0.99	0.97
CEC06_02	1.00	0.99	0.97	0.99	0.99	0.99	0.99	0.98	0.99	0.99
CEC06_03	0.70	0.65	0.69	0.78	0.99	0.99	0.80	0.79	0.68	0.55
CEC06_04	0.97	0.95	0.91	0.85	0.93	0.81	0.83	0.93	0.91	0.97

Table 2. Cont.

Function	PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC06_05	0.99	0.96	0.92	0.91	0.96	0.92	0.78	0.94	0.96	0.96
CEC06_06	0.97	0.96	0.95	0.84	0.95	0.94	0.82	0.94	0.96	0.93
CEC11_01	1.00	0.93	0.69	0.25	0.70	0.62	0.44	0.89	0.98	0.91
CEC11_02	1.00	0.57	0.45	0.40	0.48	0.56	0.45	0.34	0.61	0.65
CEC11_03	1.00	0.94	0.99	0.91	0.00	0.91	0.85	0.97	0.91	0.83
CEC11_04	1.00	0.69	0.27	0.27	0.74	0.27		1.00	0.16	0.79
CEC11_05	1.00	1.00	0.85	0.78	0.59	0.62	0.34	0.62	0.85	0.68
CEC11_06	1.00	0.00	0.95	0.72	0.00	0.96	0.77	0.98	0.99	0.98
CEC11_07	1.00	1.00	0.86	0.94	0.72	1.00	0.99	0.99	0.98	0.11
CEC11_08	1.00	0.00	0.52	0.52	0.43	0.65	0.53	0.91	0.85	0.06
CEC11_09	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	0.53
CEC11_10	1.00	0.89	0.87	0.85	0.95	1.00	0.99	0.99	0.96	0.51
CEC11_11	1.00	1.00	1.00	0.97	1.00	0.99	0.97	0.94	0.96	0.51
CEC11_12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.50
CEC11_13	1.00	0.99	0.83	0.69	0.69	0.59	0.76	0.99	0.99	0.47
CEC11_14	1.00	1.00	0.97	0.96	0.97	0.99	0.95	0.99	1.00	0.99
CEC11_15	1.00	0.99	0.97	0.96	0.96	0.99	0.95	0.99	1.00	0.99
CEC11_16	1.00	0.99	0.96	0.96	0.96	0.98	0.96	0.99	0.98	0.99

4.5. Complexity Evaluation

The complexity evaluation is carried out by estimating the T_0 , T_1 , and T_2 values. The computation time of calculating mathematical functions relates to T_0 . T_1 is the computation time of calculating the 18th CEC2017 on 200,000 evaluations and all dimensions. T_2 is the average computation time for executing CEC17_18 with 200,000 evaluations five times. The time complexity is as follows:

$$T_{Complexity} = \frac{T_2 - T_1}{T_0} \tag{22}$$

Table 3 shows the time complexities of PCOA, AVOA, PSO, DE, RW_GWO, WOA, HHO, LSHADE, GBO, and EBOwithCMAR. In 10 and 30 dimensions, the algorithms' complexities were close to each other. However, as the dimensions increased, the time complexity of the PCOA and the state-of-the-art algorithms increased, which is due to the greater efforts of these algorithms in finding optimal solutions to the problem.

Table 3. Algorithm complexity for PCOA and other competitors.

Function	PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC_17 (dim = 10)	1.49	0.09	5.58	1.94	1.05	0.17	31.21	6.63	1.57	45.77
CEC_17 (dim = 30)	1.57	12.20	4.31	0.93	1.60	2.60	49.30	9.02	0.55	113.91
CEC_17 (dim = 50)	157.90	6.99	9.77	2.09	1.43	6.47	70.31	32.78	12.92	214.50
CEC_17 (dim = 100)	241.34	21.63	6.37	5.93	8.11	12.55	136.15	24.19	21.44	373.97

5. Discussion

Figure 17 shows the average of the Friedman's ranking results for all eighty-six benchmark functions. As seen in Figure 17, PCOA, EBOwithCMAR, and LSHADE achieved ranks of 2.42, 2.46, and 2.67, respectively, placing them in the first, second, and third positions. Thus, in terms of accuracy, PCOA performed better than well-known optimization algorithms (PSO, DE, and WOA), new optimization algorithms (AVOA, RW_GWO, HHO, and GBO), and state-of-the-art algorithms (LSHADE and EBOwithCMAR). Regarding the convergence efficacy and time complexity, PCOA had competitive results compared to the other optimization algorithms. The outstanding performance of PCOA is for the following reasons:

- Using pollination operators in Equations (12)–(14) leads to good exploration and convergence to near the space of the global optimum.

- Considering new operators in pine cone dispersal by animals and gravity leads to an approximate global optimum with reasonable accuracy.
- Clustering the search domain into sub-clusters and shrinking the search space increases the chance of finding the global optimum faster and more accurately.
- Using adaptive weight creates a good balance between the exploration and exploitation ability.
- Employing Levy flight distribution and a mathematical optimizer generates diverse solutions and helps PCOA to approximate optimum solutions.

Although PCOA performed better than the other investigated algorithms, in some problems other algorithms performed better. This issue can be explained by the “no free lunch in search and optimization” theorem [1]. According to this theorem, each optimization algorithm performs better than others in some problems.

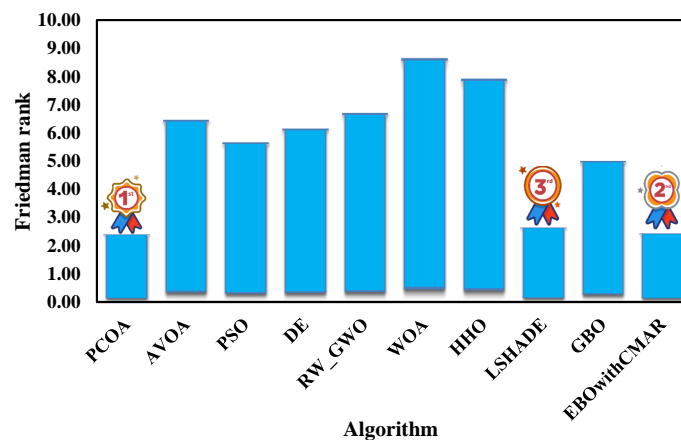


Figure 17. Average Friedman ranking for eighty-five benchmark functions and engineering problems.

6. Conclusions

The present research proposed a new optimization algorithm called the Pine Cone Optimization Algorithm (PCOA). This algorithm works based on the pine tree’s reproduction methods. PCOA benefits from new operators for pollination and cone dispersal by gravity and animals. The utilization of sub-clustering methods within the search domain, coupled with the reduction in the search domain and the incorporation of adaptive weights, leads to a significant increase in the efficacy of the PCOA algorithm. This approach balances exploration and exploitation capabilities and enhances the algorithm’s performance. The performance of this algorithm was investigated in different mathematical benchmark functions and engineering optimization problems. The mathematical benchmark test suite included 23 classic benchmark functions, 29 CEC2017, and 10 CEC2019. There were six CEC2006 and seventeen CEC2011 problems in engineering optimization. The results showed that PCOA was superior to well-known optimization algorithms (PSO, DE, and WOA) and new optimization algorithms (AVOA, RW_GWO, HHO, and GBO). In addition, PCOA was a competitor for state-of-the-art algorithms (LSHADE and EBOwithCMAR). The Friedman rank for PCOA in solving all the test suite problems was 2.42, 1.68 and 9.42 percent better than EBOwithCMAR (second-best algorithm) and LSHADE (third-best algorithm). For executing PCOA, a sensitivity analysis is needed. However, in this study, the researchers reported the default values of the PCOA parameters. Users can use these default parameters to run PCOA with no sensitivity analysis. In order to use this algorithm, programming language software is needed. PCOA has a high potential for solving various complex science and engineering problems.

Author Contributions: Conceptualization, M.V.A. and S.F.; methodology, M.V.A. and S.F.; software, M.V.A. and S.F.; validation, M.V.A. and S.F. formal analysis, M.V.A. and S.F.; investigation, M.V.A. and S.F.; resources, M.V.A. and S.F.; data curation, M.V.A. and S.F.; writing—original draft preparation, M.V.A. and S.F.; writing—review and editing, M.V.A. and S.F.; visualization, M.V.A. and S.F.; supervision, S.F.; project administration, S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

CX	Pine cone’s current position	W_2	Adaptive weight
TX	Tree’s current position	W_3	Adaptive weight
CX^{new}	Updated position of pine cone	μ_{W_1}	Average value for W_1
TX^{new}	Updated position of pine tree	μ_{W_2}	Average value for W_2
X_{best}	The best global position	μ_{W_3}	Average value for W_3
\overline{CX}	The average of all pine cones’ positions	cr	Crossover rate
\overline{TX}	The average of all trees’ positions	μ_{cr}	Average value for cr
X_{animal}	The position of pine cones that are carried by an animal	N_{tree}	Number of tree
$X_{initial}$	The initial position in quadratic programming	$Radius$	Shrinking radius
T_{best}	The top best individuals (10%)	FES	Current function evaluation
$Tpop_{all}$	The memory of PCOA to save solutions	FES_{max}	Maximum number of function evaluations
w_1	Adaptive weight	$Levy$	Random number with Levy flight distribution
w_d	Adaptive weight	P_1	Control parameter
W_1	Adaptive weight	P_2	Control parameter
P_{cycle}	Number of pollination cycle		

Appendix A

Table A1. Characteristics of classic benchmark functions.

Function Formula	[Lb, Ub]	G_{best}	Dimensions
$F_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	0	30
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	0	30
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2$	$[-100, 100]^n$	0	30
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^n$	0	30
$F_5(x) = \sum_{i=1}^n -1[100.(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	0	30
$F_6(x) = \sum_{i=1}^n ((x_i + 0.5))^2$	$[-100, 100]^n$	0	30
$F_7(x) = \sum_{i=1}^n (i.x_i^4 + random[0, 1])$	$[-1.28, 1.28]^n$	0	30
$F_8(x) = \sum_{i=1}^n (-x_i.sin\sqrt{ X_i })$	$[-500, 500]^n$	428.9829.n	30
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10.cos(2.\pi.x_i) + 10]$	$[-5.12, 5.12]^n$	0	30
$F_{10}(x) = -20.exp(-0.2.\sqrt{\frac{1}{n}\sum_{i=1}^n (x_i^2)}) - exp(\frac{1}{n}\sum_{i=1}^n cos(2.\pi.x_i)) + 20$	$[-32, 32]^n$	0	30
$F_{11}(x) = \frac{1}{4000}.\sum_{i=1}^n x_i^2 - \prod_{i=1}^n cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^n$	0	30

Table A1. Cont.

Function Formula	[Lb, Ub]	G _{best}	Dimensions
$F_{12}(x) = \frac{\pi}{n} \{10 \cdot \sin(\pi \cdot y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \cdot \sin^2(\pi \cdot y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_{i+4} u(x_i, 10, 100, 4)}{4} = \begin{cases} k \cdot (x_i - a)^m & x_i > a \\ 0 & 0 < x_i < a \\ k \cdot (-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^n$	0	30
$F_{13}(x) = 0.1 \cdot \{ \sin^2(3 \cdot \pi \cdot x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3 \cdot \pi \cdot x_i + 1)] + (x_n - 1)^2 \cdot [1 + \sin^2(2 \cdot \pi \cdot x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50, 50]^n$	0	30
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + (x_j - a_j)^6} \right)^{-1}$	$[-65.53, 65.53]^2$	1	2
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 \cdot (b_i^2 + b_i \cdot x_2)}{b_i^2 + b_i \cdot x_3 + x_4} \right]^2$	$[-5, 5]^4$	0.00030	4
$F_{16}(x) = 4 \cdot x_1^2 - 2.1 \cdot x_1^4 + \frac{1}{5} \cdot x_1^6 + x_1 \cdot x_2 - 4 \cdot x_2^2 + 4 \cdot x_2^4$	$[-5, 5]^2$	-1.0316	2
$F_{17}(x) = (x_2 - \frac{5.1}{4 \cdot \pi^2} \cdot x_1^2 + \frac{5}{\pi} \cdot x_1 - 6)^2 + 10 \cdot (1 - \frac{1}{8 \cdot \pi}) \cdot \cos(x_1) + 10$	$[-5, 5]^2$	0.398	2
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14 \cdot x_1 + 3 \cdot x_1^2 - 14 \cdot x_2 + 6 \cdot x_1 \cdot x_2 + 3 \cdot x_2^2)] \cdot [30 + (2 \cdot x_1 - 3 \cdot x_2)^2 \cdot (18 - 32 \cdot x_1 + 12 \cdot x_1^2 + 48 \cdot x_2 - 36 \cdot x_1 \cdot x_2 + 27 \cdot x_2^2)]$	$[-2, 2]^2$	3	2
$F_{19}(x) = - \sum_{i=1}^4 c_i \cdot \exp(- \sum_{j=1}^3 b_{ij} \cdot (x_j - p_{ij}))$	$[1, 3]^3$	-3.86	3
$F_{20}(x) = - \sum_{i=1}^4 c_i \cdot \exp(- \sum_{j=1}^6 b_{ij} \cdot (x_j - p_{ij})^2)$	$[0, 1]^6$	-3.32	6
$F_{21}(x) = - \sum_{i=1}^5 [(X - b_i) \cdot (X - b_i)^T + c_i]^{-1}$	$[0, 10]^n$	-10.1532	4
$F_{22}(x) = - \sum_{i=1}^7 [(X - b_i) \cdot (X - b_i)^T + c_i]^{-1}$	$[0, 10]^n$	-10.4028	4
$F_{23}(x) = - \sum_{i=1}^{10} [(X - b_i) \cdot (X - b_i)^T + c_i]^{-1}$	$[0, 10]^n$	-10.5363	4

Table A2. Description of CEC2019 benchmark functions.

Name	Benchmark Function Name	Dimensions	[Lb, Ub]	G _{best}
CEC19_01	STORN'S CHEBYSHEV POLYNOMIAL FITTING PROBLEM	9	$[-8, 192, 8, 192]$	1
CEC19_02	INVERSE HILBERT MATRIX PROBLEM	16	$[-16, 384, 16, 384]$	1
CEC19_03	LENNARD-JONES MINIMUM ENERGY CLUSTER	18	$[-4, 4]$	1
CEC19_04	RASTRIGIN'S FUNCTION	10	$[-100, 100]$	1
CEC19_05	GRIEWANGK'S FUNCTION	10	$[-100, 100]$	1
CEC19_06	WEIERSTRASS FUNCTION	10	$[-100, 100]$	1
CEC19_07	MODIFIED SCHWEFEL'S FUNCTION	10	$[-100, 100]$	1
CEC19_08	EXPANDED SCHAFFER'S CEC06 FUNCTION	10	$[-100, 100]$	1
CEC19_09	HAPPY CAT FUNCTION	10	$[-100, 100]$	1
CEC19_10	ACKLEY FUNCTION	10	$[-100, 100]$	1

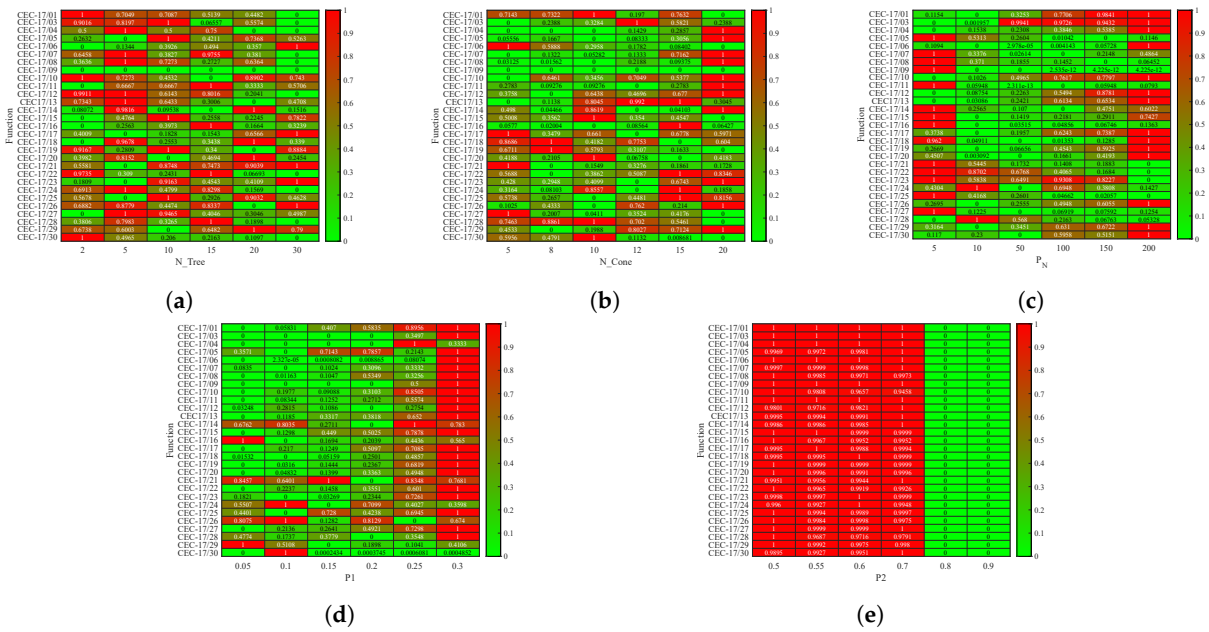


Figure A1. PCOA sensitivity analysis using CEC2017 benchmark functions on 10 dimensions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

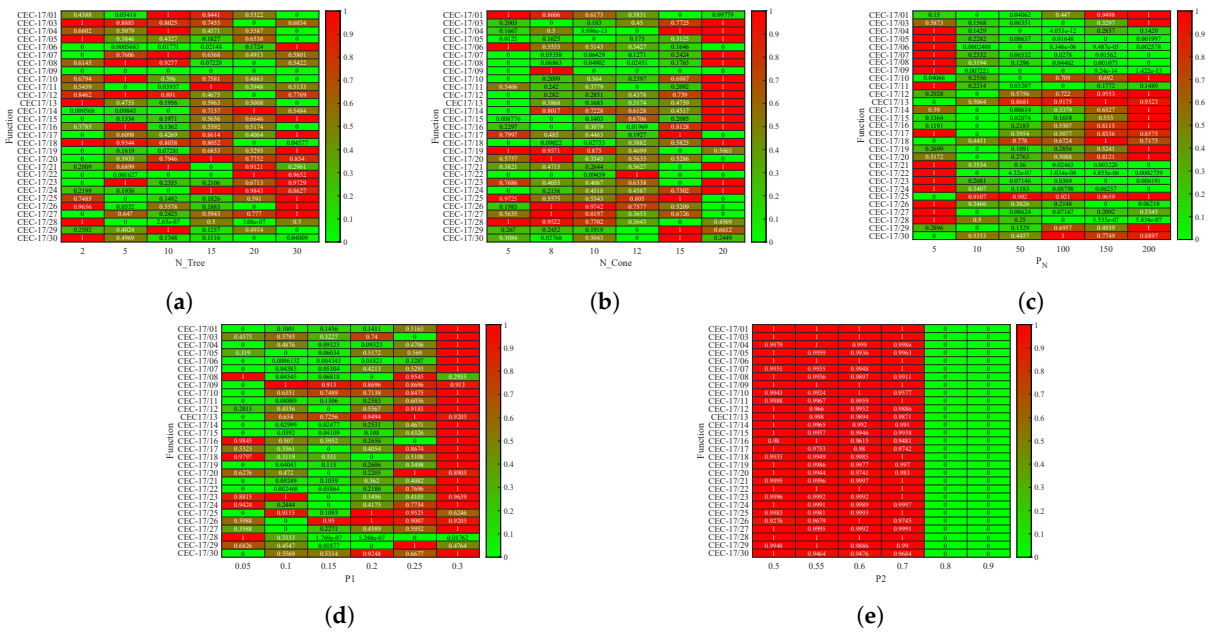


Figure A2. PCOA sensitivity analysis using CEC2017 benchmark functions on 30 dimensions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

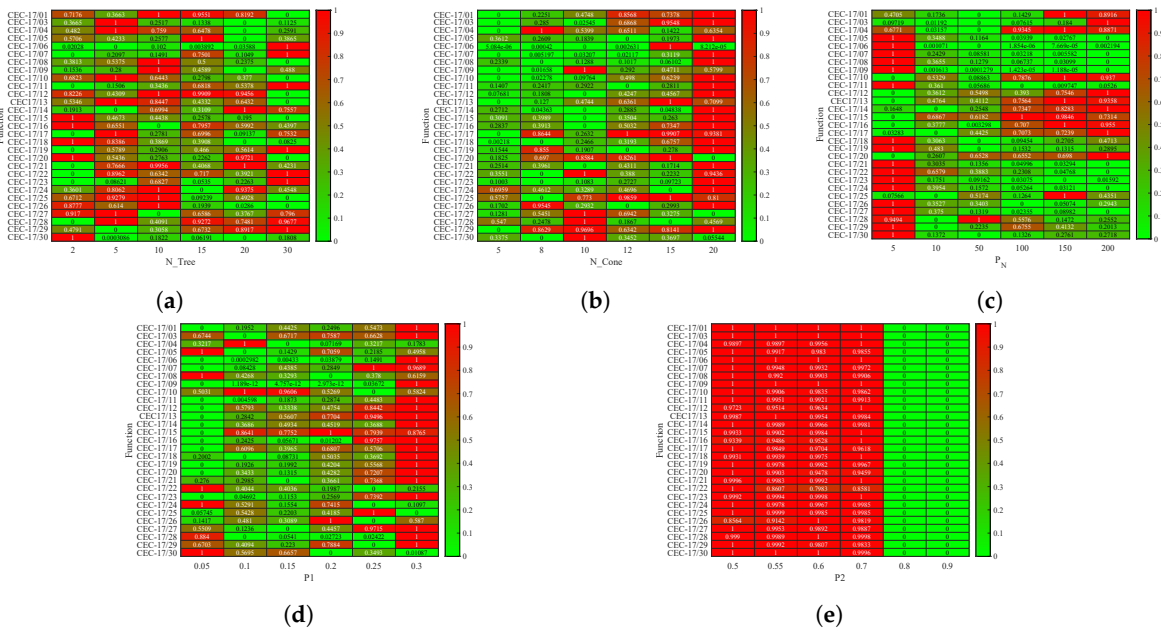


Figure A3. PCOA sensitivity analysis using CEC2017 benchmark functions on 50 dimensions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

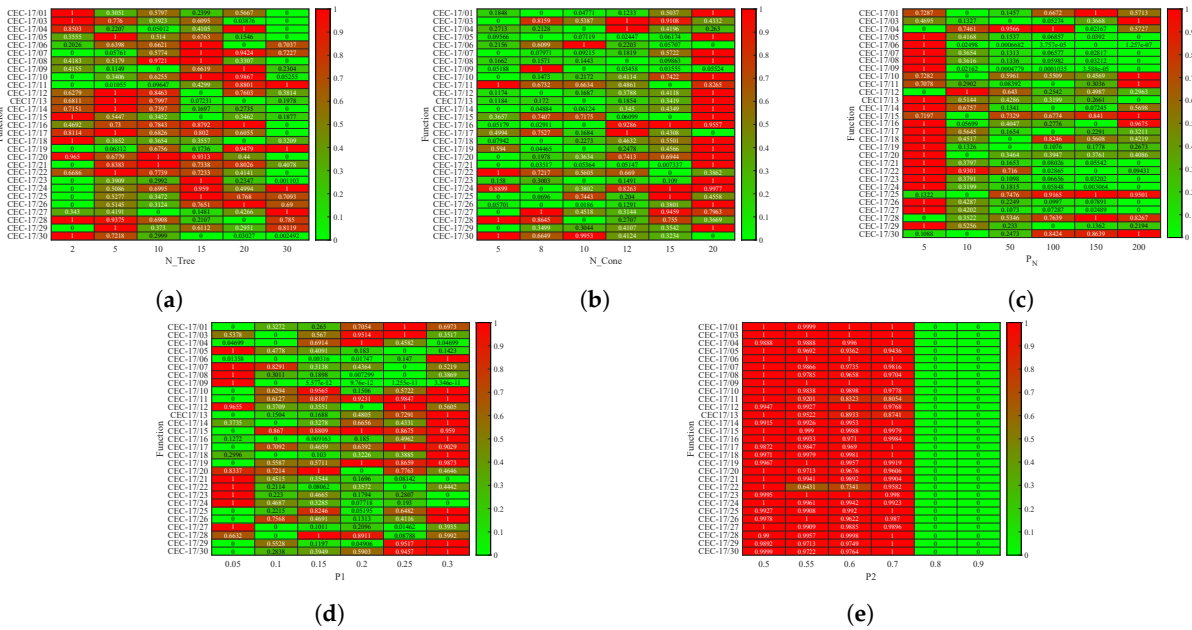


Figure A4. PCOA sensitivity analysis using CEC2017 benchmark functions on 100 dimensions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

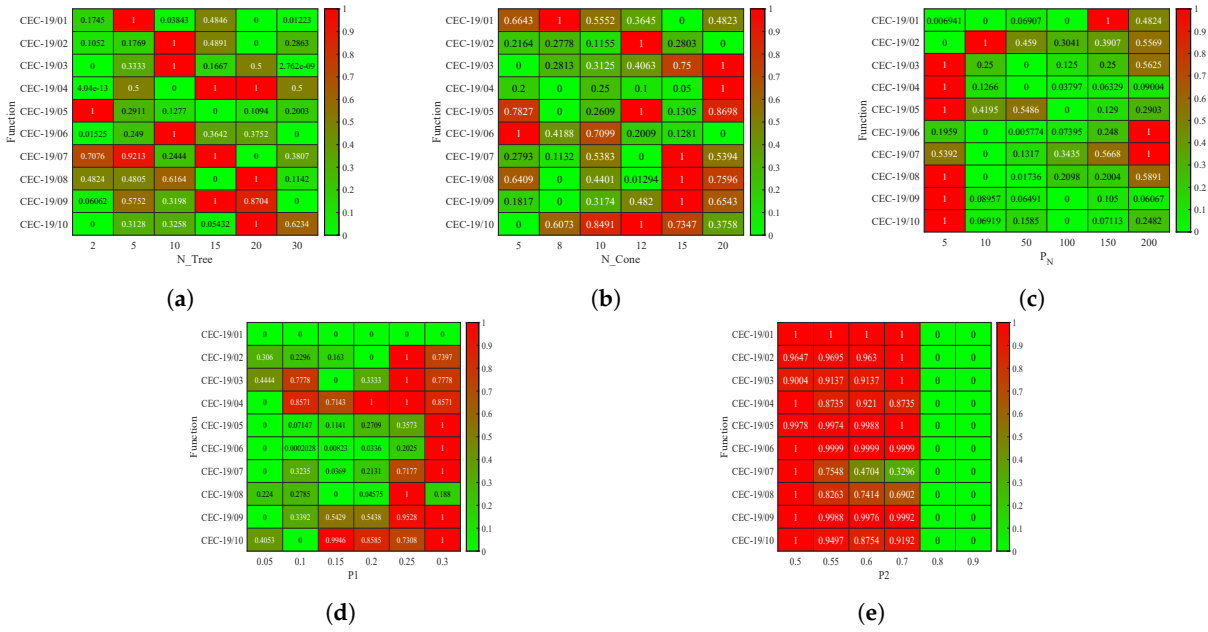


Figure A5. COA sensitivity analysis using CEC2019 benchmark functions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

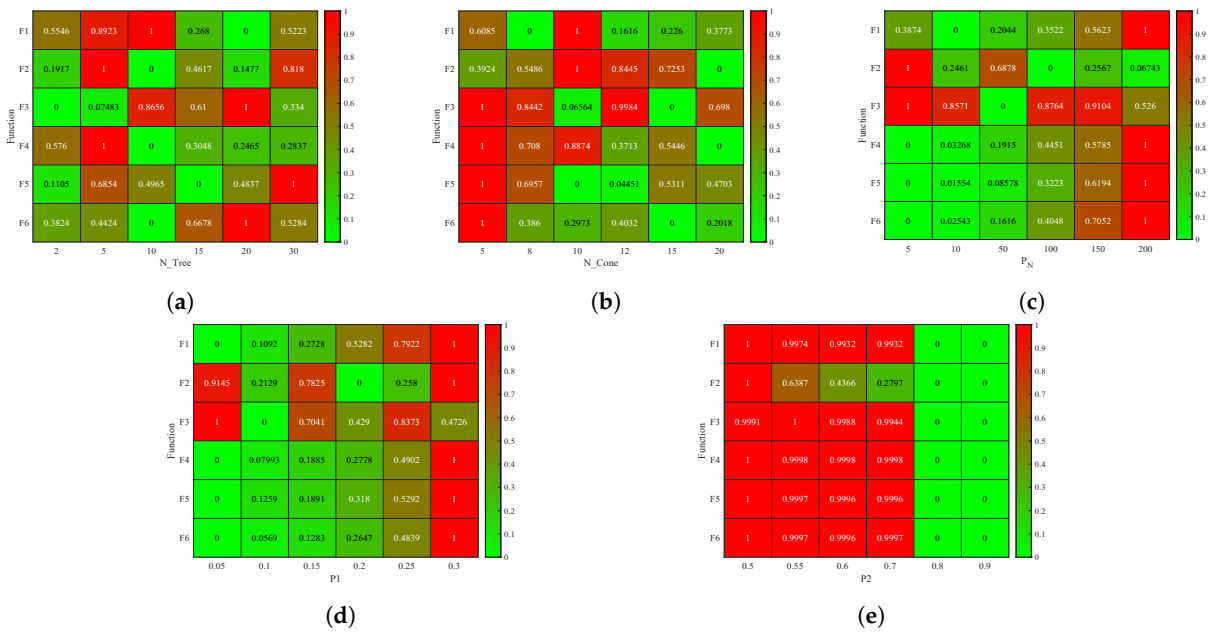


Figure A6. COA sensitivity analysis using CEC2006 benchmark functions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

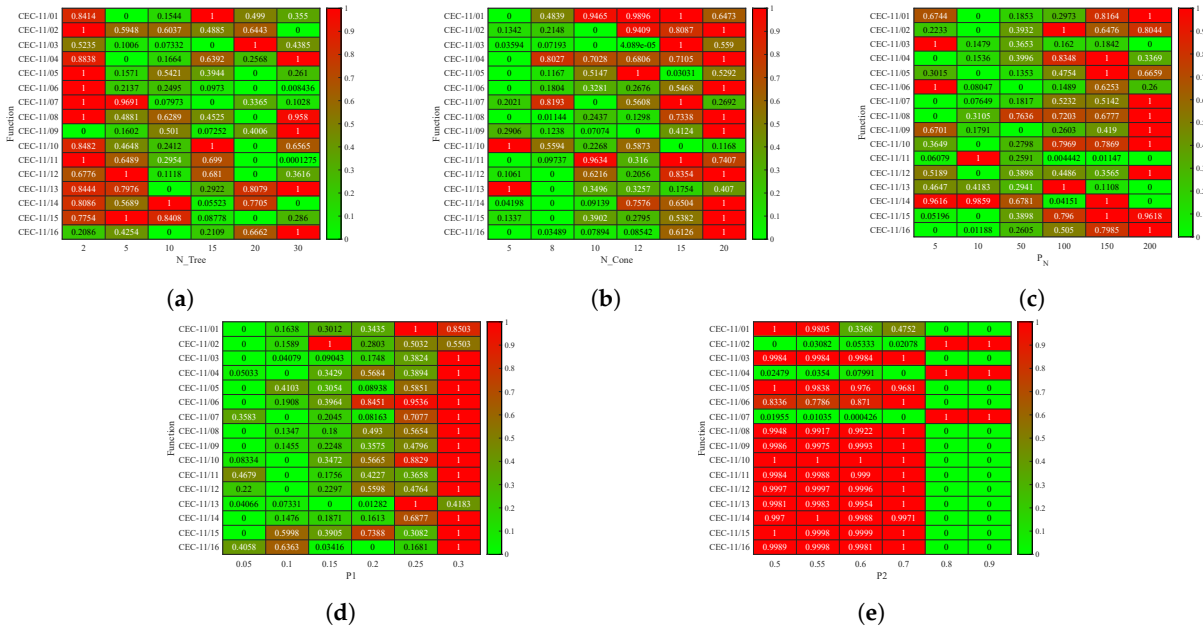


Figure A7. COA sensitivity analysis using CEC2011 benchmark functions. (a) N_Tree; (b) N_Cone; (c) N_Cycle; (d) P1; (e) P2.

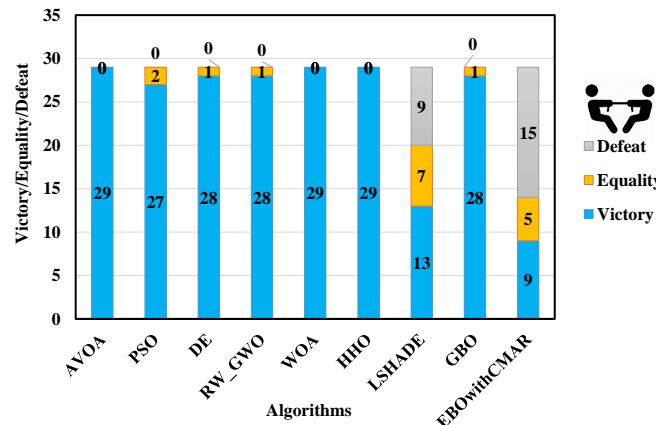


Figure A8. Comparison of Wilcoxon test for COA and other algorithms using the 30D CEC2017 benchmark functions over 51 independent runs and 300,000 NFES.

Table A3. Statistical criteria results of COA and other algorithms on classic benchmark functions, over 51 independent runs.

Function	PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR	
F1	Min	0.00	0.00	3.76×10^{-24}	4.92×10^{-12}	6.74×10^{-97}	1.04×10^{-216}	3.70×10^{-231}	1.11×10^{-306}	0.00	
	Mean	0.00	0.00	3.89×10^{-21}	8.46×10^{-12}	4.75×10^{-94}	5.56×10^{-199}	1.31×10^{-207}	1.25×10^{-73}	8.88×10^{-65}	
	Median	0.00	0.00	1.15×10^{-21}	7.94×10^{-12}	7.48×10^{-95}	1.17×10^{-206}	1.21×10^{-217}	3.86×10^{-74}	5.89×10^{-302}	
	Max	0.00	0.00	4.84×10^{-20}	1.37×10^{-11}	4.52×10^{-93}	2.62×10^{-197}	6.06×10^{-206}	2.78×10^{-72}	7.10×10^{-297}	3.65×10^{-63}
	Std	0.00	0.00	7.92×10^{-21}	2.21×10^{-12}	9.74×10^{-94}	0.00	0.00	3.96×10^{-73}	0.00	5.10×10^{-64}
F2	Min	0.00	0.00	1.66×10^{-15}	4.87×10^{-08}	3.34×10^{-55}	4.60×10^{-125}	8.52×10^{-125}	6.54×10^{-37}	6.57×10^{-156}	
	Mean	0.00	0.00	6.04×10^{-14}	6.92×10^{-08}	5.90×10^{-54}	3.74×10^{-115}	2.17×10^{-110}	1.82×10^{-35}	4.71×10^{-151}	
	Median	0.00	0.00	3.06×10^{-14}	6.74×10^{-08}	4.19×10^{-54}	1.08×10^{-118}	6.66×10^{-115}	1.21×10^{-35}	2.69×10^{-153}	
	Max	0.00	0.00	3.44×10^{-13}	9.95×10^{-08}	2.43×10^{-53}	1.40×10^{-113}	6.65×10^{-109}	8.67×10^{-35}	2.63×10^{-149}	4.33×10^{-30}
	Std	0.00	0.00	8.42×10^{-14}	1.19×10^{-08}	5.50×10^{-54}	2.02×10^{-114}	1.04×10^{-109}	1.80×10^{-35}	3.68×10^{-150}	6.11×10^{-31}
F3	Min	0.00	0.00	2.75×10^0	1.28×10^4	1.59×10^{-36}	4.47×10^1	7.80×10^{-209}	7.35×10^{-28}	4.39×10^{-255}	
	Mean	0.00	0.00	1.21×10^1	2.12×10^4	2.31×10^{-30}	2.50×10^3	3.91×10^{-189}	2.50×10^{-22}	1.49×10^{-15}	
	Median	0.00	0.00	1.11×10^1	2.19×10^4	5.91×10^{-33}	1.69×10^3	1.50×10^{-199}	7.59×10^{-24}	1.17×10^{-250}	
	Max	0.00	0.00	3.41×10^1	2.82×10^4	8.70×10^{-29}	8.39×10^3	1.16×10^{-187}	5.41×10^{-21}	7.30×10^{-246}	2.55×10^{-14}
	Std	0.00	0.00	7.30×10^0	3.73×10^3	1.22×10^{-29}	2.26×10^3	0.00	8.52×10^{-22}	0.00	4.29×10^{-15}
F4	Min	0.00	0.00	2.21×10^{-1}	1.55×10^0	3.78×10^{-26}	2.73×10^{-15}	2.84×10^{-118}	2.65×10^{-20}	1.85×10^{-142}	
	Mean	0.00	0.00	4.63×10^{-1}	2.02×10^0	3.28×10^{-24}	1.15×10^{-10}	3.24×10^{-104}	1.77×10^{-18}	1.55×10^{-11}	
	Median	0.00	0.00	4.30×10^{-1}	2.02×10^0	1.95×10^{-24}	1.96×10^{-1}	4.04×10^{-109}	6.34×10^{-19}	1.10×10^{-139}	
	Max	0.00	0.00	1.17×10^0	2.4×10^0	1.43×10^{-23}	8.26×10^1	1.39×10^{-102}	2.66×10^{-17}	3.49×10^{-138}	
	Std	0.00	0.00	2.11×10^{-1}	2.35×10^{-1}	3.34×10^{-24}	2.20×10^1	1.95×10^{-103}	3.84×10^{-18}	8.34×10^{-139}	

Table A3. Cont.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
F22	Min	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1
	Mean	-1.04×10^1	-1.04×10^1	-1.12×10^0	-1.04×10^1	-1.04×10^1	-1.00×10^1	-1.13×10^0	-1.04×10^1	-1.88×10^0	-1.04×10^1
	Median	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.09×10^0	-1.04×10^1	-1.04×10^1	-1.04×10^1
	Max	-1.04×10^1	-1.04×10^1	-1.75×10^0	-1.04×10^1	-1.04×10^1	-1.77×10^0	-1.09×10^0	-1.04×10^1	-1.09×10^0	-1.04×10^1
	Std	1.99×10^{-15}	1.44×10^{-15}	3.20×10^0	1.31×10^{-15}	7.67×10^{-5}	1.47×10^0	2.13×10^0	3.55×10^{-16}	1.60×10^0	1.79×10^{-15}
F23	Min	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1
	Mean	-1.05×10^1	-1.05×10^1	-1.10×10^0	-1.05×10^1	-1.03×10^1	-1.05×10^1	-1.87×10^0	-1.05×10^1	-1.69×10^0	-1.05×10^1
	Median	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.13×10^0	-1.05×10^1	-1.05×10^1	-1.05×10^1
	Max	-1.05×10^1	-1.05×10^1	-1.42×10^0	-1.05×10^1	-1.13×10^0	-1.05×10^1	-1.13×10^0	-1.05×10^1	-1.13×10^0	-1.05×10^1
	Std	1.79×10^{-15}	1.76×10^{-15}	3.41×10^0	1.79×10^{-15}	1.06×10^0	1.80×10^{-3}	1.88×10^0	2.04×10^{-15}	1.99×10^0	2.04×10^{-15}

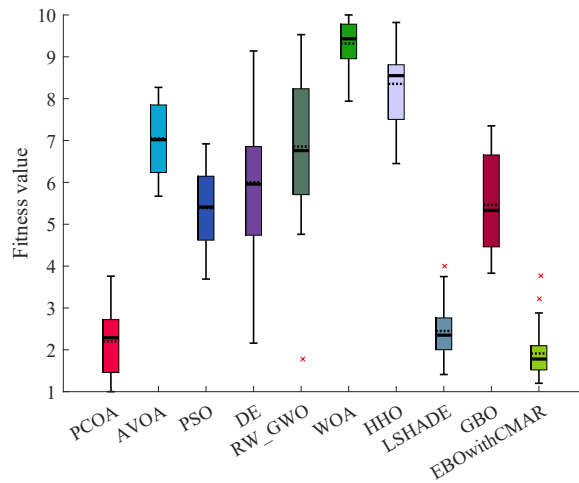


Figure A9. Comparison of Friedman ranking for PCOA and other algorithms using the 30D CEC2017 benchmark functions over 51 independent runs and 300,000 NFES.

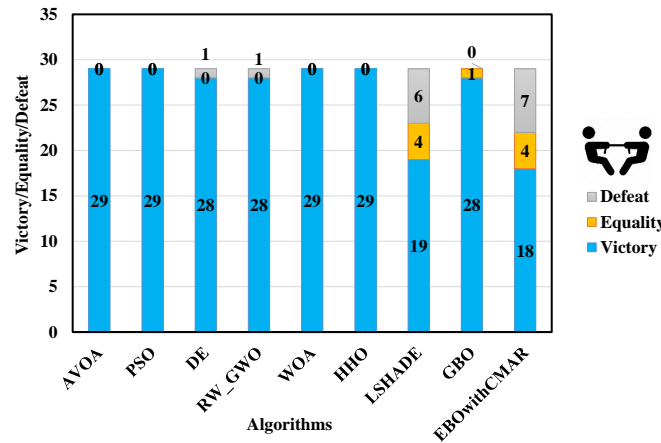


Figure A10. Comparison of Wilcoxon test for PCOA and other algorithms using the 50D CEC2017 benchmark functions over 51 independent runs and 500,000 NFES.

Table A4. Statistical criteria results of PCOA and other algorithms on the 10D CEC2017, over 51 independent runs.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC_17/01	Min	100.00	102.70	100.96	135.13	2439.47	7915.27	36460.41	100.00	100.10	100.00
	Mean	100.00	4312.89	1723.10	1371.99	292277.87	295915.24	233033.39	100.00	1004.74	100.00
	Median	100.00	3954.98	785.93	1230.12	11000.50	70542.61	218064.13	100.00	476.65	100.00
	Max	100.00	12739.83	12252.30	4280.86	7276035.49	2676940.69	577133.76	100.00	6084.59	100.00
	Std	0.00	3600.65	2329.07	1010.83	1269917.69	579476.71	117529.75	0.00	1235.43	0.00
CEC_17/03	Min	300.00	300.00	300.00	879.94	300.06	310.19	300.20	300.00	300.00	300.00
	Mean	300.00	300.00	300.00	1936.66	304.15	511.89	300.78	300.00	300.00	300.00
	Median	300.00	300.00	300.00	1922.24	302.75	468.70	300.73	300.00	300.00	300.00
	Max	300.00	300.00	300.00	3465.34	317.71	1241.93	301.66	300.00	300.00	300.00
	Std	0.00	0.00	0.00	564.41	4.39	187.70	0.35	0.00	0.00	0.00

Table A4. Cont.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC_17/22	Min	2200.00	2239.42	2211.20	2258.63	2301.00	2247.75	2231.19	2200.00	2219.29	2300.00
	Mean	2250.02	2307.06	2298.89	2296.80	2304.74	2313.36	2308.94	2298.07	2299.93	2300.00
	Median	2232.00	2307.09	2301.92	2300.57	2305.89	2313.40	2313.52	2300.00	2302.45	2300.00
	Max	2300.35	2321.07	2303.37	2301.42	2309.04	2336.06	2330.27	2300.35	2309.11	2300.00
CEC_17/23	Std	39.37	10.83	15.49	9.53	2.91	11.94	20.33	14.01	14.93	0.00
	Min	2300.00	2609.40	2605.14	2607.02	2606.54	2615.62	2610.56	2602.86	2604.32	2600.00
	Mean	2599.22	2629.83	2624.95	2611.41	2619.61	2644.45	2657.81	2604.60	2624.02	2600.44
	Median	2605.91	2627.48	2624.94	2611.60	2619.57	2642.83	2658.93	2604.60	2621.87	2600.00
CEC_17/24	Max	2613.12	2663.67	2665.34	2615.19	2632.72	2695.53	2695.42	2607.27	2656.68	2604.11
	Std	42.83	11.90	12.33	1.86	5.90	16.34	20.39	1.02	10.90	1.13
	Min	2500.00	2429.87	2500.00	2620.00	2500.36	2501.92	2500.56	2500.00	2500.00	2500.00
	Mean	2545.93	2726.69	2726.34	2717.77	2717.77	2742.39	2753.22	2717.68	2689.53	2554.74
CEC_17/25	Median	2500.00	2763.09	2748.54	2743.72	2747.15	2777.74	2793.75	2732.03	2745.95	2500.00
	Max	2734.93	2823.81	2800.40	2750.26	2764.90	2856.97	2882.39	2735.65	2775.27	2729.54
	Std	68.36	104.54	79.72	40.92	34.99	85.54	94.40	50.33	112.35	91.60
	Min	2600.03	2600.19	2897.74	2899.84	2897.79	2606.80	2609.28	2897.74	2897.74	2897.74
CEC_17/26	Mean	2896.52	2929.07	2923.15	2911.91	2924.10	2924.10	2926.75	2915.07	2925.92	2919.48
	Median	2897.75	2946.50	2943.47	2909.79	2943.42	2951.52	2945.58	2898.03	2944.18	2899.58
	Max	2943.44	3024.18	2949.15	2944.47	2948.56	3052.13	2972.85	2945.80	2971.76	2943.37
	Std	44.49	53.74	23.30	10.71	23.50	66.69	51.10	22.42	25.43	22.73
CEC_17/27	Min	2600.00	2600.00	2600.00	2715.72	2600.57	2601.90	2811.59	2900.00	2600.00	2800.00
	Mean	2833.28	3019.96	2885.47	2896.64	3004.37	3332.97	3316.84	2900.00	2953.87	2884.61
	Median	2884.67	3025.99	2900.00	2918.33	2900.17	3137.58	3206.68	2900.00	2900.00	2900.00
	Max	2900.00	3462.30	3044.26	2952.48	3997.45	4801.41	4552.35	2900.00	3857.35	2900.00
CEC_17/28	Std	86.07	153.11	81.83	54.60	305.99	487.08	426.84	0.00	173.27	36.08
	Min	3086.89	3089.52	3089.01	3089.05	3073.32	3092.24	3097.14	3089.01	3089.64	3089.52
	Mean	3088.66	3099.77	3106.63	3089.69	3079.47	3122.82	3131.65	3089.45	3103.45	3091.80
	Median	3089.01	3097.51	3097.80	3089.66	3078.33	3106.23	3120.99	3089.52	3095.74	3090.75
CEC_17/29	Max	3089.71	3131.03	3208.63	3090.47	3094.43	3221.35	3219.83	3089.52	3193.05	3095.98
	Std	0.82	8.74	27.44	0.25	4.22	35.35	30.88	0.18	22.92	2.38
	Min	2800.00	3100.00	3100.00	3172.85	3100.18	3100.18	3101.60	3100.00	3100.00	3100.00
	Mean	3200.01	3310.54	3199.33	3236.69	3235.88	3403.55	3337.01	3214.03	3330.47	3100.00
CEC_17/30	Median	3100.00	3383.75	3100.00	3210.61	3222.55	3411.82	3383.87	3100.00	3402.69	3100.00
	Max	3411.82	3411.82	3444.13	3411.82	3300.00	3749.37	3501.44	3411.82	3411.82	3100.00
	Std	158.46	124.69	135.72	66.54	64.23	185.04	115.05	145.87	120.05	0.00
	Min	3129.61	3145.87	3148.21	3145.35	3137.22	3156.68	3160.31	3128.05	3145.56	3128.28
CEC_17/29	Mean	3140.45	3233.85	3212.09	3161.97	3175.42	3308.50	3300.66	3138.34	3234.63	3132.96
	Median	3137.70	3226.35	3199.64	3161.65	3168.03	3300.99	3294.46	3138.01	3224.14	3132.98
	Max	3174.45	3412.72	3331.33	3178.83	3300.25	3473.89	3479.31	3154.30	3359.96	3139.27
	Std	9.88	58.82	45.48	7.60	30.91	77.33	75.93	5.89	55.10	2.43
CEC_17/30	Min	3408.33	4296.06	3762.14	4657.82	4676.02	6296.85	4838.12	3394.50	3416.36	3394.50
	Mean	3451.00	150294.12	167688.20	17934.90	24185.38	564983.34	527231.40	99555.88	395498.50	3409.27
	Median	3454.01	16180.83	6759.34	13574.83	15177.31	220962.22	121932.13	3442.66	3729.03	3407.44
	Max	3503.40	1237436.14	1261343.43	67896.82	168236.16	2136927.89	4219938.92	820578.06	1827680.48	3600.49
	Std	23.51	345469.25	359033.99	12874.06	27828.94	630800.00	851685.44	265899.84	540967.78	31.24

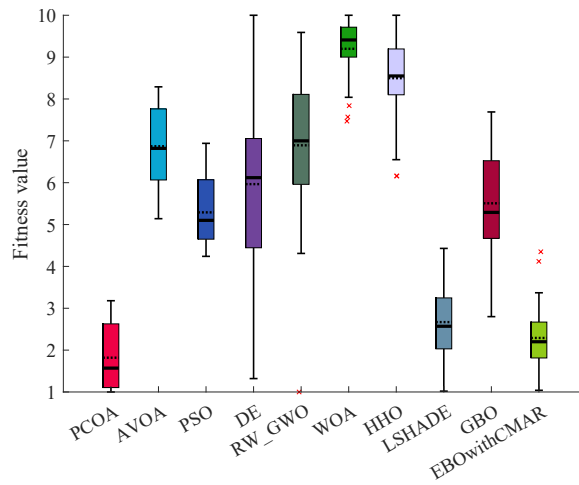


Figure A11. Comparison of Friedman ranking for PCOA and other algorithms using the 50D CEC2017 benchmark functions over 51 independent runs and 500,000 NFES.

Table A5. Cont.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC_17/20	Min	2027.65	2114.61	2207.68	2049.38	2131.39	2342.07	2263.56	2030.19	2123.39	2017.40
	Mean	2090.19	2545.71	2463.99	2147.24	2370.87	2697.81	2664.92	2078.53	2431.03	2062.60
	Median	2060.46	2542.39	2458.83	2128.83	2344.58	2694.95	2671.11	2048.39	2437.53	2041.63
	Max	2217.96	2923.51	3070.79	2268.86	2730.48	3198.92	3144.70	2175.35	2818.73	2163.91
	Std	59.01	183.45	184.22	55.23	137.49	201.51	206.20	54.18	178.13	45.71
CEC_17/21	Min	2200.00	2392.67	2345.60	2404.06	2342.50	2468.86	2398.43	2310.30	2356.23	2302.99
	Mean	2308.83	2480.64	2406.29	2432.32	2441.72	2559.22	2522.68	2316.63	2415.63	2312.41
	Median	2309.60	2486.59	2398.34	2434.47	2443.02	2548.67	2519.62	2316.61	2417.02	2313.57
	Max	2324.63	2594.28	2488.09	2449.46	2481.53	2686.51	2667.02	2322.08	2495.22	2320.12
	Std	16.27	50.71	34.36	9.60	22.67	57.94	51.96	2.64	33.89	4.77
CEC_17/22	Min	2300.00	2300.00	2300.00	2559.60	2310.52	2314.49	2316.49	2300.00	2300.00	2300.00
	Mean	2300.00	5217.29	3636.20	3254.76	6084.43	6341.02	4902.78	2300.00	3254.30	2300.00
	Median	2300.00	6308.16	2302.45	3225.11	7258.64	7314.27	6042.55	2300.00	2302.46	2300.00
	Max	2300.00	8175.04	7606.95	4246.95	8508.85	8811.62	7858.86	2300.00	7977.74	2300.00
	Std	0.00	2129.19	2039.38	333.78	2506.43	2318.24	2321.01	0.00	1759.59	0.00
CEC_17/23	Min	2644.69	2788.25	2715.95	2747.63	2704.46	2847.34	2914.39	2650.05	2713.95	2652.03
	Mean	2660.84	2897.02	2826.55	2779.46	2811.14	3055.07	3048.60	2665.11	2788.20	2660.83
	Median	2658.62	2898.06	2827.39	2780.46	2811.92	3062.16	3032.12	2665.80	2789.27	2660.63
	Max	2679.33	3024.60	2983.14	2795.50	2863.25	3334.17	3247.26	2674.96	2914.16	2671.19
	Std	9.14	63.75	47.59	9.37	24.68	102.82	78.36	5.02	43.80	4.34
CEC_17/24	Min	2826.46	2951.23	2895.97	2955.73	2924.29	2972.52	3041.84	2828.20	2880.57	2600.00
	Mean	2836.71	3114.91	2990.45	2983.34	2968.64	3155.93	3345.31	2835.97	2949.61	2826.84
	Median	2832.99	3111.11	2985.80	2984.03	2968.28	3157.83	3364.12	2835.73	2943.68	2831.60
	Max	2853.29	3326.07	3134.56	3002.68	3018.77	3354.54	3641.83	2845.93	3084.74	2841.04
	Std	7.53	83.47	56.86	10.30	17.92	94.77	131.14	3.33	44.55	32.59
CEC_17/25	Min	2883.39	2883.83	2883.48	2887.08	2884.17	2889.25	2884.33	2886.74	2883.72	2883.61
	Mean	2883.84	2900.64	2891.52	2887.33	2903.82	2938.48	2907.97	2886.84	2896.51	2886.65
	Median	2883.44	2890.40	2887.94	2887.36	2904.20	2944.49	2908.69	2886.83	2889.71	2886.79
	Max	2887.12	2945.48	2941.42	2887.62	2940.71	2978.91	2950.68	2887.06	2940.78	2888.37
	Std	1.12	17.75	10.54	0.10	10.74	25.17	17.57	0.07	15.18	0.79
CEC_17/26	Min	2800.00	2900.00	2800.00	4746.81	2901.70	3815.79	2864.07	3595.14	2800.00	2800.00
	Mean	3351.93	6193.00	4548.70	4931.23	5181.01	7397.92	6699.50	3691.92	4917.60	3544.95
	Median	3590.58	6421.87	4980.99	4936.54	5244.56	7545.58	6956.79	3688.00	5152.93	3589.33
	Max	3996.54	8311.80	6911.57	5071.31	5868.04	9707.03	8755.10	3796.25	7554.15	3760.51
	Std	460.33	1185.95	1350.94	66.71	443.32	1282.68	1294.23	48.29	1152.94	221.23
CEC_17/27	Min	3180.61	3209.05	3204.78	3201.77	3161.41	3251.25	3221.17	3181.59	3208.95	3182.53
	Mean	3200.89	3271.09	3249.76	3209.04	3198.05	3367.50	3312.66	3205.39	3248.64	3205.07
	Median	3201.75	3259.92	3244.79	3209.00	3200.01	3342.96	3299.11	3206.41	3243.92	3205.45
	Max	3230.25	3416.33	3330.63	3213.64	3200.01	3696.48	3522.39	3224.62	3342.70	3226.91
	Std	9.43	39.14	28.08	2.47	8.11	90.82	64.17	8.24	30.46	6.98
CEC_17/28	Min	3100.00	3100.00	3100.00	3208.26	3289.66	3236.60	3111.29	3100.00	3100.00	3100.00
	Mean	3100.00	3206.95	3141.43	3224.35	3299.80	3302.67	3241.66	3139.95	3133.20	3111.33
	Median	3100.00	3209.17	3100.00	3221.75	3300.01	3298.13	3251.39	3100.00	3100.00	3100.00
	Max	3100.00	3266.64	3260.11	3265.69	3300.01	3498.62	3285.49	3266.64	3261.85	3253.93
	Std	0.00	42.59	57.89	11.09	1.45	44.02	29.58	55.62	55.95	35.25
CEC_17/29	Min	3281.91	3443.19	3371.87	3473.61	3377.66	3895.34	3526.75	3288.74	3510.59	3278.90
	Mean	3451.27	4013.32	3779.36	3638.74	3710.66	4745.22	4261.91	3358.02	3948.38	3338.64
	Median	3428.50	4071.52	3745.07	3642.60	3729.95	4721.59	4274.94	3359.23	3908.64	3335.43
	Max	3661.95	4603.29	4388.51	3799.41	4104.16	5734.41	4883.95	3453.61	4438.80	3444.73
	Std	81.12	270.07	227.78	78.59	192.77	358.21	280.45	24.36	232.72	23.70
CEC_17/30	Min	5159.96	10476.10	5310.53	10044.83	177700.27	667293.22	162969.89	4941.81	5069.54	4984.43
	Mean	5543.69	28572.87	8248.09	24304.51	919685.11	9513847.27	885053.81	5088.99	7536.31	5131.40
	Median	5536.86	27824.48	6880.08	23438.21	684404.15	8068347.15	858471.02	5052.27	6953.51	5078.41
	Max	5845.63	57458.41	16725.47	42467.45	3166194.32	30866546.23	2056085.10	6146.01	14109.65	5879.85
	Std	137.41	11102.58	2882.37	6656.45	658333.94	6809939.36	387608.27	181.57	2052.50	157.63

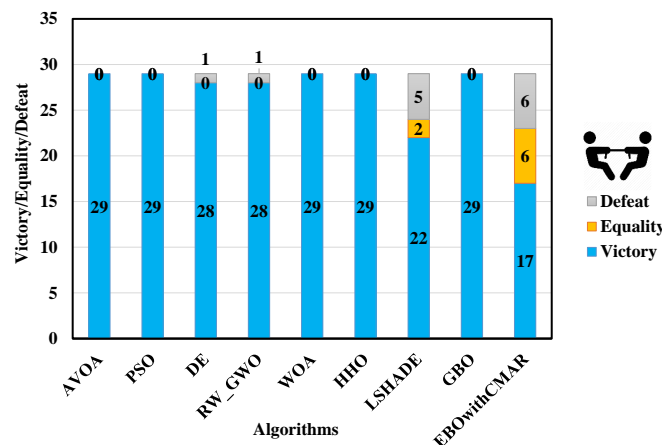


Figure A12. Comparison of Wilcoxon test for PCOA and other algorithms using the 100D CEC2017 benchmark functions over 51 independent runs and 1,000,000 NFES.

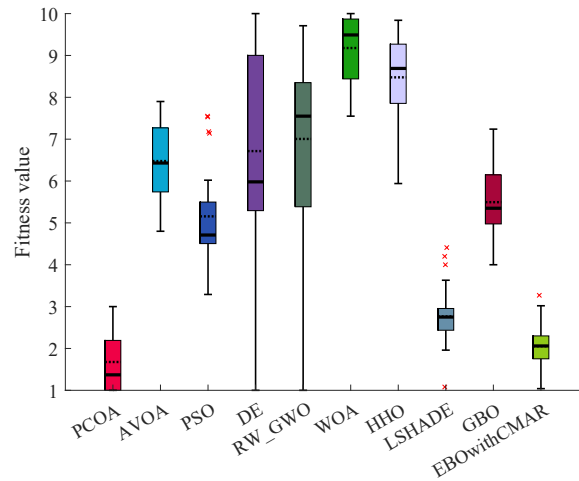


Figure A13. Comparison of Friedman ranking for PCOA and other algorithms using the 100D CEC2017 benchmark functions over 51 independent runs and 1,000,000 NFES.

Table A9. Statistical results of PCOA and other algorithms on the CEC2006, over 51 independent runs.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC06_1	Function	PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	L_SHADE	GBO	EBO_BIN
	Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	Mean	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	Median	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	Max	0.01	0.02	0.02	0.01	0.01	0.02	0.02	0.01	0.01	0.01
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CEC06_2	Min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Mean	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Max	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Std	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CEC06_3	Min	2522.48	2749.58	2749.58	2749.58	2751.00	2749.58	2749.63	2749.58	2749.58	2749.58
	Mean	2712.28	2750.28	2749.58	2749.58	2759.26	2886.49	2759.82	2749.58	2749.58	2749.58
	Median	2750.11	2749.83	2749.58	2749.58	2759.27	2776.88	2756.18	2749.58	2749.58	2749.58
	Max	2753.66	2754.02	2749.58	2749.58	2770.05	4205.24	2808.03	2749.58	2749.58	2749.58
CEC06_4	Std	64.90	0.99	0.00	0.00	4.29	284.02	11.74	0.00	0.00	0.00
	Min	464.66	464.80	464.68	466.05	464.92	478.80	468.23	464.65	464.68	464.65
	Mean	464.69	469.10	464.87	471.86	467.41	552.26	479.41	464.66	465.03	464.66
	Median	464.68	467.07	464.84	471.34	466.82	550.98	476.34	464.66	464.91	464.66
	Max	464.79	498.31	465.31	479.73	474.27	650.83	528.17	464.69	467.26	464.69
	Std	0.03	5.45	0.14	3.05	2.30	44.24	10.29	0.01	0.44	0.01
CEC06_5	Min	384.78	384.88	384.80	385.22	385.06	390.25	386.55	384.78	384.78	384.78
	Mean	384.82	385.72	385.00	386.09	386.34	471.09	397.62	384.79	384.99	384.78
	Median	384.81	385.59	384.94	386.05	386.40	465.47	396.08	384.79	384.91	384.78
	Max	384.93	387.54	385.82	387.54	388.30	624.02	418.70	384.81	385.44	384.81
CEC06_6	Std	0.04	0.58	0.20	0.48	0.73	60.32	7.57	0.01	0.18	0.01
	Min	455.35	455.50	455.37	457.97	455.80	461.81	462.46	455.35	455.36	455.35
	Mean	455.37	459.94	455.55	460.96	457.79	552.40	479.45	455.37	455.64	455.35
	Max	455.37	457.95	455.55	460.81	457.55	545.45	475.34	455.37	455.51	455.35
	Std	0.02	8.38	0.12	1.92	1.18	49.88	13.87	0.01	0.40	0.00

Table A10. Statistical results of PCOA and other algorithms on the CEC2011, over 25 independent runs.

Function		PCOA	AVOA	PSO	DE	RW_GWO	WOA	HHO	LSHADE	GBO	EBOwithCMAR
CEC11_01	Min	0.00	0.00	6.53	0.00	0.02	11.49	2.06	0.00	0.00	0.00
	Mean	1.41	15.73	16.48	4.60	13.46	19.05	19.41	0.02	14.66	0.00
	Median	0.01	16.66	14.93	3.80	12.32	19.85	21.42	0.01	14.91	0.00
	Max	9.79	22.75	24.14	12.84	21.97	24.03	25.10	0.09	24.75	0.00
	Std	3.00	5.15	5.28	4.09	5.72	3.77	4.89	0.03	6.71	0.00
CEC11_02	Min	-18.42	-18.42	-17.52	-13.65	-17.40	-16.31	-12.27	-15.25	-18.42	-12.43
	Mean	-16.66	-15.89	-13.90	-11.86	-14.58	-19.28	-14.95	-13.72	-16.70	-18.98
	Median	-17.11	-16.52	-15.33	-11.76	-15.36	-19.52	-15.17	-13.74	-16.52	-17.55
	Max	-18.86	-17.06	-11.11	-10.46	-10.27	-12.11	-1.73	-11.22	-15.34	-16.46
CEC11_03	Std	1.77	2.37	4.34	0.69	1.83	3.92	2.82	1.11	0.81	3.41
	Min	13.77	13.77	13.77	13.77	22.00	13.77	13.77	13.77	13.77	13.77
	Mean	13.77	13.93	15.74	19.04	22.00	13.86	14.06	14.08	13.96	14.56
	Median	13.77	13.77	14.33	20.82	22.00	13.77	13.98	14.33	13.85	14.33
	Max	13.77	14.33	20.96	21.54	22.00	14.33	14.33	14.33	14.33	20.96
	Std	0.00	0.26	3.00	2.92	0.00	0.21	0.27	0.28	0.24	1.39
CEC11_04	Min	-19.17	-19.17	-11.07	-11.84	-19.03	-19.16	-16.33	-19.16	-12.71	-139.58
	Mean	-15.72	-12.77	68.96	639993.58	-13.22	-19.58	-17.44	-19.15	599992.22	119927.45
	Median	-16.50	-13.01	28.67	999999.00	-13.01	-10.34	-16.84	-19.15	999999.00	-17.89
	Max	-11.27	-15.09	259.18	999999.00	-19.51	-12.42	-12.37	-19.09	999999.00	999999.00
	Std	2.91	3.64	89.24	489905.33	2.39	3.94	3.20	0.02	500008.48	331689.44

9. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
10. Karaboga, D.; Basturk, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In Proceedings of the 12th International Fuzzy Systems Association World Congress, IFSA 2007, Cancun, Mexico, 18–21 June 2007; pp. 789–798.
11. Yang, X.S. Firefly algorithm. In *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Frome, UK, 2008; pp. 79–90.
12. Yang, X.; Hossein Gandomi, A. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [[CrossRef](#)]
13. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [[CrossRef](#)]
14. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
15. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
16. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
17. Dehghani, M.; Montazeri, Z.; Trojovská, E.; Trojovský, P. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **2023**, *259*, 110011. [[CrossRef](#)]
18. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [[CrossRef](#)]
19. Dehghani, M.; Montazeri, Z.; Bektemyssova, G.; Malik, O.P.; Dhiman, G.; Ahmed, A.E.M. Kookaburra Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Biomimetics* **2023**, *8*, 470. [[CrossRef](#)]
20. Alsayyed, O.; Hamadneh, T.; Al-Tarawneh, H.; Alqudah, M.; Gochhait, S.; Leonova, I.; Malik, O.P.; Dehghani, M. Giant Armadillo Optimization: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Biomimetics* **2023**, *8*, 619. [[CrossRef](#)]
21. Dehghani, M.; Bektemyssova, G.; Montazeri, Z.; Shaikemelev, G.; Malik, O.P.; Dhiman, G. Lyrebird Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Biomimetics* **2023**, *8*, 507. [[CrossRef](#)]
22. Anaraki, M.V.; Farzin, S. Humboldt Squid Optimization Algorithm (HSOA): A Novel Nature-Inspired Technique for Solving Optimization Problems. *IEEE Access* **2023**, *11*, 122069–122115. [[CrossRef](#)]
23. Formato, R.A. Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 221–238. [[CrossRef](#)]
24. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [[CrossRef](#)]
25. Alatas, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization. *Expert Syst. Appl.* **2011**, *38*, 13170–13180. [[CrossRef](#)]
26. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
27. Karami, H.; Anaraki, M.V.; Farzin, S.; Mirjalili, S. Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems. *Comput. Ind. Eng.* **2021**, *156*, 107224. [[CrossRef](#)]
28. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [[CrossRef](#)]
29. Glover, F. Tabu Search—Part I. *ORSA J. Comput.* **1989**, *1*, 190–206. [[CrossRef](#)]
30. Samareh Moosavi, S.H.; Bardsiri, V.K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **2019**, *86*, 165–181. [[CrossRef](#)]
31. Al-Betar, M.A.; Alyasseri, Z.A.A.; Awadallah, M.A.; Doush, I.A. Coronavirus herd immunity optimizer (CHIO). *Neural Comput. Appl.* **2020**, *33*, 5011–5042. [[CrossRef](#)]
32. Mirrashid, M.; Naderpour, H. Incomprehensible but Intelligible-in-time logics: Theory and optimization algorithm. *Knowl.-Based Syst.* **2023**, *110305*. [[CrossRef](#)]
33. Hubalovska, M.; Major, S. A New Human-Based Metaheuristic Algorithm for Solving Optimization Problems Based on Technical and Vocational Education and Training. *Biomimetics* **2023**, *8*, 508. [[CrossRef](#)]
34. Zhu, D.; Wang, S.; Zhou, C.; Yan, S.; Xue, J. Human memory optimization algorithm: A memory-inspired optimizer for global optimization problems. *Expert Syst. Appl.* **2024**, *237*, 121597. [[CrossRef](#)]
35. Kaveh, A. Tug of War Optimization. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures*; Springer International Publishing: Cham, Switzerland, 2017; pp. 451–487. [[CrossRef](#)]
36. Dehghani, M.; Mardaneh, M.; Guerrero, J.; Malik, O.; Kumar, V. Football Game Based Optimization: An Application to Solve Energy Commitment Problem. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 514–523. [[CrossRef](#)]
37. Barik, S.; Das, D. Athletic Run Based Optimization: A novel method for the integration of DGs and shunt capacitors. In Proceedings of the 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 9–11 July 2021; pp. 1–6. [[CrossRef](#)]

38. Połap, D.; Woz´niak, M. Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry* **2017**, *9*, 203. [[CrossRef](#)]
39. Sun, W.z.; Wang, J.s.; Wei, X. An Improved Whale Optimization Algorithm Based on Different Searching Paths and Perceptual Disturbance. *Symmetry* **2018**, *10*, 210. [[CrossRef](#)]
40. Li, G.; Shuang, F.; Zhao, P.; Le, C. An Improved Butterfly Optimization Algorithm for Engineering Design Problems Using the Cross-Entropy Method. *Symmetry* **2019**, *11*, 1049. [[CrossRef](#)]
41. Ma, Z.; Yuan, X.; Han, S.; Sun, D.; Ma, Y. Improved Chaotic Particle Swarm Optimization Algorithm with More Symmetric Distribution for Numerical Function Optimization. *Symmetry* **2019**, *11*, 876. [[CrossRef](#)]
42. Mehrabian, A.; Lucas, C. A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inform.* **2006**, *1*, 355–366. [[CrossRef](#)]
43. Yang, X.S. Flower pollination algorithm for global optimization. In Proceedings of the International Conference on Unconventional Computing and Natural Computation, Orléans, France, 3–7 September 2012; pp. 240–249.
44. Kiran, M.S. TSA: Tree-seed algorithm for continuous optimization. *Expert Syst. Appl.* **2015**, *42*, 6686–6698. [[CrossRef](#)]
45. Cheraghalipour, A.; Hajiaghahi-Keshteli, M.; Paydar, M.M. Tree Growth Algorithm (TGA): A novel approach for solving optimization problems. *Eng. Appl. Artif. Intell.* **2018**, *72*, 393–414. [[CrossRef](#)]
46. Alimoradi, M.; Azgomi, H.; Asghari, A. Trees Social Relations Optimization Algorithm: A new Swarm-Based metaheuristic technique to solve continuous and discrete optimization problems. *Math. Comput. Simul.* **2022**, *194*, 629–664. [[CrossRef](#)]
47. Kaveh, M.; Mesgari, M.S.; Saeidian, B. Orchard Algorithm (OA): A new meta-heuristic algorithm for solving discrete and continuous optimization problems. *Math. Comput. Simul.* **2023**, *208*, 95–135. [[CrossRef](#)]
48. Mahmoodabadi, M.J.; Rasekh, M.; Yahyapour, M. Tree optimization algorithm (TOA): A novel metaheuristic approach for solving mathematical test functions and engineering problems. *Evol. Intell.* **2023**, *16*, 1325–1338. [[CrossRef](#)]
49. Bello, A.; Manyala, N.; Barzegar, F.; Khaleed, A.A.; Momodu, D.Y.; Dangbegnon, J.K. Renewable pine cone biomass derived carbon materials for supercapacitor application. *RSC Adv.* **2016**, *6*, 1800–1809. [[CrossRef](#)]
50. Culley, T.M.; Weller, S.G.; Sakai, A.K. The evolution of wind pollination in angiosperms. *Trends Ecol. Evol.* **2002**, *17*, 361–369. [[CrossRef](#)]
51. Lu, Y.; Jin, B.; Wang, L.; Wang, Y.; Wang, D.; Jiang, X.X.; Chen, P. Adaptation of male reproductive structures to wind pollination in gymnosperms: Cones and pollen grains. *Can. J. Plant Sci.* **2011**, *91*, 897–906. [[CrossRef](#)]
52. Burczyk, J.; Adams, W.T.; Shimizu, J.Y. Mating patterns and pollen dispersal in a natural knobcone pine (*Pinus attenuate* Lemmon.) stand. *Heredity* **1996**, *77*, 251–260. [[CrossRef](#)]
53. Offord, C.A.; Porter, C.L.; Meagher, P.F.; Errington, G. Sexual reproduction and early plant growth of the Wollemi pine (*Wollemia nobilis*), a rare and threatened Australian conifer. *Ann. Bot.* **1999**, *84*, 1–9. [[CrossRef](#)]
54. Benkman, C.W. Wind dispersal capacity of pine seeds and the evolution of different seed dispersal modes in pines. *Oikos* **1995**, *73*, 221–224. [[CrossRef](#)]
55. Baker, R. Mathematics in the Garden: Arranging Sweetcorn Plants for Maximum Pollination. *Math. Today* **2020**, *56*, 107–109.
56. Moller, H. Foods and foraging behaviour of Red (*Sciurus vulgaris*) and Grey (*Sciurus carolinensis*) squirrels. *Mammal Rev.* **1983**, *13*, 81–98. [[CrossRef](#)]
57. Sun, P.; Liu, H.; Zhang, Y.; Meng, Q.; Tu, L.; Zhao, J. An improved atom search optimization with dynamic opposite learning and heterogeneous comprehensive learning. *Appl. Soft Comput.* **2021**, *103*, 107140. [[CrossRef](#)]
58. Awad, N.H.; Ali, M.Z.; Liang, J.J.; Qu, B.Y.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization*; Technical Report; Nanyang Technological University: Singapore, 2016.
59. Das, S.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*; Jadavpur University: Kolkata, India; Nanyang Technological University: Singapore, 2010; pp. 341–359.
60. Awad, N.H.; Ali, M.Z.; Suganthan, P.N. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 372–379.
61. Nematollahi, A.F.; Rahiminejad, A.; Vahidi, B. A novel meta-heuristic optimization method based on golden ratio in nature. *Soft Comput.* **2020**, *24*, 1117–1151. [[CrossRef](#)]
62. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [[CrossRef](#)]
63. Kumar, A.; Misra, R.K.; Singh, D. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Donostia, Spain, 5–8 June 2017; pp. 1835–1842.
64. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.
65. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [[CrossRef](#)]

-
66. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **2021**, *181*, 115079. [[CrossRef](#)]
 67. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.