



Article

A Random Particle Swarm Optimization Based on Cosine Similarity for Global Optimization and Classification Problems

Yujia Liu ¹, Yuan Zeng ¹, Rui Li ², Xingyun Zhu ², Yuemai Zhang ², Weijie Li ², Taiyong Li ³ , Donglin Zhu ^{2,*} and Gangqiang Hu ^{2,*}

¹ School of Intelligent Manufacturing Engineering, Jiangxi College of Application Science and Technology, Nanchang 330000, China

² School of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, China

³ School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China; litaiyong@gmail.com

* Correspondence: donglin@zjnu.edu.cn (D.Z.); hugangqiang@zjnu.edu.cn (G.H.)

Abstract: In today's fast-paced and ever-changing environment, the need for algorithms with enhanced global optimization capability has become increasingly crucial due to the emergence of a wide range of optimization problems. To tackle this issue, we present a new algorithm called Random Particle Swarm Optimization (RPSO) based on cosine similarity. RPSO is evaluated using both the IEEE Congress on Evolutionary Computation (CEC) 2022 test dataset and Convolutional Neural Network (CNN) classification experiments. The RPSO algorithm builds upon the traditional PSO algorithm by incorporating several key enhancements. Firstly, the parameter selection is adapted and a mechanism called Random Contrastive Interaction (RCI) is introduced. This mechanism fosters information exchange among particles, thereby improving the ability of the algorithm to explore the search space more effectively. Secondly, quadratic interpolation (QI) is incorporated to boost the local search efficiency of the algorithm. RPSO utilizes cosine similarity for the selection of both QI and RCI, dynamically updating population information to steer the algorithm towards optimal solutions. In the evaluation using the CEC 2022 test dataset, RPSO is compared with recent variations of Particle Swarm Optimization (PSO) and top algorithms in the CEC community. The results highlight the strong competitiveness and advantages of RPSO, validating its effectiveness in tackling global optimization tasks. Additionally, in the classification experiments with optimizing CNNs for medical images, RPSO demonstrated stability and accuracy comparable to other algorithms and variants. This further confirms the value and utility of RPSO in improving the performance of CNN classification tasks.

Keywords: global optimization; particle swarm optimization; cosine similarity; classification



Citation: Liu, Y.; Zeng, Y.; Li, R.; Zhu, X.; Zhang, Y.; Li, W.; Li, T.; Zhu, D.; Hu, G. A Random Particle Swarm Optimization Based on Cosine Similarity for Global Optimization and Classification Problems.

Biomimetics **2024**, *9*, 204. <https://doi.org/10.3390/biomimetics9040204>

Academic Editors: Heming Jia, Ameer Hamza Khan, Danish Hussain and Shuai Li

Received: 26 February 2024

Revised: 23 March 2024

Accepted: 25 March 2024

Published: 28 March 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

Scientific and technological development generate a significant amount of data, including medical data, transaction data, house-price data, and more. Researchers can utilize this data to enhance their understanding of convenient technology. However, dealing with diverse types of data requires the construction of complex models, which are processed through machine learning. The diversity of data poses limitations on the machine learning approach, as the parameters in machine learning need to be updated differently depending on the data being processed. Consequently, optimizing machine learning becomes a major issue.

As an optimization technique, meta-heuristic algorithms are driven by natural phenomena and behaviors [1]. These algorithms possess the benefit of not requiring specialized understanding regarding the issue at hand. Instead, they iteratively explore and solve problems in a certain space, providing robustness and universality [2]. Particle Swarm Optimization (PSO) [3], a type of meta-heuristic algorithm, is representative and widely

used. PSO is an optimization technique that mimics the natural hunting behavior of birds. It is characterized by its simplicity, requiring fewer parameters and being easy to implement compared to other methods. PSO exhibits good learning ability through the interaction of individual information during the evolutionary process. Consequently, it effectively addresses numerous real-world issues, including energy systems [4], path planning [5], image processing [6], DNA computing [7], and engineering optimization [8]. Consequently, there is significant research interest in combining PSO with machine learning techniques. In summary, meta-heuristic algorithms, such as PSO, offer a powerful approach for solving optimization problems. Their ability to adapt and explore solution spaces without requiring explicit problem knowledge makes them valuable tools for a wide range of applications.

PSO has gained significant attention in recent research for its ability to solve challenging problems in machine learning. Several studies have applied PSO to optimize various machine learning models and algorithms. For instance, Yong Zhu improved the LeNet-5 model by employing PSO to optimize its parameters [9]. Wu Deng optimized the parameters of the Least Squares Support Vector Machine (LS-SVM) for fault classification using an enhanced PSO algorithm [10]. Rajesh K. Yadav and Anubhav introduced a hybrid training algorithm combining PSO and the genetic algorithm (GA) alongside Adam Optimization to enhance the training process of artificial neural networks [11]. Kun Zhang utilized a PSO-optimized Least Squares Support Vector Machine (LSSVM) to establish a fault diagnosis model [12]. J. Beschi Raja and S. Chenthur Pandian employed PSO to optimize the Fuzzy Clustering Mean (FCM) for disease prediction [13]. Mohammed Nasser Al-Andoli integrated PSO with the backpropagation (BP) algorithm for malware detection [14]. Chao Huang used PSO to optimize Deep Convolutional Neural Networks (DCNNs) for automatic defect detection and classification [15]. Thomas George employed PSO for feature selection in electroencephalogram (EEG) signal classification [16]. Mohamed Issa combined PSO with the bat optimization algorithm for improved detection techniques [17]. Abhishek Dixit utilized a mix of algorithms, primarily based on the Differential Evolutionary Algorithm (DE) and PSO, for optimizing feature selection and classification [18]. Muhammad Sharif employed PSO for image segmentation and the GA for feature selection [19]. Hanuman optimized the FCM clustering algorithm for medical image segmentation using PSO [20]. Debendra Muduli leveraged Multi-Objective Discrete Particle Swarm Optimization (MODPSO) to enhance the performance of feedforward neural networks in breast cancer detection by optimizing hidden node parameters [21]. However, traditional PSO approaches face challenges in complex optimization models, as they can easily become caught in local optima, resulting in premature convergence. Additionally, the stability of the optimization capability of PSO leaves room for improvement [22].

To further boost the optimization capability of PSO, this article proposes a random particle swarm optimization algorithm based on cosine similarity, referred to as RPSO. RPSO adjusts the internal parameters of PSO and introduces the Random Contrastive Interaction mechanism and quadratic interpolation (QI). It employs cosine similarity to choose between the two mechanisms, which efficiently and dynamically updates population information, enhancing the global and local optimization capabilities of the algorithm. The performance tests were carried out based on two dimensions of CEC 2022 and compared with the PSO variant and the top algorithm of CEC to verify the competitiveness and effectiveness of RPSO. Finally, RPSO was optimized for the hyperparameters of the CNN, and the optimized model was confirmed to possess superior diagnostic abilities in both datasets. The specific work performed and innovations produced in this study are as follows:

- Introduction of the Random Contrastive Interaction mechanism.
- Introduction of the Quadratic interpolation mechanism.
- Design of a judgment condition based on cosine similarity.
- Comparison of PSO variants and the top algorithm in two dimensions of CEC 2022.
- Optimization of CNN for the diagnosis of two diseases.

This paper is structured as follows: Section 2 introduces the theory of PSO and cosine similarity; Section 3 provides the introduction and analysis of the proposed algorithm; Section 5 presents the testing of the proposed algorithm based on CEC 2022; Section 4 highlights the improvement provided by the suggested algorithm when applied to CNN for classification experiments; and the last section concludes the work of the paper with a summary and discussion.

2. Related Theories

2.1. Particle Swarm Algorithms

In the PSO algorithm optimization process, the position information vector x is symbolized by a matrix of $N \times d$ and $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$, and the generated velocity vector $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$. The velocity and position are continuously revised through the following equation [12].

$$\begin{aligned} v_i(t+1) &= wv_i(t) + z_1r_1(x_{pb,i}(t) - x_i(t)) + z_2r_2(x_{gb,i}(t) - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (1)$$

In Equation (1), $v_i(t+1)$ and $x_i(t+1)$ represent individual i 's velocity and position during generation t , z_1 and z_2 denote the two acceleration coefficients, r_1 and r_2 denote the uniform random numbers $[0, 1]$, and w is the inertia weight. $x_{pb}(t)$ and $x_{gb}(t)$ denote the global optimum and the historical optimal position of an individual at generation t , respectively.

2.2. Cosine Similarity

The cosine similarity measure serves to assess how similar two vectors are by finding the cosine of their intersection angle [23]. This value ranges between -1 and 1 , where a cosine similarity value of 1 indicates that the vectors share the same direction, 0 indicates orthogonality (a 90-degree angle between the vectors), and -1 indicates that the vectors share opposite directions.

Unlike other similarity measures, cosine similarity disregards vector lengths and solely examines their pointing directions. As a result, it can determine whether two vectors are essentially heading in the same direction based solely on the cosine of the angle between them. Cosine similarity is typically utilized for positive spaces and yields values within the range $[-1, 1]$. This is illustrated in Figure 1, which represents the most common 2D space.

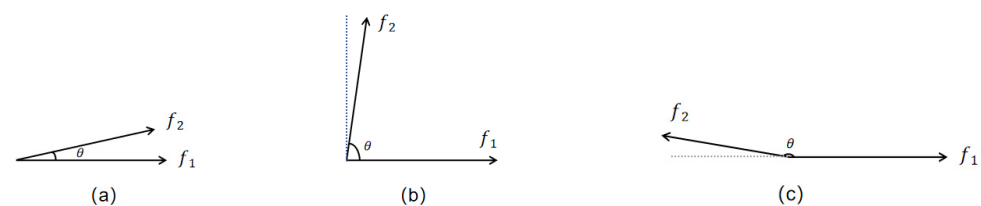


Figure 1. Angles in 2D space.

Figure 1a demonstrates that as the angle between two vectors approaches 0° , the cosine value approaches 1 , indicating a high similarity in direction. In Figure 1b, a cosine value close to 0 indicates that the vectors are orthogonal, meaning they are independent of each other. Figure 1c illustrates that a cosine value closer to -1 implies that the vectors are oriented in opposite directions.

It is important to note that cosine similarity is independent of vector length and only considers the direction of the vectors. The formula for calculating cosine similarity is as follows [24]:

$$\cos(\theta) = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2)$$

where A_i and B_i are the i -th components of A and B , respectively.

3. The Proposed Algorithm

This subsection provides a description of the proposed algorithm and its principles.

3.1. Related Research from PSO

In recent years, researchers have made significant advancements in improving the PSO, which can be broadly divided into four primary categories: hybrid algorithms, multiple populations or multi-sampling approaches, adaptive learning mechanisms, and other miscellaneous techniques.

Hybrid algorithms have gained popularity in enhancing the performance of PSO by combining different optimization algorithms. For instance, Zeng [25] incorporated the DE algorithm into PSO to address the issue of premature convergence. Zaman [26] fused the Backtracking Search Optimization Algorithm (BSA) with PSO by modifying the mutation and crossover operators using neighborhood information, resulting in an improved convergence rate. Xu [27] proposed a strategy learning framework that utilized an adaptive DE algorithm to learn an optimal combination of strategies. Taking inspiration from the bee-foraging search mechanism of the Artificial Bee Colony (ABC) algorithm, Chen [28] developed a hybrid algorithm to tackle complex global optimization problems (GOPs). Khan [29] combined the local search ability of the Gravitational Search Algorithm (GSA) with the social thinking capability (gbest) of PSO. Another approach was taken by Molaei [30], who combined PSO with the GA. Ahmed [31] proposed a novel Hybrid Particle Swarm Optimization and Genetic Algorithm (HPSOGA) to minimize the energy function of molecules. HPSOGA balances exploration and exploitation, enhances diversity, and avoids premature convergence. It outperforms standard PSO and benchmark algorithms in solving large-scale global optimization problems and molecular energy functions.

In the field of multiple population or multi-sample techniques, several approaches have been proposed to enhance Particle Swarm Optimization (PSO). Xia [32] developed a Triple Archives PSO (TAPSO) that incorporates three different types of archives: elite particles, profiteer particles, and outstanding exemplars. These archives guide future merit-seeking mechanisms through genetic properties. Zhang [33] introduced the Particle Swarm Optimization-Alternating Least Squares (PSO-ALS) algorithm, which adaptively groups the population into sub-swarms and further divides particles into ordinary particles and the locally best particle within each sub-swarm. Two different learning strategies are designed to update the particles. Li [34] divides the particle swarm into elite and normal groups based on a ranking system and defines different types of neighborhoods accordingly. Sheng [35] proposed a PSO variant with multi-level population sampling and dynamic p-learning mechanisms. Lu [36] proposed an Enhanced Multi-Swarm Cooperative Particle Swarm Optimizer (EMCPSO) that divides the entire population into four identical sub-swarms. This algorithm incorporates a delayed-activation (DA) mechanism to detect sub-swarm stagnation and introduces a repulsive mechanism to prevent premature clustering of the entire population.

In terms of adaptive learning mechanisms, Liu [37] developed a novel PSO algorithm that incorporates a sigmoid-function-based weighting strategy. This strategy adjusts the acceleration coefficients by considering the distance of particles to the global best position and their personal best position. Liu [38] also proposed several other adaptive mechanisms, including a chaos-based non-linear inertia weight, stochastic and mainstream learning strategies, an adaptive position updating strategy, and a terminal replacement mechanism. Li [39] introduced a landscape-based adaptive operator selection mechanism. This mechanism quantifies the landscape modality, selects the most suitable evolutionary operator based on the population's needs at different evolutionary stages, and adopts the mirrored boundary handling method to alleviate trapping in local optima. Zhang [40] proposed a particle swarm optimization algorithm with an empirical balance strategy (EBPSO), which selects a better search strategy from two equations using an adaptive adjustment mech-

anism. The algorithm dynamically adjusts the influence weight of the search equations and introduces a dynamic random search mechanism. Wang [41] proposed a hybrid PSO called Adaptive Strategy PSO (ASPSO), which incorporates a chaotic map, a position updating strategy, elite and dimensional learning strategies, and a competitive substitution mechanism. Li [42] proposed a novel PSO variant with a state-based adaptive velocity limit strategy (PSO-SAVL). In this strategy, the velocity limit is adaptively adjusted according to the estimated evolutionary state (ESE), with high values representing the global searching state and low values representing the local searching state.

In addition to the aforementioned categories, there are several other methods for improving PSO. Wang [43] presented a reinforcement learning strategy to adaptively adjust the optimization mechanism of PSO. Tian [44] proposed Variable Surrogate Model-based Particle Swarm Optimization (VSMPSO), which utilizes a single surrogate model constructed through simple random sampling and a variable model management strategy. Liu [45] incorporated Evolutionary Game Theory (EGT) to control the population state and proposed a selection mechanism and a mutation mechanism inspired by EGT concepts. Liu [46] employed Gaussian white noise with adjustable intensity to randomly perturb the acceleration coefficients, aiming to explore the problem space more extensively. Machado [47] introduced the concepts of complex-order derivatives (CD) and conjugate-order differentials.

Despite achieving certain optimization results, the aforementioned research approaches still have limitations. For example, hybrid algorithms may struggle to guarantee the complementary advantages of the two algorithms. Multiple swarms may not ensure effective local development. Adaptive mechanisms may possess irreversible search mechanisms. Additionally, it can be challenging to ensure the learning and adaptive ability of the algorithm in the existing research. To solve these limitations, this paper proposes a judgment mechanism based on cosine similarity and introduces the Random Contrastive Interaction (RCI) and QI mechanisms to jointly enhance the effective search capability of the algorithm.

3.2. Random Contrastive Interaction

In order to cope with complex and variable optimization environments, it is difficult for traditional particle swarms to maintain a good search ability to find high quality information, especially when the formulas use information about historical optimal positions. This can lead to a situation where multiple historical optimal positions cannot be updated with the global optimal position, resulting in a lack of diversity in the algorithm. For this reason, we introduce Random Contrastive Interaction [48], which provides better evolutionary information by means of the states within the population adaptively learning from each other. A specific schematic is shown in Figure 2.

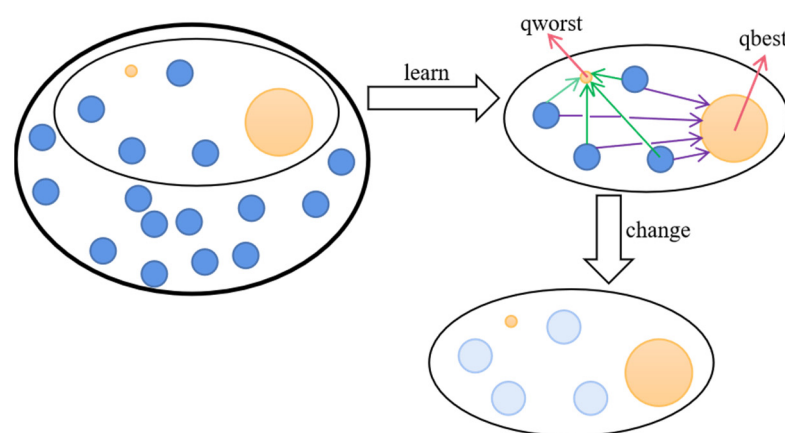


Figure 2. Schematic diagram of RCI.

As shown in Figure 2, P individuals are randomly selected from the population to form a random topology. Within this small population, there will be a local best solution,

$qbest$, and a worst solution, $qworst$. The role of RCI is to enable the individuals within the small population to learn from $qbest$ and $qworst$, and to interact with each other to obtain information, meaning that dynamically updating an individual's position enhances the flexibility of the algorithm. The details are as follows [48]:

$$\begin{aligned} Z_i &= Q_1 \cdot V_i + Q_2 \cdot (qbest - x_{pi}) + \varphi \cdot Q_3 \cdot (qworst - x_{pi}) \\ S_i &= x_{pi} + Z_i \end{aligned} \quad (3)$$

where, Q_1, Q_2, Q_3 are uniform random numbers in the range of $[0, 1]$ and φ is the learning parameter controlling $qworst$, which is 0.3. An optimal setting of φ for RPSO falls within a moderate range, neither too large (such as $\varphi = 0.5$) nor too small (such as $\varphi < 0.3$), as extremes hinder RPSO from achieving optimal performance. A φ that is too small diminishes the impact of the worst position on each updated particle, potentially causing rapid movement of particles towards the location of the best position and resulting in premature convergence or being trapped in local optima. On the other hand, a φ that is too large amplifies the influence of the worst position excessively, disrupting swarm convergence and leading to suboptimal performance of RPSO. Z_i is the velocity of the newly generated individual and S_i is the position of the newly generated individual. x_{pi} denotes the position of the i th individual in the selected population P . It can be seen that the selection of individuals makes it possible to continuously move from the locally optimal individual to the locally worst position, increasing the exchange of information in the population and also improving the global exploration capacity.

Which the number of randomly selected individuals is constantly changing, in the early stage of the algorithm, in the stage of global exploration needs global search capability, at this time a larger number of individuals need to RCI operation, and in the later stage, a portion of the individuals need to carry out a certain amount of exploration to increase the diversity of the population. In the literature [48], the variation in the size of the selected populations is shown in Figure 3. TS is the population size for each selection. As shown in the Figure 3, the randomly selected population size presents an increasing manner, which will lead to a problem that the local exploitation ability is very poor at the later stage, and the feasible solution cannot be mined effectively. Therefore, this paper designs a new population design scheme, which is formulated as follows:

$$\begin{aligned} TQ &= (rand(1) + 1) \times (1 - t/T) \times N/2 \\ TS_t &= round(max(TQ, N/10)) \end{aligned} \quad (4)$$

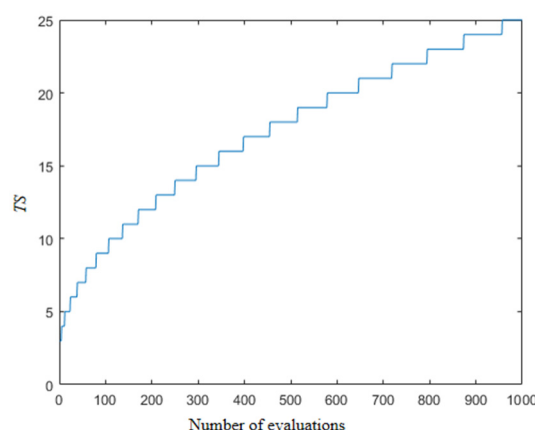


Figure 3. TS in literature [48].

The use of a max function is needed to ensure that there are at least $N/10$ individuals in the population. This ensures that the individuals need to carry out a certain exploration, in which the value of TQ is constantly changing. The overall trend is gradually decreasing, but this is an oscillating change. Taking the population number of 50 as an example, the

specific value changes as shown in Figure 4; it can be seen that the range of changes in the early stage is larger, and in the later stage, this value gradually becomes smaller, and finally the structure of $N/10$ is unanimously adopted. Such a screening mechanism not only ensures the diversity of the population but also enables a detailed search of a certain area to find a higher quality of feasible method. On the other hand, the oscillation characteristics are beneficial, mainly reflected in the reversible changes in the population size. This can diversify the search process and compensate for any vulnerabilities in the algorithm's search. The specific pseudo-code is shown in Algorithm 1.

Algorithm 1. Random Contrastive Interaction (RCI)

Input:

t : global optimum position

T : Matrix of population information

N : population size

fit : Matrix of objective function values

x : Position of the population

Output:

X^* : Matrix of newly generated stock information

The number of individuals TS to be selected for this iteration is calculated according to Equation (4);

1. $rs = randperm(N, TS)$; %Generate non-repeating individual subscripts based on TS .

2. $[af, bf] = sort(fit)$; % af is the result of the ascending order of the fitness values and bf is the corresponding subscript position.

3. $db = find(bf == rs(1))$; Find the location of the optimal individual in rs .

4. $dw = find(bf == rs(TS))$; Find the worst individual position in rs .

5. $dbest = pos(bbb(db),:)$;

6. $dworst = pos(bbb(dw),:)$;

7. **For** $i = 1:size(rs,2)$

8. **For** $j = 1:size(x,2)$

9. Calculate the corresponding new position in rs according to Equation (3).

10. **End**

11. **End**

Return X^*

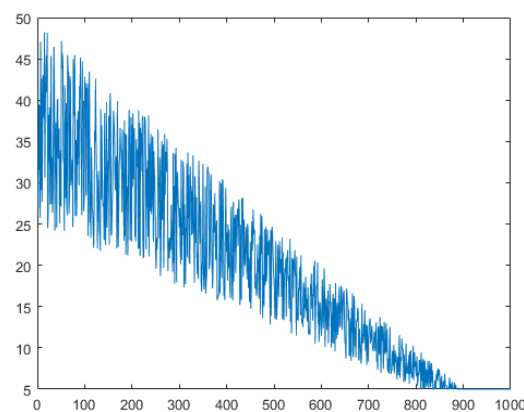


Figure 4. Proposed TS .

3.3. Quadratic Interpolation (QI)

QI is a better local search technique that employs a parabola to adapt the form of a quadratic function to locate the extreme points of a curve [49]. The quadratic interpolation technique is used to further accelerate the information exchange of the population and improve the local search ability. The specific formula is as follows [50]:

$$x_{new} = 0.5 \cdot \left(\frac{(x_1^2 - x_2^2) \cdot f(x_3) + (x_3^2 - x_1^2) \cdot f(x_1) + (x_1^2 - x_3^2) \cdot f(x_2)}{(x_1 - x_2) \cdot f(x_3) + (x_2 - x_3) \cdot f(x_1) + (x_1 - x_3) \cdot f(x_2)} \right) \quad (5)$$

where x_1 , x_2 , and x_3 are three individuals randomly selected from the population. In this paper, x_1 represents the location of the globally optimal individual, while x_2 and x_3 represent two randomly selected individuals from the population. Algorithm 2 provides the specific pseudo-code. By facilitating information interaction between the globally optimal individual and the two individuals from the population, the algorithm guides the population to converge towards the vicinity of the optimal solution, thereby enhancing the algorithm's local search capability.

Algorithm 2. Quadratic interpolation (QI)

Input:

gbest: global optimum position

x: Matrix of population information

Output:

x^{*}: Matrix of newly generated stock information

1. **For** $i = 1:\text{size}(x)$

2. $k = \text{randperm}(N, 2)\%$ Two non-repeating individuals were randomly generated.

3. The positional information of the new individual is calculated by substituting the three individuals $k(1)$, $k(2)$, and *gbest* into Equation (5).

4. **End**

Return *x*^{*}

3.4. Setting of Parameters

In the particle swarm algorithm, with fixed parameters, there is a fixed search amplitude and it is difficult to increase the diversity. Many scholars have proposed using the following formula for the setting of z_1 and z_2 with weights w [51].

$$\begin{aligned} w &= w_{\max} - (w_{\max} - w_{\min}) \times t/T \\ z_1 &= 2.5 - 2 \times t/T \\ z_2 &= 0.5 + 2 \times t/T \end{aligned} \quad (6)$$

where $w_{\max}$ is 0.9 and $w_{\min}$ is 0.2, w decreases with the number of iterations, and the population's diversity levels are gradually reducing. z_1 and z_2 changes are set by biasing the historical optimum and the global optimum; in the early stage of the algorithm, a larger z_1 and a smaller z_2 cause the historical optimum position to have a larger bias, which can cause the global exploration ability of the algorithm to increase, and a smaller z_1 and a larger z_2 can cause the search of the algorithm to bias towards the global optimum position and enhance the precision of the solution. This analysis shows that the combined effect of these three factors causes the algorithm to tend away from global exploration to local exploitation, and also balances the search scope of the algorithm.

3.5. Selection Mechanisms

In the process of algorithm optimization, establishing how to judge the population state more accurately is always an important direction of research. Making a clear plan according to the population state is the only way to ensure that the algorithm is constantly close to the theoretical optimal solution. Qiang Zhang used cosine similarity to dynamically update the mutation operation of DE [52], Moutaz Alazab used CS to convert continuous problems into binary problems [53], and Wei Li et al. used CS to upgrade algorithm speed [54]. In this paper, cosine similarity is introduced to determine the distance of an individual from the current optimal position, the number of individual similarity angles greater than 90 are recorded separately, and the choice of mechanism is made based on the number of recorded CN. When the angle is greater than 90 the cosine similarity is greater than 0, and vice versa. The specific selection mechanism is as follows:

$$\begin{cases} RCI & \text{if } CN < N/2 \\ QI & \text{otherwise} \end{cases} \quad (7)$$

where if $CN < N/2$ indicates that the similarity angle between most of the individuals within the current population and the current optimal position is less than 90 degrees. This reflects the fact that most of the individuals have high degrees of similarity with each other, and there is low population diversity. There also exists the probability of falling into the local optimum. This occurs when the RCI is used to improve the learning degree of the worst position and enhance the diversity of the population.

Meanwhile, if the number of individuals within the population with an angle greater than ninety degrees from the current optimal position is high, then this reflects the fact that the individuals within the population are not similar to the current optimal position that gives better global exploration ability. Under these circumstances, the QI mechanism is used to enhance the local search of the global optimal position, balancing the local and global search ability of the algorithm. This mechanism of selection ensures that each iteration does not completely lose the local exploitation or global exploration capacity, and it dynamically regulates the learning ability of the algorithm.

3.6. Flow of the Algorithm

To improve the judgment and learning ability of PSO, this paper proposes a stochastic learning PSO algorithm based on cosine similarity. Firstly, the algorithm is dynamically set up, and secondly, the selection of RCI and QI is carried out according to the characteristics of cosine similarity, jointly ensuring the capacity of the algorithm to globally explore and locally develop and improving the accuracy of the solution. Using the minimization problem as a case study, the specific pseudo-code of the Algorithm 3 is as follows.

Algorithm 3. RPSO

Input:

N : population size
 T : Matrix of population information
 $Maxfes$: Maximum number of iterations
 Dim : Dimension of the problem
 Lb : Lower bound of the problem
 Ub : Upper bound of the problem
 f : objective function

Output:

x_{gbest} : Matrix of newly generated stock information
 $Fmin$: optimal solution
1. The position matrix x is obtained by random initialization based on the set parameters.
2. Calculate the objective function value based on f to obtain the optimal position x_{gbest} , the optimal solution f_{min}
3. $Fes = N$;
4. $T = 1$;
5. **While** ($t \leq T$) && ($Fes \leq Maxfes$)
6. Calculate the new population position x according to Equations (1) and (7)
7. Calculate the objective function value according to f to get the current optimal position x_t its subscripts q , update the optimal position x_{gbest} , the optimal solution f_{min}
8. $Fes = Fes + N$;
9. $Nk = 0$
10. **For** $i = 1:N$
11. **If** $i \neq q$
12. The cosine similarity between the individual and x_l was calculated to obtain nc according to Equation (2)
13. **End**
14. **If** $nc < 0$
15. $Nk = Nk + 1$;
16. **end**
17. **end**

```

18.   If  $Nk > N/2$ 
19.     Perform according to Algorithm 2.
20.   Calculate the objective function value based on  $f$  to obtain the optimal position  $x_{gbest}$ , the optimal
      solution  $f_{min}$ 
21.    $Fes = Fes + TS$ ;
22.   else
23.     Perform according to Algorithm 1.
24.   Calculate the objective function value based on  $f$  to obtain the optimal position  $x_{gbest}$ , the optimal
      solution  $f_{min}$ 
25.    $Fes = Fes + N$ ;
26.   End
27.    $t = t + 1$ ;
28.   End
Return  $x_{gbest}, f_{min}$ 

```

4. Performance Tests

This section validates the proposed algorithm in the CEC 2022 test set, compares it with the variant of PSO and the top algorithm in the CEC competition, and conducts ablation experiments.

4.1. Comparison with PSO Variants

To demonstrate the optimized performance of RPSO, a comparison is conducted with recent variants of PSO algorithms to highlight the novelty and feasibility of RPSO. The compared variants include Adaptive multi-strategy ensemble particle swarm optimization (AMEPSO) [55], Elite archives-driven particle swarm optimization (EAPSO) [56], Improved Phasor Particle Swarm Optimization with Fitness Distance Balance (FDBPSO) [57], Velocity pausing particle swarm optimization (VPPSO) [58], a PSO variant for single-objective numerical optimization (PSOsono) [59], modified particle swarm optimization (MPSO), and Pyramid particle swarm optimization (PPSO) [60]. The internal parameter settings for each algorithm are based on the literature. The population size is set to 100, and the algorithms are tested on 10 and 20 dimensions using a maximum number of evaluations of 200,000 and 1,000,000. The experiments are performed utilizing MATLAB 2019a on an 11th Gen Intel(R) Core(TM) i5-11500 @ 2.70GHz processor. It is worth noting that in the program, all functions of CEC 2022 are minimization functions. We subtract the theoretical optimal value from the optimized result, so the theoretical optimal value becomes 0. The closer the optimization results of each algorithm are to 0, the better the optimization performance. Each algorithm is independently run 30 times, and the optimal value, worst value, median, mean, and standard deviation are recorded. These metrics are used to measure the stability and optimization capability of the algorithm. In simple terms, the smaller the value, the better the optimization effect of the algorithm. The Wilcoxon rank-sum test is used to analyze the differences between RPSO and other algorithms, with a significance level of 0.05. The symbol “+” indicates that RPSO performs better than a certain algorithm in a specific function, “−” indicates the opposite, and “=” shows that the optimization efficiency of the two algorithms is equal. Additionally, the Friedman test is used to measure the overall differences among algorithms and provide a final ranking based on the entire CEC 2022 test set [61]. The detailed optimization results are presented in Tables 1 and 2. To visualize the distribution of optimization performance for each algorithm, Figures 5 and 6 display the distribution of optimal solutions for each algorithm across different functions.

As shown in Table 1 and Figure 4, in the case of 10 dimensions the RPSO algorithm performs better than the other variants of the PSO algorithm in most cases. Specifically, the optimization standard deviation of the RPSO algorithm is the lowest value in the functions F1(X), F2(X), F5(X), F6(X), F7(X), F10(X), and F12(X). The theoretical optimum is found every time in F1(X), F5(X), and F11(X). In other cases, from small to large, the variance ranks second or third, indicating strong search stability.

Table 1. Ten dimensions with PSO algorithm variants.

F	Index	AMSE PSO	EAPSO	FDBPSO	MPSO	PPSO	PSOsono	VPPSO	RPSO
F1(x)	Best	0.00 × 10 ⁰⁰	1.13 × 10 ⁰³	1.12 × 10 ⁰³	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.93 × 10 ^{−9}	0.00 × 10 ⁰⁰
	Worst	1.26 × 10 ⁰²	1.76 × 10 ⁰⁴	1.29 × 10 ⁰⁴	2.25 × 10 ^{−07}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	7.62 × 10 ^{−9}	0.00 × 10 ⁰⁰
	Median	2.49 × 10 ⁰¹	6.75 × 10 ⁰³	5.09 × 10 ⁰³	1.38 × 10 ^{−10}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	4.34 × 10 ^{−9}	0.00 × 10 ⁰⁰
	Mean	3.25 × 10 ⁰¹	6.90 × 10 ⁰³	6.04 × 10 ⁰³	1.40 × 10 ^{−8}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	4.39 × 10 ^{−9}	0.00 × 10 ⁰⁰
	Standard	3.56 × 10 ⁰¹	4.25 × 10 ⁰³	2.94 × 10 ⁰³	4.57 × 10 ^{−8}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.28 × 10 ^{−9}	0.00 × 10 ⁰⁰
F2(x)	Contest	1.94 × 10 ^{−9} (+)	1.21 × 10 ^{−12} (+)	1.21 × 10 ^{−12} (+)	1.21 × 10 ^{−12} (+)	N/A (=)	N/A (=)	1.21 × 10 ^{−12} (+)	
	Best	9.47 × 10 ^{−03}	4.54 × 10 ⁰⁰	1.01 × 10 ⁰¹	8.20 × 10 ^{−8}	2.45 × 10 ^{−04}	0.00 × 10 ⁰⁰	1.45 × 10 ^{−02}	1.71 × 10 ^{−07}
	Worst	1.63 × 10 ⁰¹	9.26 × 10 ⁰¹	9.65 × 10 ⁰¹	7.08 × 10 ⁰¹	8.92 × 10 ⁰⁰	1.28 × 10 ⁰¹	9.28 × 10 ⁰⁰	7.32 × 10 ^{−02}
	Median	1.62 × 10 ⁰⁰	1.21 × 10 ⁰¹	3.11 × 10 ⁰¹	7.32 × 10 ⁰⁰	8.92 × 10 ⁰⁰	3.99 × 10 ⁰⁰	6.70 × 10 ⁰⁰	5.91 × 10 ^{−04}
	Mean	2.83 × 10 ⁰⁰	2.17 × 10 ⁰¹	4.00 × 10 ⁰¹	9.12 × 10 ⁰⁰	5.44 × 10 ⁰⁰	4.65 × 10 ⁰⁰	4.81 × 10 ⁰⁰	9.12 × 10 ^{−03}
F3(x)	Standard	3.58 × 10 ⁰⁰	2.16 × 10 ⁰¹	2.28 × 10 ⁰¹	1.73 × 10 ⁰¹	4.06 × 10 ⁰⁰	3.69 × 10 ⁰⁰	4.33 × 10 ⁰⁰	1.80 × 10 ^{−02}
	Contest	9.92 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	3.59 × 10 ^{−07} (+)	7.41 × 10 ^{−10} (+)	7.13 × 10 ^{−05} (+)	2.15 × 10 ^{−10} (+)	
	Best	0.00 × 10 ⁰⁰	3.31 × 10 ⁰⁰	5.74 × 10 ⁰⁰	2.09 × 10 ^{−11}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.44 × 10 ^{−04}	0.00 × 10 ⁰⁰
	Worst	1.45 × 10 ^{−11}	8.85 × 10 ⁰⁰	3.02 × 10 ⁰¹	2.33 × 10 ^{−05}	1.57 × 10 ^{−05}	1.22 × 10 ^{−11}	8.73 × 10 ⁰⁰	8.79 × 10 ^{−05}
	Median	5.53 × 10 ^{−03}	5.38 × 10 ⁰⁰	1.39 × 10 ⁰¹	3.06 × 10 ^{−8}	0.00 × 10 ⁰⁰	2.56 × 10 ^{−06}	5.67 × 10 ^{−11}	1.42 × 10 ^{−06}
F4(X)	Mean	2.40 × 10 ^{−02}	5.75 × 10 ⁰⁰	1.47 × 10 ⁰¹	8.73 × 10 ^{−07}	1.30 × 10 ^{−06}	5.53 × 10 ^{−03}	1.74 × 10 ⁰⁰	7.62 × 10 ^{−06}
	Standard	3.41 × 10 ^{−02}	1.55 × 10 ⁰⁰	5.78 × 10 ⁰⁰	4.25 × 10 ^{−06}	3.55 × 10 ^{−06}	2.24 × 10 ^{−02}	2.65 × 10 ⁰⁰	1.90 × 10 ^{−05}
	Contest	6.48 × 10 ^{−11} (=)	2.95 × 10 ^{−11} (+)	2.95 × 10 ^{−11} (+)	9.43 × 10 ^{−03} (−)	6.38 × 10 ^{−05} (−)	4.67 × 10 ^{−11} (=)	2.95 × 10 ^{−11} (+)	
	Best	1.55 × 10 ⁰⁰	2.40 × 10 ⁰¹	6.81 × 10 ⁰⁰	1.99 × 10 ⁰⁰	0.00 × 10 ⁰⁰	2.98 × 10 ⁰⁰	5.97 × 10 ⁰⁰	2.98 × 10 ⁰⁰
	Worst	1.60 × 10 ⁰¹	6.64 × 10 ⁰¹	3.70 × 10 ⁰¹	1.69 × 10 ⁰¹	6.96 × 10 ⁰⁰	1.69 × 10 ⁰¹	2.59 × 10 ⁰¹	9.95 × 10 ⁰⁰
F5(X)	Median	8.46 × 10 ⁰⁰	4.51 × 10 ⁰¹	1.88 × 10 ⁰¹	9.45 × 10 ⁰⁰	1.99 × 10 ⁰⁰	8.95 × 10 ⁰⁰	1.69 × 10 ⁰¹	8.46 × 10 ⁰⁰
	Mean	8.82 × 10 ⁰⁰	4.44 × 10 ⁰¹	1.86 × 10 ⁰¹	9.72 × 10 ⁰⁰	2.32 × 10 ⁰⁰	9.09 × 10 ⁰⁰	1.64 × 10 ⁰¹	7.63 × 10 ⁰⁰
	Standard	3.45 × 10 ⁰⁰	1.03 × 10 ⁰¹	6.25 × 10 ⁰⁰	3.82 × 10 ⁰⁰	1.60 × 10 ⁰⁰	3.07 × 10 ⁰⁰	5.12 × 10 ⁰⁰	2.08 × 10 ⁰⁰
	Contest	1.58 × 10 ^{−11} (=)	2.93 × 10 ^{−11} (+)	3.72 × 10 ^{−10} (+)	1.94 × 10 ^{−03} (+)	4.00 × 10 ^{−10} (−)	1.64 × 10 ^{−11} (=)	5.96 × 10 ^{−10} (+)	
	Best	0.00 × 10 ⁰⁰	3.75 × 10 ⁰⁰	4.85 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	8.43 × 10 ^{−10}	0.00 × 10 ⁰⁰
F6(X)	Worst	2.69 × 10 ^{−03}	9.28 × 10 ⁰¹	6.91 × 10 ⁰²	4.54 × 10 ^{−11}	4.54 × 10 ^{−11}	8.95 × 10 ^{−02}	2.82 × 10 ⁰⁰	0.00 × 10 ⁰⁰
	Median	0.00 × 10 ⁰⁰	1.77 × 10 ⁰¹	1.97 × 10 ⁰²	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.53 × 10 ^{−9}	0.00 × 10 ⁰⁰
	Mean	2.77 × 10 ^{−04}	2.58 × 10 ⁰¹	2.42 × 10 ⁰²	5.44 × 10 ^{−02}	2.41 × 10 ^{−02}	2.98 × 10 ^{−03}	3.99 × 10 ^{−11}	0.00 × 10 ⁰⁰
	Standard	7.17 × 10 ^{−04}	2.10 × 10 ⁰¹	1.77 × 10 ⁰²	1.38 × 10 ^{−11}	8.57 × 10 ^{−02}	1.63 × 10 ^{−02}	6.90 × 10 ^{−11}	0.00 × 10 ⁰⁰
	Contest	2.16 × 10 ^{−04} (+)	1.72 × 10 ^{−12} (+)	1.72 × 10 ^{−12} (+)	1.26 × 10 ^{−9} (+)	1.31 × 10 ^{−03} (+)	1.00 × 10 ⁰⁰ (=)	1.72 × 10 ^{−12} (+)	
F7(X)	Best	5.43 × 10 ⁰¹	1.84 × 10 ⁰³	1.39 × 10 ⁰⁴	2.46 × 10 ⁰¹	9.30 × 10 ⁰⁰	4.72 × 10 ⁰⁰	1.22 × 10 ⁰²	2.70 × 10 ⁰⁰
	Worst	4.98 × 10 ⁰²	7.29 × 10 ^{−14}	6.85 × 10 ^{−14}	1.96 × 10 ⁰³	4.18 × 10 ⁰³	1.37 × 10 ⁰³	6.25 × 10 ⁰³	1.96 × 10 ⁰²
	Median	1.26 × 10 ⁰²	7.69 × 10 ⁰⁴	1.27 × 10 ^{−14}	1.57 × 10 ⁰²	5.23 × 10 ⁰²	5.01 × 10 ⁰¹	1.82 × 10 ⁰³	6.32 × 10 ⁰¹
	Mean	1.64 × 10 ⁰²	1.24 × 10 ^{−14}	1.81 × 10 ^{−14}	3.70 × 10 ⁰²	1.13 × 10 ⁰³	1.47 × 10 ⁰²	2.38 × 10 ⁰³	7.83 × 10 ⁰¹
	Standard	1.11 × 10 ⁰²	1.44 × 10 ^{−14}	1.67 × 10 ⁰⁵	5.25 × 10 ⁰²	1.36 × 10 ⁰³	2.79 × 10 ⁰²	1.92 × 10 ⁰³	6.02 × 10 ⁰¹
F8(X)	Contest	3.01 × 10 ^{−04} (+)	3.02 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	4.23 × 10 ^{−03} (+)	5.61 × 10 ^{−05} (+)	6.10 × 10 ^{−11} (=)	2.37 × 10 ^{−10} (+)	
	Best	6.35 × 10 ^{−11}	2.07 × 10 ⁰¹	2.74 × 10 ⁰¹	6.44 × 10 ^{−05}	2.28 × 10 ^{−02}	2.81 × 10 ^{−11}	1.30 × 10 ⁰¹	1.96 × 10 ^{−11}
	Worst	2.73 × 10 ⁰¹	7.21 × 10 ⁰¹	7.86 × 10 ⁰¹	2.25 × 10 ⁰¹	2.79 × 10 ⁰¹	2.48 × 10 ⁰¹	5.04 × 10 ⁰¹	2.26 × 10 ⁰¹
	Median	1.83 × 10 ⁰¹	4.48 × 10 ⁰¹	5.22 × 10 ⁰¹	2.62 × 10 ⁰⁰	7.69 × 10 ⁰⁰	2.05 × 10 ⁰¹	3.00 × 10 ⁰¹	5.20 × 10 ⁰⁰
	Mean	1.43 × 10 ⁰¹	4.73 × 10 ⁰¹	5.29 × 10 ⁰¹	8.33 × 10 ⁰⁰	1.23 × 10 ⁰¹	1.43 × 10 ⁰¹	2.96 × 10 ⁰¹	9.88 × 10 ⁰⁰
F9(X)	Standard	1.10 × 10 ⁰¹	1.17 × 10 ⁰¹	1.27 × 10 ⁰¹	9.35 × 10 ⁰⁰	1.04 × 10 ⁰¹	9.93 × 10 ⁰⁰	8.52 × 10 ⁰⁰	9.20 × 10 ⁰⁰
	Contest	5.94 × 10 ^{−02} (=)	6.70 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	5.69 × 10 ^{−11} (=)	1.71 × 10 ^{−11} (=)	2.34 × 10 ^{−11} (=)	3.50 × 10 ^{−9} (+)	
	Best	3.80 × 10 ⁰⁰	2.83 × 10 ⁰¹	2.63 × 10 ⁰¹	9.43 × 10 ^{−11}	6.80 × 10 ⁰⁰	1.63 × 10 ^{−11}	3.10 × 10 ⁰⁰	2.21 × 10 ^{−11}
	Worst	2.61 × 10 ⁰¹	4.35 × 10 ⁰¹	4.45 × 10 ⁰¹	1.43 × 10 ⁰²	2.15 × 10 ⁰¹	2.32 × 10 ⁰¹	2.78 × 10 ⁰¹	2.10 × 10 ⁰¹
	Median	1.98 × 10 ⁰¹	3.36 × 10 ⁰¹	3.23 × 10 ⁰¹	2.09 × 10 ⁰¹	2.04 × 10 ⁰¹	2.03 × 10 ⁰¹	2.33 × 10 ⁰¹	4.36 × 10 ⁰⁰
F10(X)	Mean	1.70 × 10 ⁰¹	3.47 × 10 ⁰¹	3.33 × 10 ⁰¹	2.23 × 10 ⁰¹	1.99 × 10 ⁰¹	1.43 × 10 ⁰¹	2.28 × 10 ⁰¹	6.25 × 10 ⁰⁰
	Standard	8.23 × 10 ⁰⁰	4.10 × 10 ⁰⁰	4.77 × 10 ⁰⁰	2.40 × 10 ⁰¹	2.69 × 10 ⁰⁰	9.38 × 10 ⁰⁰	4.99 × 10 ⁰⁰	6.40 × 10 ⁰⁰
	Contest	1.17 × 10 ^{−05} (+)	3.02 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	4.42 × 10 ^{−06} (+)	2.57 × 10 ^{−07} (+)	3.64 × 10 ^{−02} (+)	5.07 × 10 ^{−10} (+)	
	Best	2.31 × 10 ⁰²	2.31 × 10 ⁰²	2.47 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²	2.30 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²
	Worst	2.35 × 10 ⁰²	2.52 × 10 ⁰²	5.27 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²	2.31 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²
F11(X)	Median	2.32 × 10 ⁰²	2.34 × 10 ⁰²	3.77 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²	2.31 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²
	Mean	2.32 × 10 ⁰²	2.35 × 10 ⁰²	3.60 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²	2.31 × 10 ⁰²	2.29 × 10 ⁰²	2.29 × 10 ⁰²
	Standard	1.07 × 10 ⁰⁰	4.26 × 10 ⁰⁰	6.49 × 10 ⁰¹	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	3.41 × 10 ^{−11}	1.58 × 10 ^{−06}	0.00 × 10 ⁰⁰
	Contest	1.21 × 10 ^{−12} (+)	1.21 × 10 ^{−12} (+)	1.21 × 10 ^{−12} (+)	N/A (=)	N/A (=)	1.21 × 10 ^{−12} (+)	1.21 × 10 ^{−12} (+)	
	Best	1.00 × 10 ⁰²	1.01 × 10 ⁰²	1.01 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²
F12(X)	Worst	2.18 × 10 ⁰²	2.60 × 10 ⁰²	3.15 × 10 ⁰²	2.17 × 10 ⁰²	2.13 × 10 ⁰²	2.19 × 10 ⁰²	2.32 × 10 ⁰²	1.00 × 10 ⁰²
	Median	1.02 × 10 ⁰²	2.39 × 10 ⁰²	1.03 × 10 ⁰²	1.00 × 10 ⁰²	1.01 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²
	Mean	1.06 × 10 ⁰²	2.26 × 10 ⁰²	1.40 × 10 ⁰²	1.29 × 10 ⁰²	1.33 × 10 ⁰²	1.22 × 10 ⁰²	1.05 × 10 ⁰²	1.00 × 10 ⁰²
	Standard	2.13 × 10 ⁰¹	4.33 × 10 ⁰¹	7.33 × 10 ⁰¹	4.92 × 10 ⁰¹	5.04 × 10 ⁰¹	4.46 × 10 ⁰¹	2.41 × 10 ⁰¹	5.59 × 10 ^{−02}
	Contest	5.53 × 10 ^{−8} (+)	3.02 × 10 ^{−11} (+)	3.02 × 10 ^{−11} (+)	1.69 × 10 ^{−9} (+)	4.98E-1 (+)	3.26 × 10 ^{−07} (+)	2.20 × 10 ^{−07} (+)	0.00 × 10 ⁰⁰
F13(X)	Best	0.00 × 10 ⁰⁰	6.70 × 10 ⁰¹	1.55 × 10 ⁰²	5.78 × 10 ^{−11}	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	4.29 × 10 ^{−04}	0.00 × 10 ⁰⁰
	Worst								

Regarding the worst value, the RPSO algorithm beats the AMSEPSO, EAPSO, and PDBPSO algorithms 12 times, beats the PSOsono algorithms 11 times, beats the VPPSO algorithm 10 times, and beats the MPSO and PPSO algorithms 9 times and 8 times, respectively. The above information reveals that the RPSO algorithm outperforms the AMSEPSO, EAPSO, FDBPSO, and VPPSO algorithms multiple times in terms of both the optimal value and the worst value. The RPSO algorithm is obviously superior to the above four algorithms. Although the advantages of the MPSO, PPSO and PSOsono algorithms are not obvious in the optimal value, they show great advantages in the worst value. Therefore, the superiority of the proposed algorithm over the variants of the PSO algorithm is verified.

Table 2. Twenty dimensions with PSO algorithm variants.

F	Index	AMSE PSO	EAPSO	FDBPSO	MPSO	PPSO	PSOsono	VPPSO	RPSO
F1(x)	Best	0.00 × 10 ⁰⁰	2.34 × 10 ⁰³	7.81 × 10 ⁰³	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.02 × 10 ⁻⁸	0.00 × 10 ⁰⁰
	Worst	1.87 × 10 ⁰²	1.78 × 10 ⁰⁴	3.39 × 10 ⁰⁴	0.00 × 10 ⁰⁰	7.59 × 10 ⁻¹¹	0.00 × 10 ⁰⁰	2.35 × 10 ⁻⁸	0.00 × 10 ⁰⁰
	Median	4.17 × 10 ⁰¹	9.44 × 10 ⁰³	1.88 × 10 ⁰⁴	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	1.59 × 10 ⁻⁸	0.00 × 10 ⁰⁰
	Mean	5.48 × 10 ⁰¹	8.82 × 10 ⁰³	1.88 × 10 ⁰⁴	0.00 × 10 ⁰⁰	2.53 × 10 ⁻⁰²	0.00 × 10 ⁰⁰	1.57 × 10 ⁻⁸	0.00 × 10 ⁰⁰
	Standard	5.89 × 10 ⁰¹	4.09 × 10 ⁰³	6.63 × 10 ⁰³	0.00 × 10 ⁰⁰	1.38 × 10 ⁻¹¹	0.00 × 10 ⁰⁰	2.94 × 10 ⁻⁹	0.00 × 10 ⁰⁰
F2(x)	Contest	3.98 × 10 ⁻⁸ (+)	7.57 × 10 ⁻¹² (+)	7.57 × 10 ⁻¹² (+)	2.63 × 10 ⁻⁸ (+)	7.46 × 10 ⁻¹² (+)	3.43 × 10 ⁻⁰² (+)	7.57 × 10 ⁻¹² (+)	
	Best	2.28 × 10 ⁰¹	4.91 × 10 ⁰¹	8.97 × 10 ⁰¹	2.66 × 10 ⁻¹¹	2.39 × 10 ⁻⁸	2.32 × 10 ⁻⁰⁴	2.47 × 10 ⁻⁰²	6.67 × 10 ⁰⁰
	Worst	7.85 × 10 ⁰¹	7.43 × 10 ⁰¹	4.98 × 10 ⁰²	6.73 × 10 ⁰¹	6.66 × 10 ⁰¹	7.15 × 10 ⁰¹	7.07 × 10 ⁰¹	4.91 × 10 ⁰¹
	Median	5.94 × 10 ⁰¹	4.91 × 10 ⁰¹	1.62 × 10 ⁰²	4.91 × 10 ⁰¹	4.91 × 10 ⁰¹	5.14 × 10 ⁰¹	4.70 × 10 ⁰¹	4.91 × 10 ⁰¹
	Mean	5.86 × 10 ⁰¹	5.68 × 10 ⁰¹	1.96 × 10 ⁰²	4.45 × 10 ⁰¹	4.15 × 10 ⁰¹	5.09 × 10 ⁰¹	3.12 × 10 ⁰¹	4.54 × 10 ⁰¹
F3(x)	Standard	1.07 × 10 ⁰¹	1.11 × 10 ⁰¹	1.05 × 10 ⁰²	1.71 × 10 ⁰¹	1.88 × 10 ⁰¹	1.50 × 10 ⁰¹	2.55 × 10 ⁰¹	8.82 × 10 ⁰⁰
	Contest	5.04 × 10 ⁻¹⁰ (+)	3.00 × 10 ⁻¹¹ (+)	3.00 × 10 ⁻¹¹ (+)	8.19 × 10 ⁻¹¹ (=)	4.97 × 10 ⁻⁰³ (-)	8.44 × 10 ⁻⁹ (+)	7.51 × 10 ⁻¹¹ (=)	
	Best	0.00 × 10 ⁰⁰	9.09 × 10 ⁻⁰²	1.02 × 10 ⁰¹	0.00 × 10 ⁰⁰	3.19 × 10 ⁻⁰⁷	0.00 × 10 ⁰⁰	6.42 × 10 ⁻¹¹	7.22 × 10 ⁻⁰⁴
	Worst	7.91 × 10 ⁻⁰²	4.40 × 10 ⁻¹¹	3.43 × 10 ⁰¹	9.10 × 10 ⁻⁰²	5.53 × 10 ⁻⁰³	5.94 × 10 ⁻¹¹	1.85 × 10 ⁰¹	8.08 × 10 ⁻¹¹
	Median	1.14 × 10 ⁻¹³	1.82 × 10 ⁻¹¹	1.75 × 10 ⁰¹	0.00 × 10 ⁰⁰	1.69 × 10 ⁻⁰⁵	8.24 × 10 ⁻⁰³	4.70 × 10 ⁰⁰	6.66 × 10 ⁻⁰²
F4(X)	Mean	1.21 × 10 ⁻⁰²	2.02 × 10 ⁻¹¹	1.83 × 10 ⁰¹	3.03 × 10 ⁻⁰³	4.48 × 10 ⁻⁰⁴	1.02 × 10 ⁻¹¹	7.11 × 10 ⁰⁰	1.28 × 10 ⁻¹¹
	Standard	2.06 × 10 ⁻⁰²	8.29 × 10 ⁻⁰²	6.23 × 10 ⁰⁰	1.66 × 10 ⁻⁰²	1.21 × 10 ⁻⁰³	1.68 × 10 ⁻¹¹	6.15 × 10 ⁰⁰	1.71 × 10 ⁻¹¹
	Contest	1.17 × 10 ⁻⁰⁶ (-)	9.52 × 10 ⁻⁰⁴ (+)	3.02 × 10 ⁻¹¹ (+)	1.37 × 10 ⁻¹¹ (-)	5.46 × 10 ⁻¹¹ (-)	2.71 × 10 ⁻⁰² (-)	4.08 × 10 ⁻¹¹ (+)	
	Best	8.33 × 10 ⁰⁰	1.81 × 10 ⁰¹	3.08 × 10 ⁰¹	1.69 × 10 ⁰¹	5.97 × 10 ⁰⁰	5.97 × 10 ⁰⁰	3.28 × 10 ⁰¹	6.96 × 10 ⁰⁰
	Worst	6.07 × 10 ⁰¹	1.14 × 10 ⁰²	6.69 × 10 ⁰¹	5.27 × 10 ⁰¹	1.49 × 10 ⁰¹	4.38 × 10 ⁰¹	9.15 × 10 ⁰¹	1.99 × 10 ⁰¹
F5(X)	Median	3.89 × 10 ⁰¹	4.33 × 10 ⁰¹	4.98 × 10 ⁰¹	3.34 × 10 ⁰¹	8.95 × 10 ⁰⁰	1.99 × 10 ⁰¹	4.97 × 10 ⁰¹	1.69 × 10 ⁰¹
	Mean	4.01 × 10 ⁰¹	4.71 × 10 ⁰¹	5.00 × 10 ⁰¹	3.20 × 10 ⁰¹	9.42 × 10 ⁰⁰	2.07 × 10 ⁰¹	5.38 × 10 ⁰¹	1.62 × 10 ⁰¹
	Standard	1.08 × 10 ⁰¹	1.75 × 10 ⁰¹	9.60 × 10 ⁰⁰	1.08 × 10 ⁰¹	2.66 × 10 ⁰⁰	8.06 × 10 ⁰⁰	1.54 × 10 ⁰¹	3.27 × 10 ⁰⁰
	Contest	4.48 × 10 ⁻¹⁰ (+)	5.31 × 10 ⁻¹¹ (+)	2.92 × 10 ⁻¹¹ (+)	5.55 × 10 ⁻¹⁰ (+)	1.91 × 10 ⁻⁸ (-)	2.78 × 10 ⁻⁰² (+)	2.92 × 10 ⁻¹¹ (+)	
	Best	1.77 × 10 ⁻⁰⁷	2.37 × 10 ⁻⁰²	5.38 × 10 ⁰²	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	0.00 × 10 ⁰⁰	3.51 × 10 ⁻⁹	0.00 × 10 ⁰⁰
F6(X)	Worst	1.56 × 10 ⁰⁰	2.64 × 10 ⁰⁰	2.93 × 10 ⁰³	3.34 × 10 ⁰¹	3.97 × 10 ⁰⁰	1.29 × 10 ⁰¹	8.87 × 10 ⁰²	0.00 × 10 ⁰⁰
	Median	5.08 × 10 ⁻¹¹	2.00 × 10 ⁻¹¹	1.33 × 10 ⁰³	2.24 × 10 ⁰⁰	4.48 × 10 ⁻⁰²	1.14 × 10 ⁻¹³	9.09 × 10 ⁻¹¹	0.00 × 10 ⁰⁰
	Mean	6.13 × 10 ⁻¹¹	4.13 × 10 ⁻¹¹	1.41 × 10 ⁰³	5.77 × 10 ⁰⁰	4.27 × 10 ⁻¹¹	1.11 × 10 ⁰⁰	3.98 × 10 ⁰¹	0.00 × 10 ⁰⁰
	Standard	4.49 × 10 ⁻¹¹	5.54 × 10 ⁻¹¹	5.85 × 10 ⁰²	8.84 × 10 ⁰⁰	8.98 × 10 ⁻¹¹	2.89 × 10 ⁰⁰	1.62 × 10 ⁰²	0.00 × 10 ⁰⁰
	Contest	1.40 × 10 ⁻¹¹ (+)	1.40 × 10 ⁻¹¹ (+)	1.40 × 10 ⁻¹¹ (+)	7.34 × 10 ⁻¹¹ (+)	4.94 × 10 ⁻¹¹ (=)	8.70 × 10 ⁻¹¹ (=)	1.40 × 10 ⁻¹¹ (+)	
F7(X)	Best	1.86 × 10 ⁰²	2.53 × 10 ⁰²	2.78 × 10 ⁻¹⁴	3.82 × 10 ⁰¹	6.54 × 10 ⁰¹	2.16 × 10 ⁰¹	1.44 × 10 ⁰²	5.55 × 10 ⁰¹
	Worst	3.29 × 10 ⁰³	2.17 × 10 ⁰⁴	6.58E+08	4.52 × 10 ⁰³	6.74 × 10 ⁰³	7.07 × 10 ⁰³	2.30 × 10 ⁰⁴	2.47 × 10 ⁰²
	Median	8.08 × 10 ⁰²	3.82 × 10 ⁰³	2.79E+06	7.34 × 10 ⁰²	1.49 × 10 ⁰³	3.14 × 10 ⁰²	1.95 × 10 ⁰³	1.26 × 10 ⁰²
	Mean	1.03 × 10 ⁰³	5.20 × 10 ⁰³	4.94E+07	1.29 × 10 ⁰³	2.01 × 10 ⁰³	1.65 × 10 ⁰³	3.46 × 10 ⁰³	1.30 × 10 ⁰²
	Standard	8.26 × 10 ⁰²	5.68 × 10 ⁰³	1.41E+08	1.21 × 10 ⁰³	1.76 × 10 ⁰³	2.31 × 10 ⁰³	5.12 × 10 ⁰³	5.08 × 10 ⁰¹
F8(X)	Contest	6.70 × 10 ⁻¹¹ (+)	3.02 × 10 ⁻¹¹ (+)	3.02 × 10 ⁻¹¹ (+)	9.26 × 10 ⁻⁹ (+)	7.77 × 10 ⁻⁹ (+)	2.07 × 10 ⁻⁰² (+)	1.17 × 10 ⁻⁹ (+)	
	Best	2.13 × 10 ⁰¹	1.82 × 10 ⁰¹	5.92 × 10 ⁰¹	2.19 × 10 ⁰¹	2.31 × 10 ⁰¹	2.63 × 10 ⁰¹	2.20 × 10 ⁰¹	2.11 × 10 ⁰¹
	Worst	5.24 × 10 ⁰¹	9.03 × 10 ⁰¹	2.05 × 10 ⁰²	6.74 × 10 ⁰¹	7.53 × 10 ⁰¹	5.35 × 10 ⁰¹	8.82 × 10 ⁰¹	3.00 × 10 ⁰¹
	Median	3.82 × 10 ⁰¹	4.03 × 10 ⁰¹	1.14 × 10 ⁰²	3.70 × 10 ⁰¹	3.87 × 10 ⁰¹	3.28 × 10 ⁰¹	3.74 × 10 ⁰¹	2.66 × 10 ⁰¹
	Mean	3.67 × 10 ⁰¹	4.49 × 10 ⁰¹	1.20 × 10 ⁰²	3.81 × 10 ⁰¹	4.16 × 10 ⁰¹	3.58 × 10 ⁰¹	4.16 × 10 ⁰¹	2.58 × 10 ⁰¹
F9(X)	Standard	7.47 × 10 ⁰⁰	1.79 × 10 ⁰¹	3.97 × 10 ⁰¹	1.24 × 10 ⁰¹	1.31 × 10 ⁰¹	7.73 × 10 ⁰⁰	1.51 × 10 ⁰¹	2.75 × 10 ⁰⁰
	Contest	3.01 × 10 ⁻⁰⁷ (-)	6.01 × 10 ⁻⁸ (+)	3.02 × 10 ⁻¹¹ (+)	1.53 × 10 ⁻⁰⁵ (+)	5.00 × 10 ⁻⁹ (+)	1.31 × 10 ⁻⁸ (+)	2.60 × 10 ⁻⁸ (+)	
	Best	2.16 × 10 ⁰¹	2.18 × 10 ⁰¹	2.96 × 10 ⁰¹	2.04 × 10 ⁰¹	2.04 × 10 ⁰¹	2.06 × 10 ⁰¹	2.34 × 10 ⁰¹	1.90 × 10 ⁰¹
	Worst	3.17 × 10 ⁰¹	2.62 × 10 ⁰²	3.25 × 10 ⁰²	1.21 × 10 ⁰²	1.42 × 10 ⁰²	2.51 × 10 ⁰¹	4.13 × 10 ⁰¹	2.91 × 10 ⁰¹
	Median	2.70 × 10 ⁰¹	1.44 × 10 ⁰²	3.88 × 10 ⁰¹	2.12 × 10 ⁰¹	2.13 × 10 ⁰¹	2.18 × 10 ⁰¹	2.62 × 10 ⁰¹	2.40 × 10 ⁰¹
F10(X)	Mean	2.70 × 10 ⁰¹	1.07 × 10 ⁰²	7.49 × 10 ⁰¹	2.47 × 10 ⁰¹	2.92 × 10 ⁰¹	2.21 × 10 ⁰¹	2.69 × 10 ⁰¹	2.46 × 10 ⁰¹
	Standard	2.36 × 10 ⁰⁰	7.85 × 10 ⁰¹	8.19 × 10 ⁰¹	1.82 × 10 ⁰¹	3.04 × 10 ⁰¹	1.08 × 10 ⁰⁰	3.31 × 10 ⁰⁰	2.60 × 10 ⁰⁰
	Contest	7.70 × 10 ⁻⁰⁴ (+)	1.29 × 10 ⁻⁰⁶ (+)	3.02 × 10 ⁻¹¹ (+)	6.05 × 10 ⁻⁰⁷ (+)	3.83 × 10 ⁻⁰⁶ (+)	4.08 × 10 ⁻⁰⁵ (-)	2.89 × 10 ⁻⁰³ (+)	
	Best	1.83 × 10 ⁰²	1.81 × 10 ⁰²	2.35 × 10 ⁰²	1.81 × 10 ⁰²	1.81 × 10 ⁰²	1.83 × 10 ⁰²	1.81 × 10 ⁰²	1.82 × 10 ⁰²
	Worst	1.84 × 10 ⁰²	1.81 × 10 ⁰²	5.39 × 10 ⁰²	1.81 × 10 ⁰²	1.81 × 10 ⁰²	1.89 × 10 ⁰²	1.81 × 10 ⁰²	1.85 × 10 ⁰²
F11(X)	Median	1.84 × 10 ⁰²	1.81 × 10 ⁰²	3.90 × 10 ⁰²	1.81 × 10 ⁰²	1.81 × 10 ⁰²	1.85 × 10 ⁰²	1.81 × 10 ⁰²	1.84 × 10 ⁰²
	Mean	1.84 × 10 ⁰²	1.81 × 10 ⁰²	3.83 × 10 ⁰²	1.81 × 10 ⁰²	1.81 × 10 ⁰²	1.85 × 10 ⁰²	1.81 × 10 ⁰²	1.84 × 10 ⁰²
	Standard	4.88 × 10 ⁻¹¹	1.60 × 10 ⁻⁰²	7.32 × 10 ⁰¹	0.00 × 10 ⁰⁰	4.51 × 10 ⁻⁰⁵	1.67 × 10 ⁰⁰	2.48 × 10 ⁻⁰⁵	7.37 × 10 ⁻¹¹
	Contest	4.20 × 10 ⁻¹¹ (=)	3.02 × 10 ⁻¹¹ (-)	3.02 × 10 ⁻¹¹ (+)	2.36 × 10 ⁻¹² (-)	3.01 × 10 ⁻¹¹ (-)	5.32 × 10 ⁻⁰³ (+)	3.02 × 10 ⁻¹¹ (-)	
	Best	6.60 × 10 ⁰¹	2.16 × 10 ⁰²	1.01 × 10 ⁰²	1.27 × 10 ⁰¹	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²
F12(X)	Worst	1.24 × 10 ⁰²	2.61 × 10 ⁰²	2.80 × 10 ⁰³	4.98 × 10 ⁰²	8.76 × 10 ⁰²	2.78 × 10 ⁰²	1.71 × 10 ⁰³	1.01 × 10 ⁰²
	Median	1.06 × 10 ⁰²	2.39 × 10 ⁰²	1.13 × 10 ⁰²	1.01 × 10 ⁰²	1.01 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²	1.00 × 10 ⁰²
	Mean	1.06 × 10 ⁰²	2.39 × 10 ⁰²	4.53 × 10 ⁰²	1.80 × 10 ⁰²	1.87 × 10 ⁰²	1.35 × 10 ⁰²	1.85 × 10 ⁰²	1.00 × 10 ⁰²
	Standard	1.01 × 10 ⁰¹	1.10 × 10 ⁰¹	7.57 × 10 ⁰²	1.21 × 10 ⁰²	1.76 × 10 ⁰²	6.39 × 10 ⁰¹	2.97 × 10 ⁰²	8.67 × 10 ⁻⁰²
	Contest	4.80 × 10 ⁻⁰⁷ (+)	3.02 × 10 ⁻¹¹ (+)	3.02 × 10 ⁻¹¹ (+)	2.43 × 10 ⁻⁰⁵ (+)	5.00 × 10 ⁻⁹ (+)	1.05 × 10 ⁻¹¹ (=)	7.06 × 10 ⁻¹¹ (=)	
F13(X)	Best	5.60 × 10 ⁻¹¹	3.01 × 10 ⁰²	8.19 × 10 ⁰²	3.00 × 10 ⁰²	3.00 × 10 ⁰²	0.00 × 10 ⁰⁰	2.67 × 10 ⁻⁰³	0.00 × 10 ⁰⁰
	Worst	5.43 × 10 ⁰²	4.02 × 10 ⁰²	4.10 × 10 ⁰³	9.06 × 10 ⁰²	4.00 × 10 ⁰²	4.00 × 10 ⁰²	1.76 × 10 ⁰³	3.00 × 10 ⁰²
	Median	3.00 × 10 ⁰²	3.02 × 10 ⁰²	1.60 × 10 ⁰³	3.00 × 10 ⁰²	3.00 × 10 ⁰²	3.00 × 10 ⁰²	3.00 × 10 ⁰²	3.00 × 10 ⁰²
	Mean	3.21 × 10 ⁰²	3.09 × 10 ⁰²	1.91 × 10 ⁰³	4.29 × 10 ⁰²	3.10 × 10 ⁰²	3.13 × 10 ⁰²	3.86 × 10 ⁰²	2

Table 2. Cont.

F	Index	AMSE PSO	EAPSO	FDBPSO	MPSO	PPSO	PSOsono	VPPSO	RPSO
F12(X)	Best	2.42×10^{02}	2.36×10^{02}	2.65×10^{02}	2.34×10^{02}	2.37×10^{02}	2.47×10^{02}	2.36×10^{02}	2.53×10^{02}
	Worst	3.00×10^{02}	2.73×10^{02}	5.30×10^{02}	2.95×10^{02}	2.76×10^{02}	3.50×10^{02}	5.03×10^{02}	2.70×10^{02}
	Median	2.69×10^{02}	2.44×10^{02}	3.19×10^{02}	2.50×10^{02}	2.53×10^{02}	2.75×10^{02}	2.52×10^{02}	2.62×10^{02}
	Mean	2.69×10^{02}	2.48×10^{02}	3.39×10^{02}	2.53×10^{02}	2.55×10^{02}	2.77×10^{02}	2.67×10^{02}	2.62×10^{02}
	Standard	1.22×10^{01}	9.10×10^{00}	6.34×10^{01}	1.35×10^{01}	1.10×10^{01}	2.22×10^{01}	4.92×10^{01}	3.94×10^{00}
	Contest	3.85×10^{-03} (+)	3.65×10^{-8} (−)	7.39×10^{-11} (+)	7.70×10^{-04} (−)	1.52×10^{-03} (−)	4.98×10^{-04} (+)	1.70×10^{-02} (+)	
	+ / − / Rank	2009/1/2 4.42	10/0/2 5.12	12/0/0 7.83	2008/1/3 3.50	2005/2/5 3.58	2007/3/2 3.92	2009/2/1 5.00	2.58

As shown in Table 2 and Figure 5, in the case of 20 dimensions, the RPSO algorithm performs better than other variants of the PSO algorithm in most cases. Specifically, the optimization standard deviation of the RPSO algorithm is the lowest value in functions F1(X), F2(X), F6(X), F10(X), F11(X), and F12(X). The theoretical optimum is found every time in F1(X) and F5(X). In other cases, from small to large, the variance ranks second or third, indicating strong search stability.

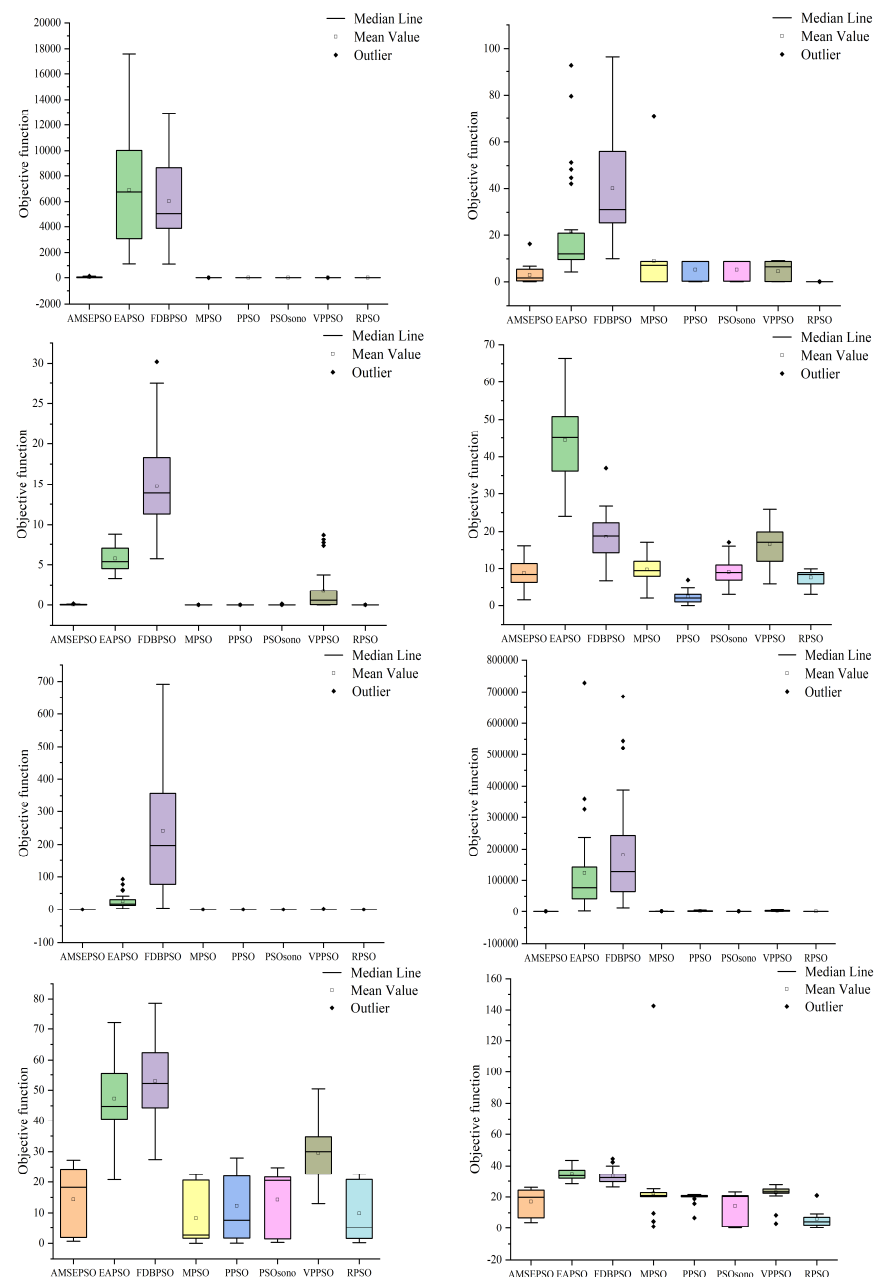


Figure 5. Cont.

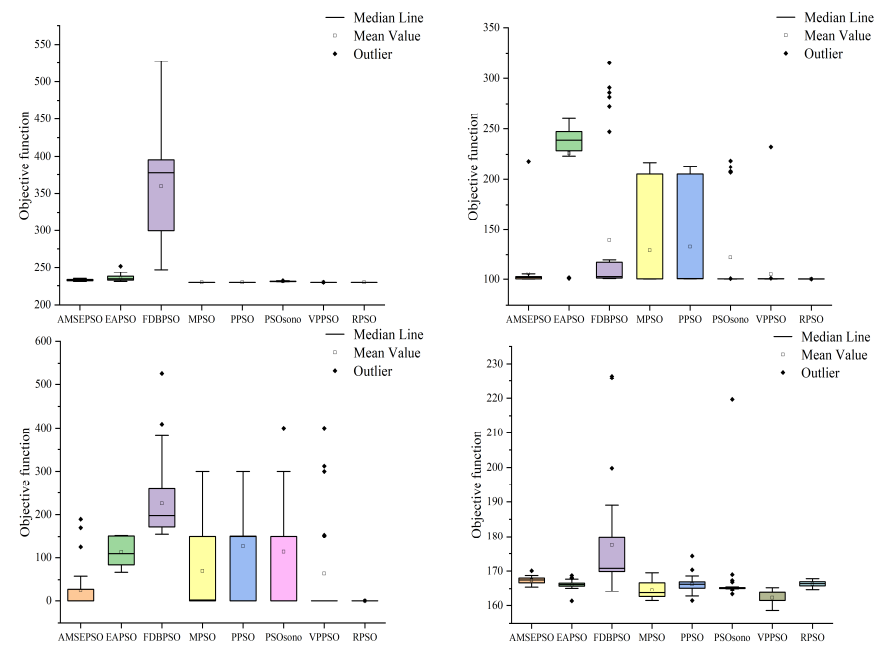


Figure 5. Distribution of solutions for each algorithm in 10 dimensions.

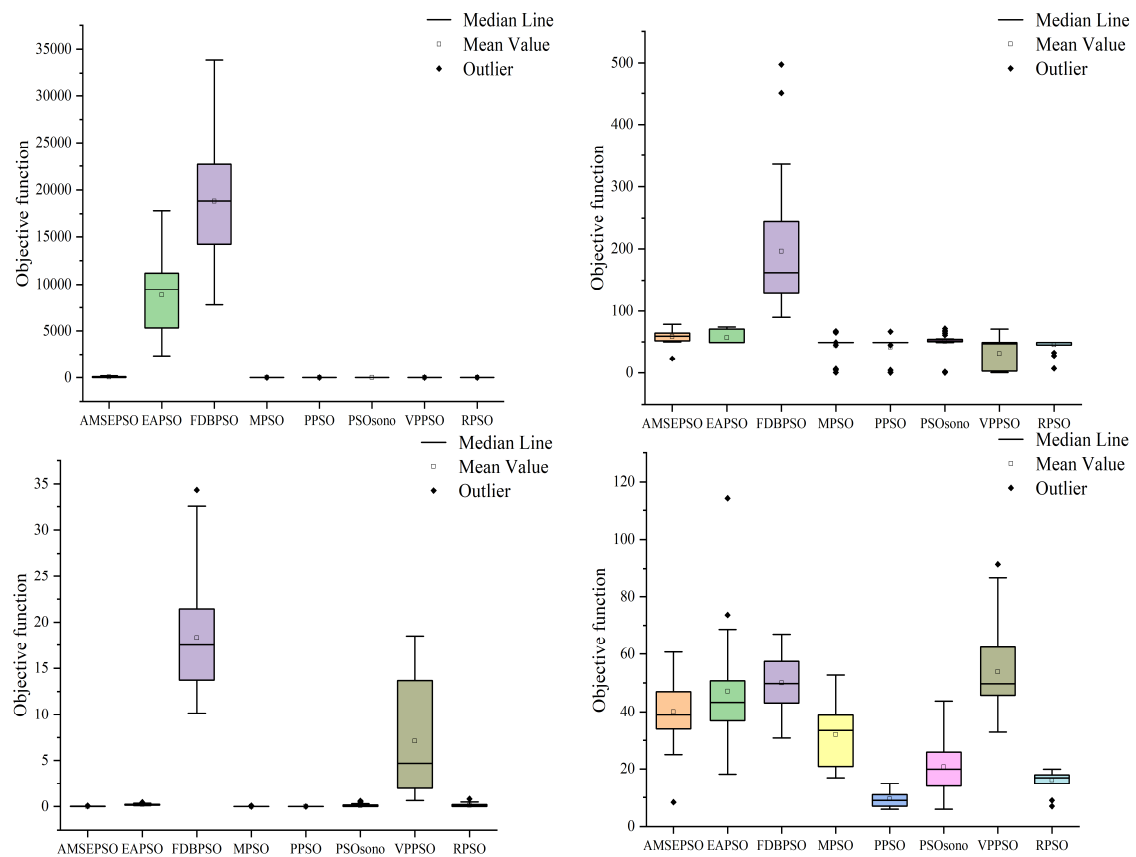


Figure 6. Cont.

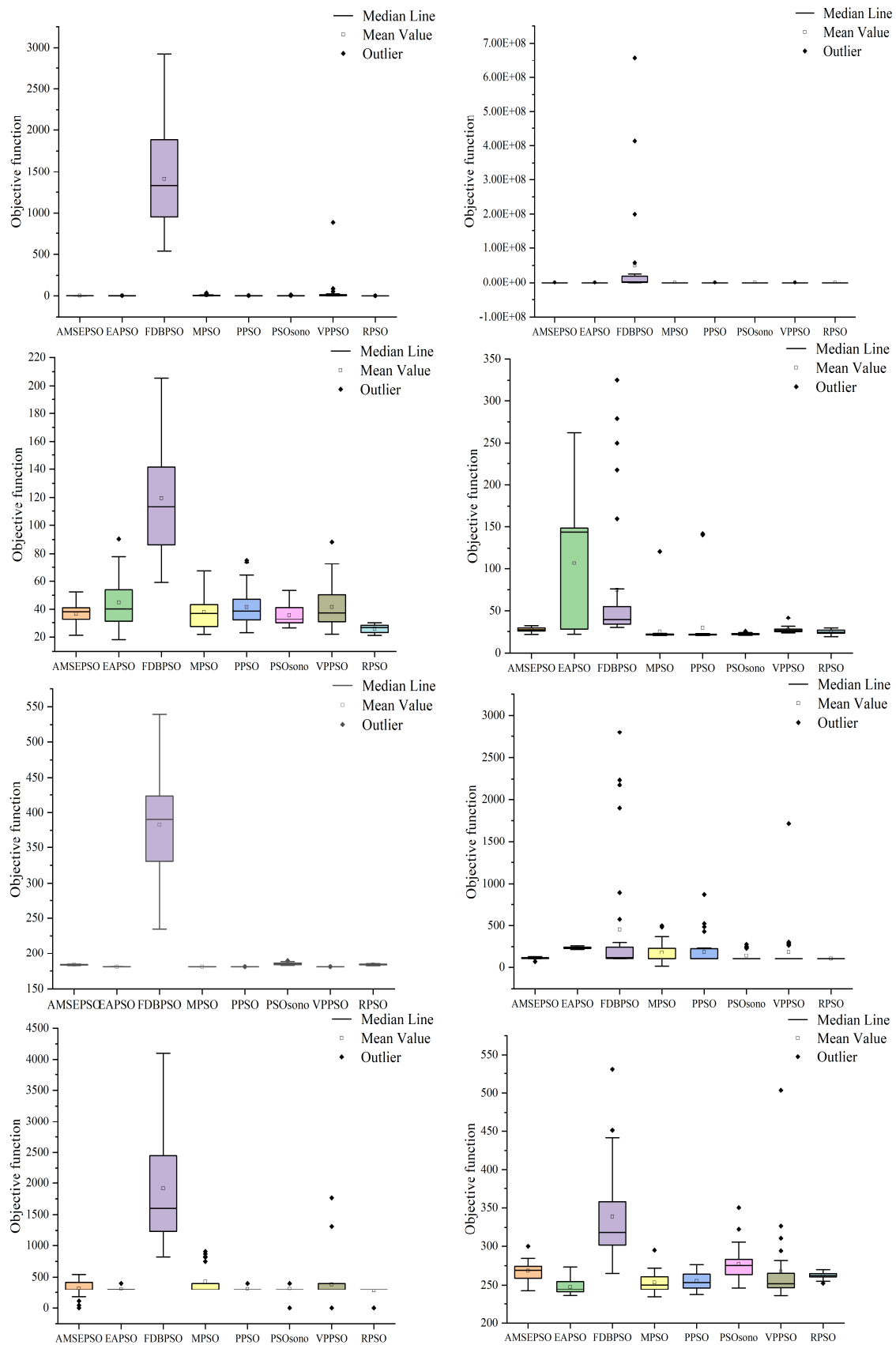


Figure 6. Distribution of solutions for each algorithm in 20 dimensions.

The best and worst values of the PSO algorithm variant are compared with the best and worst values of the RPSO algorithm, and the number of times that the best and worst values of the PSO algorithm variant is greater than those of the RPSO algorithm in 12 functions is counted. The RPSO algorithm beats the FDBPSO algorithm 12 times, the EAPSO algorithm 9 times, the VPPSO and AMSEPSO algorithms 8 times, the PPSO algorithm 5 times, and the MPSO and PSOsosono algorithms 4 times regarding the optimal value. Regarding the worst value, the RPSO algorithm beats the FDBPSO algorithm 12 times, the EAPSO and VPPSO algorithms 11 times, the AMSEPSO and PPSO algorithms 10 times, and the MPSO and PSOsosono algorithms 9 times and 8 times, respectively. The above information shows that the RPSO algorithm beats the AMSEPSO, EAPSO, FDBPSO and VPPSO algorithms many times in terms of the optimal value and the worst value. The RPSO algorithm is clearly superior to the four algorithms mentioned above. While the advantages of the MPSO, PPSO, and PSOsosono algorithms may not be obvious in terms of optimal value, they demonstrate significant advantages in terms of worst value. Thus, the proposed algorithm's superiority over the variants of the PSO algorithm is confirmed. The comparison values mentioned above are similar to the comparison values between the RPSO algorithm and the variants of the PSO algorithm in the case of ten dimensions, indicating the strong robustness of the RPSO algorithm.

4.2. Convergence Analysis

The convergence effect is also an important assessment index for measuring the optimization ability of the algorithm, and the results of the iterative convergence can show the responsiveness of the algorithm in different optimization-seeking environments, which further reflects the learning and adaptability of the algorithm. In this section, the convergence process for the above 30 times is averaged to obtain values, and the average convergence of each algorithm in each function is shown in Figures 7 and 8.

As presented in Figures 7 and 8, the convergence effect of RPSO in both dimensions has some advantages; furthermore, in ten dimensions, the convergence effect of RPSO on F(1–2), F(5–6), F8, and F(10–11) displays a reliable ability to seek optima compared with other algorithms, especially F(2), F(5–6), F(8), and F(11), where the convergence speed of RPSO is more significant. In 20 dimensions, the convergence of RPSO is better in the functions F(1), F(5–7), and F(9–11), particularly in the functions F(5–7), F(10–11), where the convergence speeds of RPSO display a significant advantage. The optimization search effect of RPSO still demonstrates better convergence performance in different dimensions, which verifies the learning and adaptability of RPSO.

4.3. Comparison with the Top Algorithm

This subsection aims to reveal the competitive edge of RPSO by comparing it with top algorithms from different years of the CEC competition, including the gaining-sharing knowledge-based algorithm with adaptive parameters hybrid with IMODE (APGSK-IMODE) [62], Eigen crossover in cooperative model of evolutionary algorithms (EA4eig) [63], Improved multi-operator differential evolution (IMODE) [64], Population's variance-based adaptive differential evolution (PVADE) [65], and the adaptive gaining-sharing knowledge-based algorithm (AGSK) [66]. EA4eig was the champion algorithm in CEC 2022, APGSK-IMODE ranked fourth in CEC 2021, IMODE was the champion in CEC 2020, AGSK was the runner-up in CEC 2020, and PVADE was a recognized algorithm in the competition. The experimental parameters for each algorithm are set consistently as mentioned earlier, and the optimization results are presented in Tables 3 and 4.

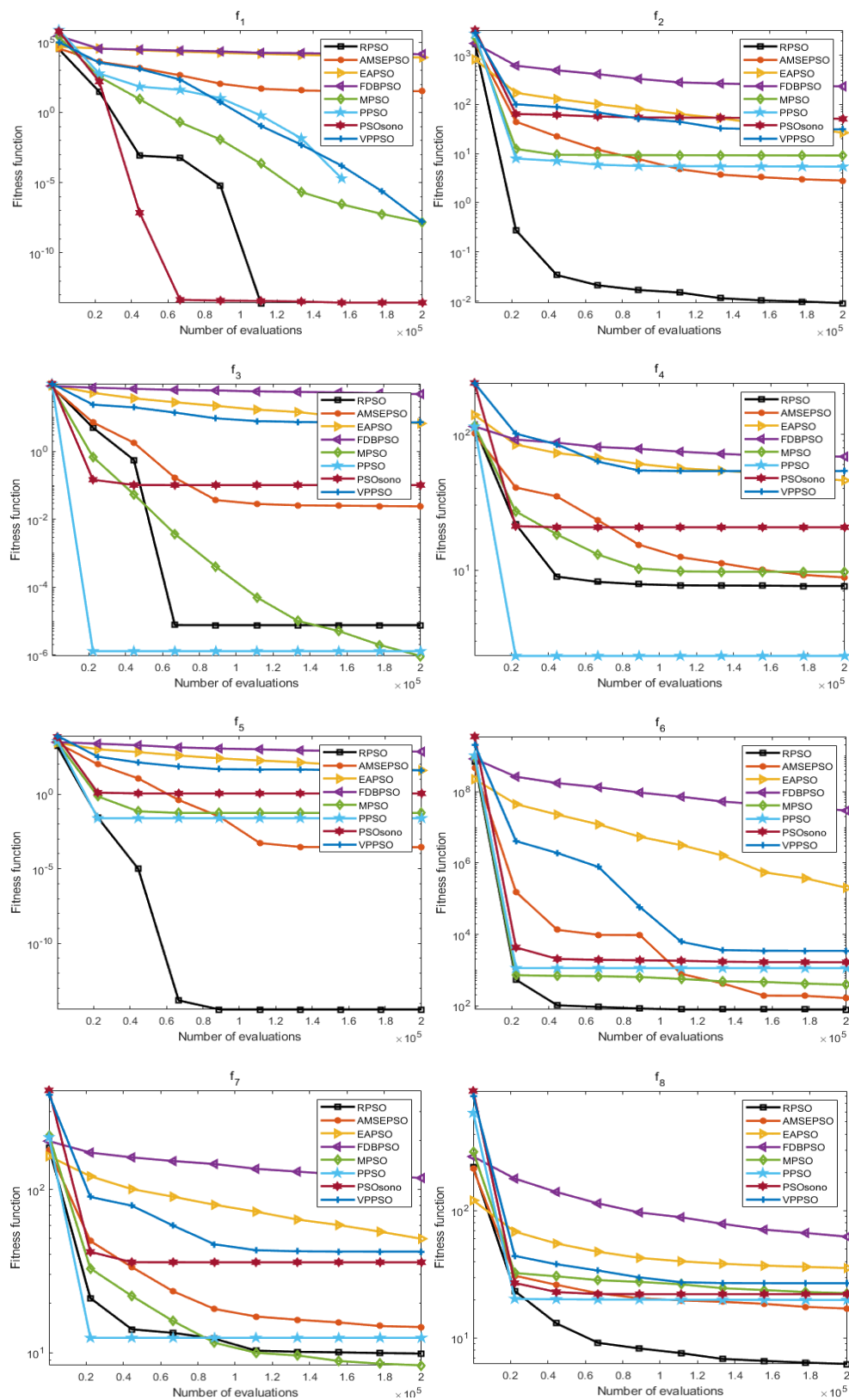


Figure 7. Cont.

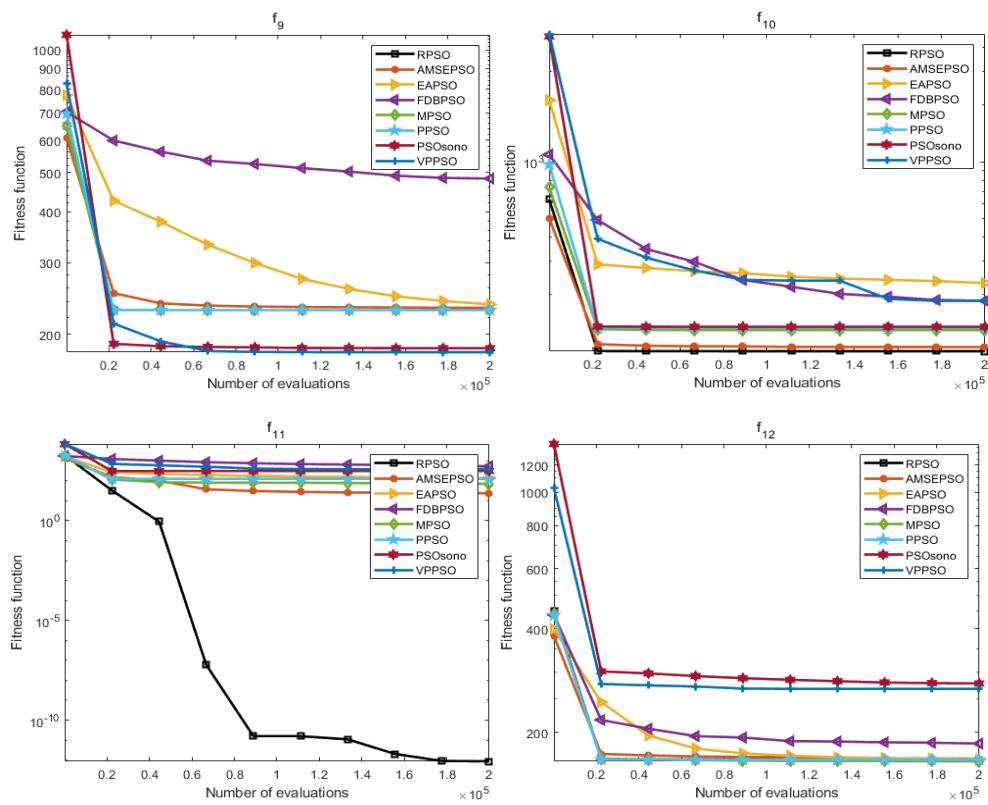


Figure 7. Plot of average convergence effect of each algorithm (dim = 10).

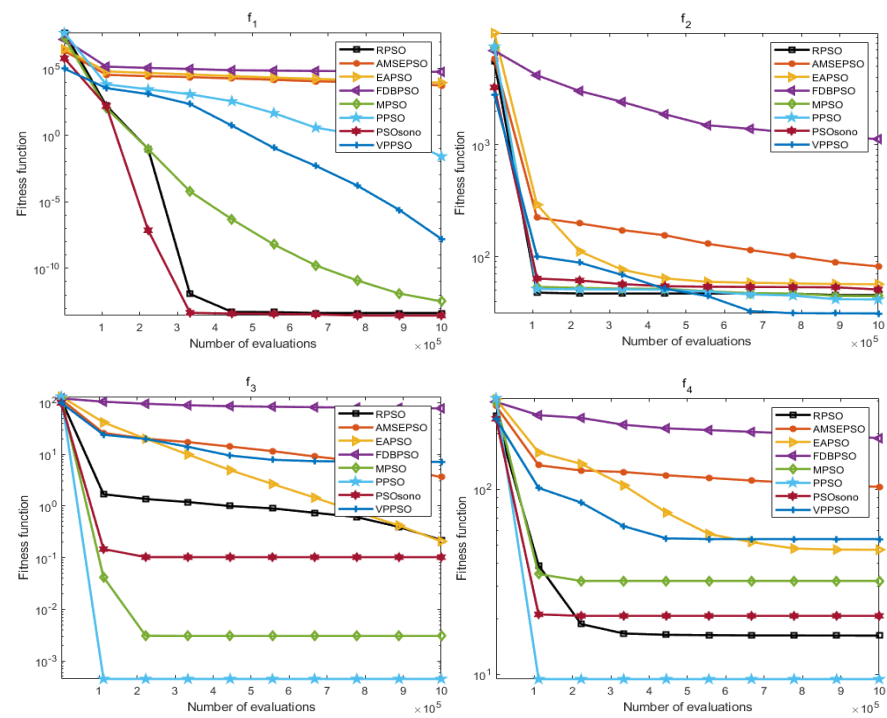


Figure 8. Cont.

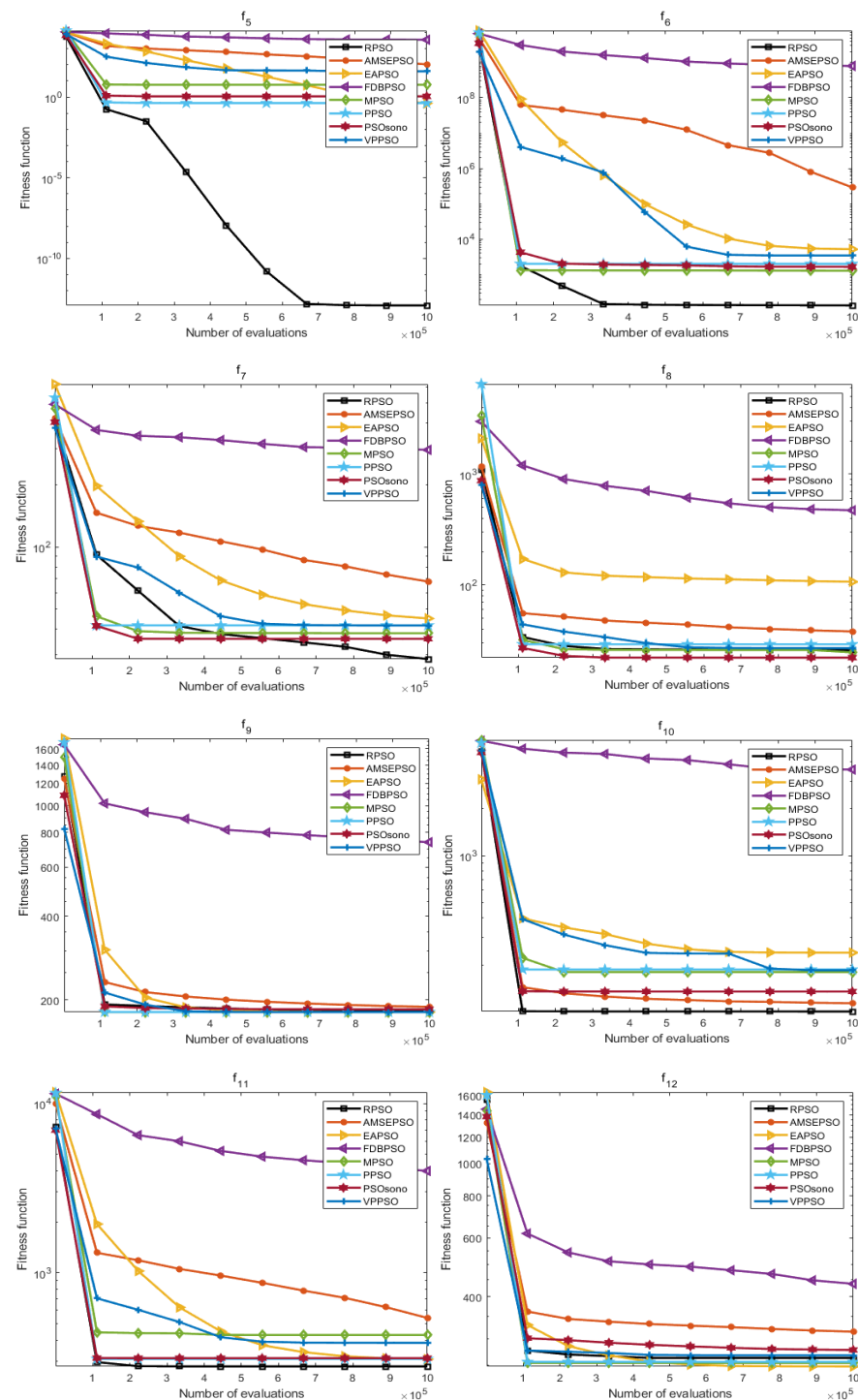


Figure 8. Plot of average convergence effect of each algorithm (dim = 20).

Analyzing Table 3, when the RPSO algorithm was compared with the five most advanced algorithms in ten dimensions, it can be observed that RPSO does not always achieve the best results across the 12 functions when compared with these algorithms. However, the optimization standard deviation of RPSO is the lowest in F(2), F(5), F(9), F(10), and F(11), indicating a strong search stability. Moreover, RPSO outperforms the EA4eig algorithm on the F(10) function, the IMODE algorithm on the F(4) function, the PVADE algorithm on the F(8) and F(10) functions, and the AGSK algorithm on the F(4) and F(10) functions in terms of the optimal value. RPSO also outperforms the APGSK-IMODE algorithm on the F(10) function; the EA4eig algorithm on the F(2) function; the IMODE

algorithm on the F(4), F(5), F(10), and F(11) functions; and the PVADE algorithm on the F(2), F(8), and F(11) functions. Additionally, RPSO outperforms the AGSK algorithm on the F(2), F(4), F(5), and F(11) functions.

Table 3. Ten dimensions with other algorithms.

F	Index	APGSK–IMODE	EA4eig	IMODE	PVADE	AGSK	RPSO
F1(x)	Best	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Worst	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Median	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Mean	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Standard	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
F2(x)	Contest	N/A (=)	N/A (=)	N/A (=)	N/A (=)	N/A (=)	
	Best	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	1.71×10^{-07}
	Worst	0.00×10^{00}	3.99×10^{00}	0.00×10^{00}	8.92×10^{00}	8.92×10^{00}	7.32×10^{-02}
	Median	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	8.92×10^{00}	3.99×10^{00}	5.91×10^{-04}
	Mean	0.00×10^{00}	6.64×10^{-11}	0.00×10^{00}	5.55×10^{00}	3.09×10^{00}	9.12×10^{-03}
F3(x)	Standard	0.00×10^{00}	1.51×10^{00}	0.00×10^{00}	3.91×10^{00}	2.09×10^{00}	1.80×10^{-02}
	Contest	1.21×10^{-12} (−)	4.70×10^{-06} (+)	1.21×10^{-12} (−)	1.74×10^{-03} (+)	1.53×10^{-03} (+)	
	Best	0.00×10^{00}	0.00×10^{00}	2.69×10^{-07}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Worst	0.00×10^{00}	0.00×10^{00}	4.86×10^{-05}	4.85×10^{-05}	2.08×10^{-05}	8.79×10^{-05}
	Median	0.00×10^{00}	0.00×10^{00}	1.73×10^{-06}	0.00×10^{00}	1.42×10^{-06}	1.42×10^{-06}
F4(X)	Mean	0.00×10^{00}	0.00×10^{00}	5.93×10^{-06}	1.93×10^{-06}	2.75×10^{-06}	7.62×10^{-06}
	Standard	0.00×10^{00}	0.00×10^{00}	9.55×10^{-06}	8.84×10^{-06}	4.75×10^{-06}	1.90×10^{-05}
	Contest	9.65×10^{-11} (−)	5.63×10^{-11} (−)	8.75×10^{-02} (=)	1.98×10^{-06} (−)	1.62×10^{-11} (=)	
	Best	9.96×10^{-11}	0.00×10^{00}	4.97×10^{00}	2.72×10^{00}	3.02×10^{00}	2.98×10^{00}
	Worst	6.96×10^{00}	2.98×10^{00}	1.29×10^{01}	8.68×10^{00}	1.13×10^{01}	9.95×10^{00}
F5(X)	Median	4.97×10^{00}	9.95×10^{-11}	8.95×10^{00}	5.48×10^{00}	7.97×10^{00}	8.46×10^{00}
	Mean	4.61×10^{00}	9.62×10^{-11}	9.02×10^{00}	5.45×10^{00}	7.63×10^{00}	7.63×10^{00}
	Standard	1.21×10^{00}	9.23×10^{-11}	2.20×10^{00}	1.65×10^{00}	2.14×10^{00}	2.08×10^{00}
	Contest	7.86×10^{-06} (−)	2.39×10^{-11} (−)	2.06×10^{-02} (+)	9.69×10^{-05} (−)	9.06×10^{-11} (=)	
	Best	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
F6(X)	Worst	0.00×10^{00}	0.00×10^{00}	1.68×10^{01}	0.00×10^{00}	1.14×10^{-13}	0.00×10^{00}
	Median	0.00×10^{00}	0.00×10^{00}	6.33×10^{-11}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Mean	0.00×10^{00}	0.00×10^{00}	1.83×10^{00}	0.00×10^{00}	7.58×10^{-15}	0.00×10^{00}
	Standard	0.00×10^{00}	0.00×10^{00}	3.20×10^{00}	0.00×10^{00}	2.88×10^{-14}	0.00×10^{00}
	Contest	N/A (=)	N/A (=)	6.93×10^{-12} (+)	N/A (=)	5.70×10^{-11} (=)	
F7(X)	Best	6.55×10^{-02}	1.63×10^{-03}	9.90×10^{-02}	2.67×10^{-02}	5.02×10^{-03}	2.70×10^{00}
	Worst	4.49×10^{-11}	4.28×10^{-11}	4.99×10^{00}	1.70×10^{01}	4.75×10^{-11}	1.96×10^{02}
	Median	1.86×10^{-11}	1.85×10^{-02}	1.18×10^{00}	3.72×10^{-11}	2.47×10^{-11}	6.32×10^{01}
	Mean	2.30×10^{-11}	5.70×10^{-02}	1.65×10^{00}	1.48×10^{00}	2.50×10^{-11}	7.83×10^{01}
	Standard	1.17×10^{-11}	9.46×10^{-02}	1.29×10^{00}	3.57×10^{00}	1.32×10^{-11}	6.02×10^{01}
F8(X)	Contest	3.01×10^{-11} (−)	3.02×10^{-11} (−)	4.98×10^{-11} (−)	6.70×10^{-11} (−)	3.02×10^{-11} (−)	
	Best	0.00×10^{00}	0.00×10^{00}	2.65×10^{-05}	0.00×10^{00}	0.00×10^{00}	1.96×10^{-11}
	Worst	1.94×10^{-07}	3.76×10^{-9}	1.24×10^{-03}	2.10×10^{01}	2.09×10^{-10}	2.26×10^{01}
	Median	0.00×10^{00}	5.68×10^{-13}	1.71×10^{-04}	3.12×10^{-11}	4.55×10^{-13}	5.20×10^{00}
	Mean	6.47×10^{-9}	2.77×10^{-10}	3.05×10^{-04}	3.76×10^{00}	1.83×10^{-11}	9.88×10^{00}
F9(X)	Standard	3.54×10^{-8}	8.53×10^{-10}	3.52×10^{-04}	7.67×10^{00}	5.06×10^{-11}	9.20×10^{00}
	Contest	1.65×10^{-11} (−)	2.71×10^{-11} (−)	3.02×10^{-11} (−)	7.49×10^{-06} (−)	2.70×10^{-11} (−)	
	Best	2.17×10^{-02}	1.39×10^{-03}	4.30×10^{-02}	7.44×10^{-11}	1.19×10^{-11}	2.21×10^{-11}
	Worst	2.64×10^{00}	8.21×10^{-11}	8.09×10^{00}	2.16×10^{01}	1.61×10^{00}	2.10×10^{01}
	Median	1.99×10^{-11}	3.37×10^{-11}	1.42×10^{00}	1.17×10^{00}	6.47×10^{-11}	4.36×10^{00}
F10(X)	Mean	3.68×10^{-11}	3.24×10^{-11}	2.72×10^{00}	4.35×10^{00}	7.84×10^{-11}	6.25×10^{00}
	Standard	5.11×10^{-11}	2.21×10^{-11}	2.34×10^{00}	7.15×10^{00}	4.37×10^{-11}	6.40×10^{00}
	Contest	2.66×10^{-9} (−)	2.44×10^{-9} (−)	2.32×10^{-02} (−)	4.06×10^{-02} (−)	1.03×10^{-06} (−)	
	Best	2.29×10^{02}	1.86×10^{02}	1.37×10^{-06}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}
	Worst	2.29×10^{02}	1.86×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}
F11(X)	Median	2.29×10^{02}	1.86×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}
	Mean	2.29×10^{02}	1.86×10^{02}	2.14×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}
	Standard	0.00×10^{00}	4.22×10^{-13}	5.82×10^{01}	0.00×10^{00}	8.67×10^{-14}	0.00×10^{00}
	Contest	N/A (=)	4.16×10^{-14} (−)	7.64×10^{-9} (−)	1.69×10^{-14} (−)	N/A (=)	
	Best	6.72×10^{01}	1.00×10^{02}	3.75×10^{00}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}
F12(X)	Worst	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}
	Median	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}
	Mean	9.91×10^{01}	1.00×10^{02}	8.35×10^{01}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}
	Standard	6.02×10^{00}	3.69×10^{-02}	3.45×10^{01}	3.82×10^{-02}	3.49×10^{-02}	5.59×10^{-02}
	Contest	9.47×10^{-11} (=)	1.24×10^{-03} (−)	1.63×10^{-02} (−)	6.10×10^{-03} (−)	4.20×10^{-11} (=)	
F11(X)	Best	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Worst	4.55×10^{-13}	0.00×10^{00}	2.61×10^{-07}	3.00×10^{02}	4.55×10^{-13}	0.00×10^{00}
	Median	0.00×10^{00}	0.00×10^{00}	2.96×10^{-8}	0.00×10^{00}	4.55×10^{-13}	0.00×10^{00}
	Mean	1.36×10^{-13}	0.00×10^{00}	5.19×10^{-8}	1.00×10^{01}	3.49×10^{-13}	0.00×10^{00}
	Standard	2.12×10^{-13}	0.00×10^{00}	6.11×10^{-8}	5.48×10^{01}	1.96×10^{-13}	0.00×10^{00}
F12(X)	Contest	5.19×10^{-02} (=)	N/A (=)	6.92×10^{-8} (+)	4.76×10^{-05} (+)	1.56×10^{-11} (=)	
	Best	1.59×10^{02}	1.45×10^{02}	1.63×10^{02}	1.59×10^{02}	1.59×10^{02}	1.65×10^{02}
	Worst	1.63×10^{02}	1.59×10^{02}	1.65×10^{02}	1.65×10^{02}	1.61×10^{02}	1.68×10^{02}
	Median	1.62×10^{02}	1.46×10^{02}	1.64×10^{02}	1.63×10^{02}	1.60×10^{02}	1.66×10^{02}
	Mean	1.61×10^{02}	1.47×10^{02}	1.64×10^{02}	1.63×10^{02}	1.60×10^{02}	1.66×10^{02}
F12(X)	Standard	1.54×10^{00}	3.55×10^{00}	6.54×10^{-11}	2.08×10^{00}	9.91×10^{-11}	8.15×10^{-11}
	Contest	2.52×10^{-11} (−)	3.02×10^{-11} (−)	4.08×10^{-11} (−)	9.36×10^{-11} (−)	2.56×10^{-11} (−)	
	+/= / −	2000/5/7	2001/3/8	2003/2/7	2002/2/8	2001/7/4	
	Rank	2.54	1.96	4.00	4.12	3.83	4.50

Table 4. Twenty dimensions with other algorithms.

F	Index	APGSK-IMODE	EA4eig	IMODE	PVADE	AGSK	RPSO
F1(x)	Best	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Worst	5.68×10^{-14}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	5.68×10^{-14}	0.00×10^{00}
	Median	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	Mean	7.58×10^{-15}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	1.89×10^{-14}	0.00×10^{00}
	Standard	1.97×10^{-14}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	2.73×10^{-14}	0.00×10^{00}
F2(x)	Contest	N/A (=)	N/A (=)	N/A (=)	N/A (=)	N/A (=)	N/A (=)
	Best	3.63×10^{-06}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	6.67×10^{00}
	Worst	4.91×10^{01}	3.99×10^{00}	4.91×10^{01}	8.92×10^{00}	4.91×10^{01}	4.91×10^{01}
	Median	4.91×10^{01}	0.00×10^{00}	4.91×10^{01}	3.99×10^{00}	4.91×10^{01}	4.91×10^{01}
	Mean	4.47×10^{01}	7.97×10^{-11}	3.45×10^{01}	5.36×10^{00}	2.92×10^{01}	4.54×10^{01}
F3(x)	Standard	1.18×10^{01}	1.62×10^{00}	2.27×10^{01}	3.69×10^{00}	2.42×10^{01}	8.82×10^{00}
	Contest	1.20×10^{-03} (−)	6.27×10^{-12} (−)	3.14×10^{-11} (=)	8.20×10^{-11} (−)	2.95×10^{-05} (−)	3.49×10^{-03}
	Best	0.00×10^{00}	0.00×10^{00}	2.11×10^{-11}	0.00×10^{00}	0.00×10^{00}	4.56×10^{00}
	Worst	1.14×10^{-13}	1.14×10^{-13}	3.49×10^{00}	1.42×10^{-06}	8.36×10^{-11}	1.61×10^{00}
	Median	1.14×10^{-13}	1.14×10^{-13}	1.03×10^{00}	0.00×10^{00}	3.41×10^{-13}	1.75×10^{00}
F4(X)	Mean	1.06×10^{-13}	7.96×10^{-14}	1.08×10^{00}	7.11×10^{-8}	3.56×10^{-12}	1.30×10^{00}
	Standard	2.88×10^{-14}	5.30×10^{-14}	6.96×10^{-11}	2.70×10^{-07}	1.52×10^{-11}	6.96×10^{00}
	Contest	2.36×10^{-12} (−)	1.01×10^{-11} (−)	2.37×10^{-10} (−)	3.16×10^{-12} (−)	2.89×10^{-11} (−)	1.99×10^{01}
	Best	1.29×10^{01}	3.98×10^{00}	3.68×10^{01}	0.00×10^{00}	1.98×10^{01}	1.69×10^{01}
	Worst	2.98×10^{01}	1.69×10^{01}	8.36×10^{01}	5.97×10^{00}	4.58×10^{01}	1.62×10^{01}
F5(X)	Median	2.19×10^{01}	8.95×10^{00}	5.62×10^{01}	1.99×10^{00}	3.72×10^{01}	3.27×10^{00}
	Mean	2.23×10^{01}	9.22×10^{00}	5.86×10^{01}	2.46×10^{00}	3.63×10^{01}	6.85×10^{00}
	Standard	4.51×10^{00}	3.49×10^{00}	1.23×10^{01}	1.32×10^{00}	6.85×10^{00}	3.56×10^{-11} (+)
	Contest	2.58×10^{-06} (+)	4.59×10^{-8} (−)	2.92×10^{-11} (+)	2.58×10^{-11} (−)	3.56×10^{-11} (+)	1.14×10^{-13}
	Best	0.00×10^{00}	0.00×10^{00}	2.61×10^{02}	0.00×10^{00}	1.14×10^{-13}	0.00×10^{00}
F6(X)	Worst	0.00×10^{00}	0.00×10^{00}	1.03×10^{03}	0.00×10^{00}	5.44×10^{-11}	0.00×10^{00}
	Median	0.00×10^{00}	0.00×10^{00}	6.72×10^{02}	0.00×10^{00}	8.95×10^{-02}	0.00×10^{00}
	Mean	0.00×10^{00}	0.00×10^{00}	6.82×10^{02}	0.00×10^{00}	7.48×10^{-02}	0.00×10^{00}
	Standard	0.00×10^{00}	0.00×10^{00}	2.08×10^{02}	0.00×10^{00}	1.11×10^{-11}	0.00×10^{00}
	Contest	N/A (=)	N/A (=)	1.40×10^{-11} (+)	N/A (=)	1.53×10^{-9} (+)	2.18×10^{-11}
F7(X)	Best	1.04×10^{00}	2.18×10^{-02}	9.17×10^{00}	2.20×10^{-02}	2.18×10^{-11}	5.55×10^{01}
	Worst	2.48×10^{01}	2.08×10^{00}	4.54×10^{01}	1.12×10^{00}	1.90×10^{00}	2.47×10^{02}
	Median	6.91×10^{00}	1.20×10^{-11}	2.19×10^{01}	1.25×10^{-11}	3.95×10^{-11}	1.26×10^{02}
	Mean	9.11×10^{00}	2.13×10^{-11}	2.36×10^{01}	3.23×10^{-11}	6.36×10^{-11}	1.30×10^{02}
	Standard	6.77×10^{00}	3.69×10^{-11}	8.37×10^{00}	3.73×10^{-11}	5.27×10^{-11}	5.08×10^{01}
F8(X)	Contest	3.01×10^{-11} (−)	3.02×10^{-11} (−)	3.02×10^{-11} (−)	3.02×10^{-11} (−)	3.02×10^{-11} (−)	3.12×10^{-11}
	Best	2.51×10^{-02}	0.00×10^{00}	2.51×10^{01}	0.00×10^{00}	3.12×10^{-11}	2.11×10^{01}
	Worst	2.10×10^{01}	2.10×10^{01}	4.48×10^{01}	2.10×10^{01}	2.25×10^{01}	3.00×10^{01}
	Median	3.86×10^{00}	1.80×10^{00}	3.36×10^{01}	0.00×10^{00}	2.11×10^{01}	2.66×10^{01}
	Mean	7.00×10^{00}	4.23×10^{00}	3.36×10^{01}	3.68×10^{00}	1.82×10^{01}	2.58×10^{01}
F9(X)	Standard	7.08×10^{00}	6.18×10^{00}	5.16×10^{00}	7.62×10^{00}	6.09×10^{00}	2.75×10^{00}
	Contest	3.01×10^{-11} (−)	3.00×10^{-11} (−)	2.39×10^{-8} (−)	1.78×10^{-11} (−)	8.89×10^{-10} (−)	1.61×10^{01}
	Best	1.41×10^{01}	2.69×10^{-11}	2.08×10^{01}	5.64×10^{-05}	1.61×10^{01}	1.90×10^{01}
	Worst	2.09×10^{01}	2.07×10^{01}	2.33×10^{01}	2.04×10^{01}	2.29×10^{01}	2.91×10^{01}
	Median	2.04×10^{01}	2.02×10^{01}	2.18×10^{01}	1.06×10^{-11}	2.21×10^{01}	2.40×10^{01}
F10(X)	Mean	2.01×10^{01}	1.77×10^{01}	2.19×10^{01}	1.69×10^{00}	2.20×10^{01}	2.46×10^{01}
	Standard	1.44×10^{00}	6.14×10^{00}	6.63×10^{-11}	5.16×10^{00}	1.16×10^{00}	2.60×10^{00}
	Contest	4.61×10^{-10} (−)	3.16×10^{-10} (−)	3.83×10^{-06} (−)	3.69×10^{-11} (−)	1.02×10^{-05} (−)	1.81×10^{02}
	Best	1.81×10^{02}	1.65×10^{02}	1.81×10^{02}	2.29×10^{02}	1.81×10^{02}	1.85×10^{02}
	Worst	1.81×10^{02}	1.65×10^{02}	1.81×10^{02}	2.29×10^{02}	1.81×10^{02}	1.84×10^{02}
F11(X)	Median	1.81×10^{02}	1.65×10^{02}	1.81×10^{02}	2.29×10^{02}	1.81×10^{02}	1.84×10^{02}
	Mean	1.81×10^{02}	1.65×10^{02}	1.81×10^{02}	2.29×10^{02}	1.81×10^{02}	1.84×10^{02}
	Standard	8.67×10^{-14}	0.00×10^{00}	1.18×10^{-07}	0.00×10^{00}	8.67×10^{-14}	7.37×10^{-11}
	Contest	1.21×10^{-12} (−)	1.21×10^{-12} (−)	3.02×10^{-11} (−)	1.21×10^{-12} (+)	1.21×10^{-12} (−)	1.00×10^{02}
	Best	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.01×10^{02}
F12(X)	Worst	1.00×10^{02}	2.24×10^{02}	1.01×10^{02}	2.00×10^{02}	1.00×10^{02}	1.00×10^{02}
	Median	1.00×10^{02}	1.00×10^{02}	1.01×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}
	Mean	1.00×10^{02}	1.08×10^{02}	1.01×10^{02}	1.03×10^{02}	1.00×10^{02}	1.00×10^{02}
	Standard	3.76×10^{-02}	3.06×10^{01}	7.66×10^{-02}	1.82×10^{01}	2.85×10^{-02}	8.67×10^{-02}
	Contest	9.21×10^{-05} (−)	1.01×10^{-8} (−)	6.53×10^{-8} (+)	5.57×10^{-10} (−)	8.99×10^{-11} (−)	0.00×10^{00}
F13(X)	Best	0.00×10^{00}	3.00×10^{02}	3.35×10^{-04}	0.00×10^{00}	3.00×10^{02}	3.00×10^{02}
	Worst	3.00×10^{02}	4.00×10^{02}	3.22×10^{02}	1.50×10^{02}	4.00×10^{02}	3.00×10^{02}
	Median	3.00×10^{02}	3.00×10^{02}	3.00×10^{02}	0.00×10^{00}	4.00×10^{02}	3.00×10^{02}
	Mean	2.50×10^{02}	3.23×10^{02}	2.81×10^{02}	5.01×10^{00}	3.87×10^{02}	2.80×10^{02}
	Standard	1.14×10^{02}	4.30×10^{01}	7.64×10^{01}	2.75×10^{01}	3.46×10^{01}	7.61×10^{01}
F14(X)	Contest	1.08×10^{-11} (=)	2.18×10^{-02} (+)	1.28×10^{-9} (+)	6.71×10^{-13} (−)	8.54×10^{-12} (+)	2.31×10^{02}
	Best	2.31×10^{02}	1.89×10^{02}	2.40×10^{02}	1.60×10^{02}	2.39×10^{02}	2.70×10^{02}
	Worst	2.37×10^{02}	2.00×10^{02}	2.67×10^{02}	1.65×10^{02}	2.39×10^{02}	2.62×10^{02}
	Median	2.33×10^{02}	2.00×10^{02}	2.53×10^{02}	1.64×10^{02}	2.34×10^{02}	2.62×10^{02}
	Mean	2.34×10^{02}	2.00×10^{02}	2.53×10^{02}	1.63×10^{02}	2.34×10^{02}	2.62×10^{02}
F15(X)	Standard	1.52×10^{00}	2.05×10^{00}	7.63×10^{00}	1.51×10^{00}	2.07×10^{00}	3.94×10^{00}
	Contest	3.02×10^{-11} (−)	3.02×10^{-11} (−)	7.74×10^{-06} (−)	2.89×10^{-11} (−)	2.97×10^{-11} (−)	$2001/3/8$
	Rank	2001/3/8	2001/2/9	2004/2/6	2001/2/9	2003/1/8	4.50
		3.21	2.33	4.54	2.42	4.00	

Examining Table 4, when comparing the RPSO algorithm with the top algorithms in 20 dimensions, it can be observed that RPSO does not always achieve the best results across the 12 functions when compared with these algorithms. However, RPSO exhibits better

performance compared to the 10-dimensional case. The optimization standard deviation is the lowest in F(1), F(5), and F(7), indicating strong search stability. Additionally, RPSO outperforms the APGSK-IMODE algorithm on the F(1), F(5), and F(11) functions; the EA4eig algorithm on the F(11) function; and the IMODE algorithm on the F(3), F(4), F(5), F(7), F(8), F(10), and F(11) functions in terms of the optimal value. RPSO also outperforms the AGSK algorithm on the F(4), F(5), and F(11) functions. In terms of the worst value, RPSO outperforms the APGSK-IMODE algorithm on the F(1) and F(4) functions; the EA4eig algorithm on the F(10) and F(11) functions; and the IMODE algorithm on the F(4), F(5), F(7), F(10), and F(11) functions. RPSO also outperforms the PVADE algorithm on the F(9) and F(10) functions; and the AGSK algorithm on the F(1), F(4), F(5), and F(11) functions.

4.4. Ablation Experiments

In order to further confirm the effectiveness of strategy fusion, this subsection conducts ablation experiments one by one on the mechanisms proposed in this paper. The PSO that only incorporates random exchange learning is set as PSO-1, the PSO algorithm that only incorporates QI is set as PSO-2, and the PSO algorithm that fuses the two mechanisms and does not use the selection mechanism is named PSO-3. It is worth mentioning that the parameter settings of RPSO are derived from previous research by other scholars, and thus are not included in the scope of validation. The other experimental parameters are consistent with those described, and the specific optimization results are shown in Table 5.

Table 5. Data from ablation experiments.

F	Index	Best	Worst	Median	Mean	Standard
F1(x)	PSO-1	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	PSO-2	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	PSO-3	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
	RPSO	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
F2(x)	PSO-1	0.00×10^{00}	1.08×10^{01}	3.99×10^{00}	5.15×10^{00}	4.29×10^{00}
	PSO-2	0.00×10^{00}	1.31×10^{01}	6.33×10^{00}	5.97×10^{00}	4.09×10^{00}
	PSO-3	0.00×10^{00}	1.04×10^{01}	3.99×10^{00}	3.46×10^{00}	3.53×10^{00}
	RPSO	1.71×10^{-07}	7.32×10^{-02}	5.91×10^{-04}	9.12×10^{-03}	1.80×10^{-02}
F3(x)	PSO-1	0.00×10^{00}	2.31×10^{-02}	1.26×10^{-04}	2.80×10^{-03}	6.28×10^{-03}
	PSO-2	1.06×10^{-10}	7.47×10^{-02}	6.67×10^{-04}	9.25×10^{-03}	1.89×10^{-02}
	PSO-3	0.00×10^{00}	1.91×10^{-02}	9.10×10^{-05}	3.05×10^{-03}	5.55×10^{-03}
	RPSO	0.00×10^{00}	8.79×10^{-05}	1.42×10^{-06}	7.62×10^{-06}	1.90×10^{-05}
F4(X)	PSO-1	2.98×10^{00}	2.09×10^{01}	6.96×10^{00}	8.76×10^{00}	4.74×10^{00}
	PSO-2	9.95×10^{-01}	1.49×10^{01}	5.97×10^{00}	7.23×10^{00}	3.74×10^{00}
	PSO-3	9.95×10^{-01}	1.69×10^{01}	7.46×10^{00}	8.13×10^{00}	4.27×10^{00}
	RPSO	2.98×10^{00}	9.95×10^{00}	8.46×10^{00}	7.63×10^{00}	2.08×10^{00}
F5(X)	PSO-1	0.00×10^{00}	4.54×10^{-01}	0.00×10^{00}	2.11×10^{-02}	8.49×10^{-02}
	PSO-2	0.00×10^{00}	8.95×10^{-02}	0.00×10^{00}	2.98×10^{-03}	1.63×10^{-02}
	PSO-3	0.00×10^{00}	1.79×10^{-01}	0.00×10^{00}	2.09×10^{-02}	4.51×10^{-02}
	RPSO	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
F6(X)	PSO-1	2.95×10^{00}	1.72×10^{-03}	4.73×10^{01}	2.26×10^{02}	4.03×10^{02}
	PSO-2	1.28×10^{00}	2.55×10^{-03}	3.93×10^{01}	2.10×10^{02}	4.89×10^{02}
	PSO-3	1.52×10^{00}	1.45×10^{-03}	4.53×10^{01}	1.90×10^{02}	3.35×10^{02}
	RPSO	2.70×10^{00}	1.96×10^{02}	6.32×10^{01}	7.83×10^{01}	6.02×10^{01}
F7(X)	PSO-1	2.69×10^{-08}	2.55×10^{01}	1.28×10^{01}	1.20×10^{01}	9.76×10^{00}
	PSO-2	6.61×10^{-05}	2.30×10^{01}	5.23×10^{00}	1.05×10^{01}	9.26×10^{00}
	PSO-3	8.44×10^{-09}	2.56×10^{01}	1.28×10^{01}	1.19×10^{01}	1.02×10^{01}
	RPSO	1.96×10^{-01}	2.26×10^{01}	5.20×10^{00}	9.88×10^{00}	9.20×10^{00}
F8(X)	PSO-1	1.76×10^{-01}	2.47×10^{01}	2.06×10^{01}	1.43×10^{01}	9.86×10^{00}
	PSO-2	8.83×10^{-01}	2.29×10^{01}	2.02×10^{01}	1.29×10^{01}	9.29×10^{00}
	PSO-3	2.84×10^{-01}	2.42×10^{01}	2.10×10^{01}	1.36×10^{01}	9.57×10^{00}
	RPSO	2.21×10^{-01}	2.10×10^{01}	4.36×10^{00}	6.25×10^{00}	6.40×10^{00}

Table 5. Cont.

F	Index	Best	Worst	Median	Mean	Standard
F9(X)	PSO-1	2.30×10^{02}	3.76×10^{02}	2.32×10^{02}	2.36×10^{02}	2.64×10^{01}
	PSO-2	2.30×10^{02}	2.32×10^{02}	2.31×10^{02}	2.31×10^{02}	5.53×10^{-01}
	PSO-3	2.30×10^{02}	2.32×10^{02}	2.31×10^{02}	2.31×10^{02}	4.77×10^{-01}
	RPSO	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}	2.29×10^{02}	0.00×10^{00}
F10(X)	PSO-1	1.00×10^{02}	1.22×10^{02}	1.00×10^{02}	1.04×10^{02}	6.10×10^{00}
	PSO-2	1.00×10^{02}	2.20×10^{02}	1.00×10^{02}	1.49×10^{02}	5.70×10^{01}
	PSO-3	1.00×10^{02}	2.20×10^{02}	1.00×10^{02}	1.45×10^{02}	5.55×10^{01}
	RPSO	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	1.00×10^{02}	5.59×10^{-02}
F11(X)	PSO-1	0.00×10^{00}	3.00×10^{02}	0.00×10^{00}	4.01×10^{01}	8.76×10^{01}
	PSO-2	0.00×10^{00}	4.00×10^{02}	0.00×10^{00}	8.22×10^{01}	1.20×10^{02}
	PSO-3	0.00×10^{00}	4.00×10^{02}	0.00×10^{00}	4.34×10^{01}	9.08×10^{01}
	RPSO	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
F12(X)	PSO-1	1.65×10^{02}	1.67×10^{02}	1.66×10^{02}	1.66×10^{02}	6.60×10^{-01}
	PSO-2	1.64×10^{02}	1.67×10^{02}	1.65×10^{02}	1.66×10^{02}	7.50×10^{-01}
	PSO-3	1.65×10^{02}	1.68×10^{02}	1.65×10^{02}	1.66×10^{02}	8.05×10^{-01}
	RPSO	1.65×10^{02}	1.68×10^{02}	1.66×10^{02}	1.66×10^{02}	8.15×10^{-01}

As can be seen from the Table 5, except for F12, RPSO has optimal values in all other functions, especially in the functions F1–3, F5, F7–11, where the indexes of RPSO are optimal in these functions. The other variants are all able to find the theoretical optimal values in the functions F1–3, F5, and F11. This verifies that the incorporation of the various mechanisms has a certain degree of rationality. Taken together, RPSO has certain advantages in the arrangement and combination of integrated mechanisms, verifying the feasibility and effectiveness of RPSO.

5. Classification Experiments

In this section, we optimize the hyperparameters of a CNN for enhancing the diagnostic rate through RPSO and verify the feasibility of this model using two datasets.

5.1. Means

In this study, the optimization of hyperparameters was performed using RPSO. The accuracy and convergence of CNNs are heavily determined by hyperparameters [67]. Properly choosing hyperparameters is crucial, as they are determined by the particular use of the CNN. Some commonly employed hyperparameters for training CNNs include the learning rate, the number of epochs, the momentum, and the regularization factor. The learning rate governs the step size of the gradient descent algorithm, dictating how quickly or slowly the network parameters are updated. An increased learning rate facilitates faster convergence yet may result in overshooting the optimal solution. On the other hand, a decreased learning rate may take more iterations to converge, but it can yield more accurate results. The number of epochs specifies the frequency of updating the network parameters based on the training dataset. Striking a balance between not training enough epochs and overtraining, which could lead to underfitting and overfitting, is crucial. Momentum is a parameter that governs the influence of previous weight updates on the current weight update. It helps to accelerate convergence and overcome local minima. Regularization is a technique employed to prevent overfitting in the network. It accomplishes this by adding a regularization term to the loss function, penalizing complex models and promoting simpler models that can be generalized to unseen data more effectively. Optimizing hyperparameters is imperative to achieve the optimal performance of the CNN. Techniques such as RPSO can be utilized to explore the hyperparameter space and identify the most suitable values for the specific problem at hand.

5.2. Training

In this section, two datasets, CT (<https://github.com/UCSD-AI4H/COVID-CT>, accessed on 20 March 2023) and XY (<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>, accessed on 20 March 2023), were selected to validate the algorithm's ability to optimize the CNN. Overall, 70% of the data were utilized for training, while the remaining 30% were allocated for network testing. The proposed architecture aims to distinguish between normal and abnormal images. The dataset was randomly divided into training and testing sets. Through data augmentation techniques, all images were resized to $224 \times 224 \times 3$ and converted into color images. A six convolutional layer structure was employed for image classification, with Stochastic Gradient Descent (SGD) training used to adjust parameters and RPSO for hyperparameter optimization during the training process.

5.3. Testing

Firstly, the proposed network was augmented with data and trained by resizing the image to $224 \times 224 \times 3$. Then, the test image was fed into the trained CNN that had already optimized all the parameters of Convolutional Layers (CLs) and Fully Connected Layers (FCL). The CNN starts by extracting image characteristics and then categorizes them into appropriate groups through FCL and soft-max classifiers.

5.4. Experiments

To show the diagnostic performance of RPSO, the diagnostic effectiveness of RPSO was compared with that of PSO, the whale optimization algorithm (WOA) [68], the grey wolf optimizer (GWO), the hybrid differential evolution algorithm and particle swarm optimization (DEPSO) [69], and the advanced squirrel search optimization algorithm (ASSOA) [70]. Each algorithm was run independently for ten epochs; the best value, mean value, and worst value of every algorithm were recorded; and the confusion matrix of the optimal value, Accuracy, Precision, Recall, and F1 Score [71] was produced. The specific results are shown in Table 6 and Figures 9 and 10.

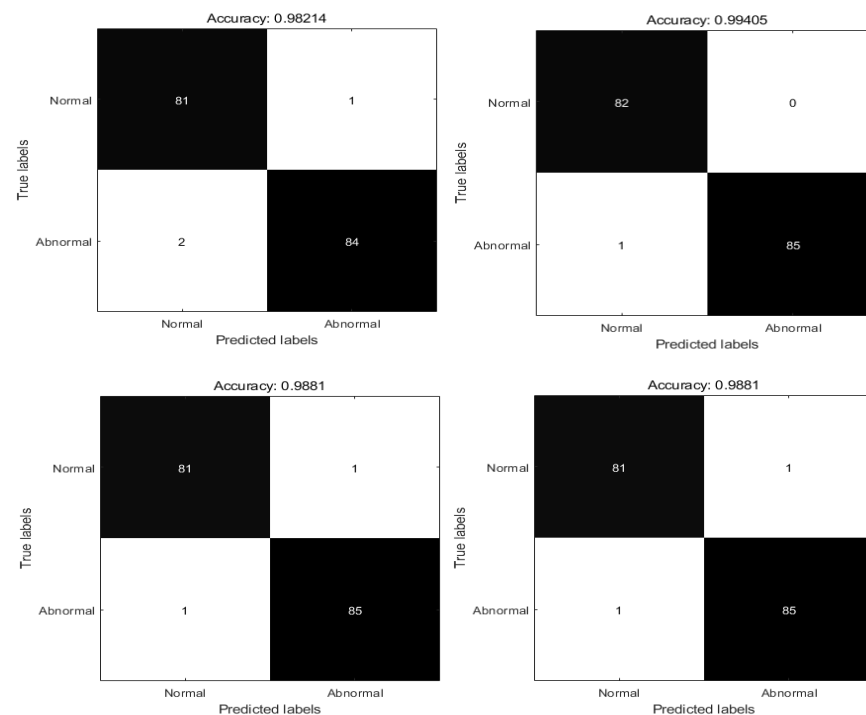


Figure 9. Cont.

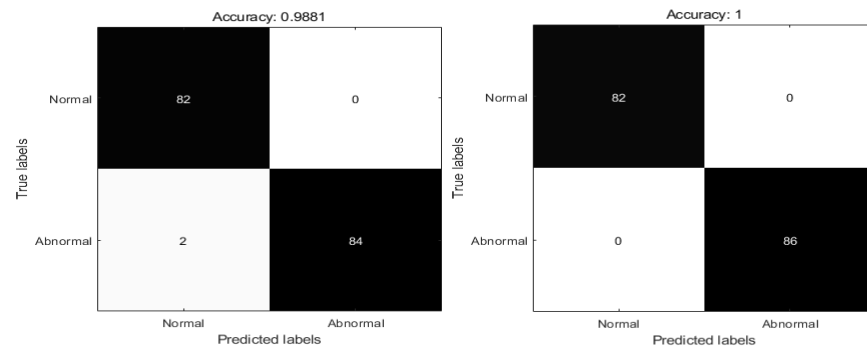


Figure 9. Confusion matrix for each algorithm with the XY dataset.

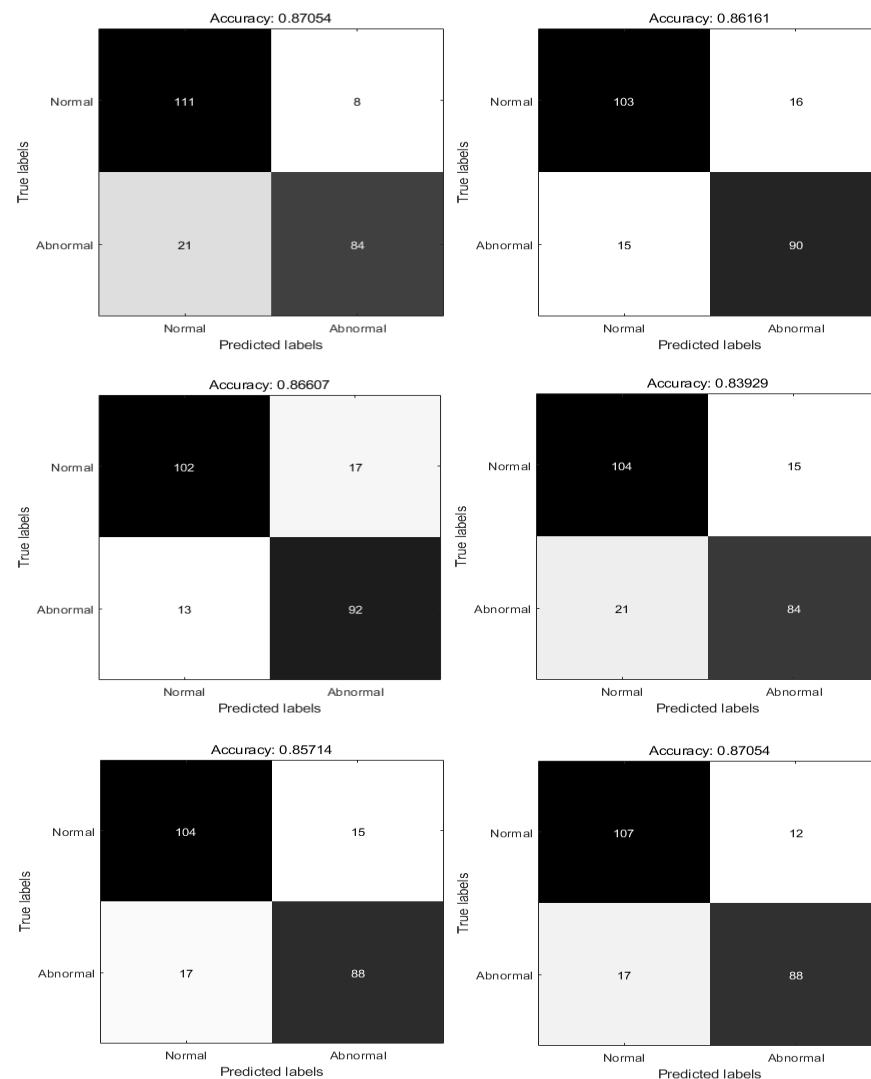


Figure 10. Confusion matrix for each algorithm with the XY dataset.

According to Table 6 and Figures 9 and 10, in the XY data set the algorithm accuracy is sorted from low to high. The results show that the accuracy rate of the PSO algorithm in the diagnosis of pneumonia is 98.21%, the number of normal correct diagnoses times is 85, and the number of correct diagnoses of COVID-19 is 84. The accuracy rate of the WOA algorithm in the diagnosis of pneumonia is 98.81%, the number of normal and correct diagnoses times is 81, and the number of times with correct diagnosis of COVID-19 is 85. The accuracy of the DEPSO algorithm in the diagnosis of pneumonia is 98.81%, the number of normal correct diagnoses is 81, and the number of correct diagnoses of COVID-19 is 85.

The accuracy of the ASSOS algorithm in the diagnosis of pneumonia is 98.81%, the number of normal correct diagnoses is 82, and the number of correct diagnoses of COVID-19 is 84. The accuracy of the GWO algorithm in the diagnosis of pneumonia is 99.40%, and the number of normal correct diagnoses is 82, while the number of times with the correct diagnosis of COVID-19 is 85. The accuracy rate of the RPSO algorithm in diagnosing pneumonia is 100%. The number of correct diagnoses for normal cases is 82, while the number of correct diagnoses for COVID-19 cases is 86.

Table 6. Classification results for each algorithm.

Method	Dataset	Accuracy	Precision	Recall	F1 Score	Best	Mean	Worst
PSO	XY	98.21%	98.82%	97.67%	98.25%	98.21%	97.86%	97.02%
	CT	87.05%	91.30%	80.00%	85.28%	87.05%	84.38%	83.04%
GWO	XY	99.40%	100.00%	98.84%	99.42%	99.40%	98.10%	96.43%
	CT	86.16%	84.91%	85.71%	85.31%	86.16%	84.91%	83.93%
WOA	XY	98.81%	98.84%	98.84%	98.84%	98.81%	97.62%	95.83%
	CT	86.61%	84.40%	87.62%	85.98%	86.61%	84.91%	83.48%
DEPSO	XY	98.81%	98.84%	98.84%	98.84%	98.81%	98.45%	97.62%
	CT	83.93%	84.85%	80.00%	82.35%	83.93%	82.59%	79.91%
ASSOA	XY	98.81%	100.00%	97.67%	98.82%	98.81%	97.86%	96.43%
	CT	85.71%	85.44%	83.81%	84.62%	85.71%	84.73%	83.93%
RPSO	XY	100%	100%	100%	100%	100%	98.69%	97.02%
	CT	87.05%	88.00%	83.81%	85.85%	87.05%	85.71%	84.82%

In the CT dataset, the algorithm accuracy is ranked from low to high. The results indicate that the ASSOA algorithm has an accuracy of 85.71% in diagnosing pneumonia, the number of normal correct diagnoses is 104, and the number of correct diagnoses of COVID-19 cases is 88. The DEPSO algorithm has an accuracy rate of 83.92% in diagnosing pneumonia. The number of correct diagnoses for normal cases is 104, and the number of correct diagnoses for COVID-19 cases is 84. The GWO algorithm has an accuracy of 86.16% in diagnosing pneumonia. The number of correct diagnoses for normal cases is 103, and the number of correct diagnoses for COVID-19 cases is 90. The WOA algorithm has an accuracy of 86.61% in diagnosing pneumonia. The number of correct diagnoses for normal cases is 102, and the number of correct diagnoses for COVID-19 cases is 92. The accuracy of the PSO algorithm in diagnosing pneumonia is 87.05%. The number of correct diagnoses for normal cases is 111, and for COVID-19 cases it is 84. The accuracy rate of the RPSO algorithm in diagnosing pneumonia is also 87.05%. The number of correct diagnoses for normal cases is 107, and for COVID-19 cases it is 88.

It can be seen that, compared with other algorithms, the RPSO algorithm has the highest accuracy in both data sets, surpassing all of the compared algorithms. Although the accuracy of the PSO algorithm in the CT data set is equal to that of RPSO, with both ranking first, it has the lowest accuracy in the XY data set and is highly unstable. It is concluded that RPSO has obvious superiority and good robustness.

6. Conclusions

To further enhance the global optimization ability of the particle swarm algorithm and its applicability in threshold segmentation problems, this paper proposes an RPSO based on cosine similarity. This algorithm introduces the RCI mechanism to enhance the mechanism and the global search capability of the algorithm. Utilizing cosine similarity for RCI and QI selection, the population information is effectively and dynamically updated, thereby improving the global and local optimization capabilities of the algorithm. RPSO is evaluated based on the CEC 2022 test set and compared with other PSO variants and top algorithms in the CEC community. In comparison with other variants of PSO, RPSO is ranked top at 7.9167 and 2.5833 in 10 and 20 dimensions, respectively. When compared with the top algorithms, RPSO achieves optimal values in F(1), F(5), and F(11), performing

on par with the leading algorithms. The results confirm that RPSO exhibits strong competitive and global optimization capabilities. In the optimization of CNN classification experiments, RPSO demonstrates the highest classification accuracy of 100% on the XY dataset, with an average accuracy exceeding 98%. On the CT dataset, the highest accuracy reached is over 87%, with an average accuracy of above 85%. RPSO demonstrates excellent diagnostic performance based on two disease datasets compared to other basic algorithms and algorithm variants, highlighting its practical value.

Despite the promising optimization results achieved by RPSO based on the CEC 2022 test set, some limitations still exist. For instance, in functions F(3) and F(4), RPSO exhibits some shortcomings in search accuracy, indicating the need to improve its local search capability while maintaining its global search ability, as RPSO is not capable of making intelligent choices based on the optimization-seeking environment. Additionally, optimizing the hyperparameters of the CNN to significantly enhance the classification accuracy is a challenging task, as optimization is a major problem that takes an inordinate amount of time to ensure. Therefore, the next step of this research focuses on three main areas:

- Designing smarter judgment mechanisms for rational global and local searching.
- Incorporating efficient parameter estimation schemes to reduce the time consumption of the optimization process.
- Optimizing the architecture of the neural network to enable automatic architectural tuning for different datasets.
- Applying the algorithm to practical scenarios in high demand today, such as 6G base station deployment, optimization of new energy systems, and structural design optimization.

Author Contributions: Conceptualization, Y.Z. (Yuan Zeng). and R.L.; methodology, G.H.; software, X.Z.; validation Y.Z. (Yuemai Zhang). and T.L.; formal analysis, W.L.; data curation, D.Z.; writing—original draft preparation, Y.L.; writing—review and editing, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Research Project of Jiangxi Provincial Education Department (Grant No. GJJ2203314).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Eltamaly, A.M. A novel musical chairs algorithm applied for MPPT of PV systems. *Renew. Sustain. Energy Rev.* **2021**, *146*, 111135. [\[CrossRef\]](#)
2. 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. [\[CrossRef\]](#)
3. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; IEEE: New York, NY, USA, 1995; pp. 1942–1948.
4. Eltamaly, A.M. A novel particle swarm optimization optimal control parameter determination strategy for maximum power point trackers of partially shaded photovoltaic systems. *Eng. Optim.* **2022**, *54*, 634–650. [\[CrossRef\]](#)
5. Zhu, D.; Wang, S.; Shen, J.; Zhou, C.; Li, T.; Yan, S. A multi-strategy particle swarm algorithm with exponential noise and fitness-distance balance method for low-altitude penetration in secure space. *J. Comput. Sci.* **2023**, *74*, 102149. [\[CrossRef\]](#)
6. Kocak, O.; Erkan, U.; Toktas, A.; Gao, S. PSO-based image encryption scheme using modular integrated logistic exponential map. *Expert Syst. Appl.* **2024**, *237*, 121452. [\[CrossRef\]](#)
7. Xie, L.; Wang, S.; Zhu, D.; Hu, G.; Zhou, C. DNA Sequence Optimization Design of Arithmetic Optimization Algorithm Based on Billiard Hitting Strategy. *Interdisciplinary Sciences. Comput. Life Sci.* **2023**, *15*, 231–248.
8. Guedria, N.B. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **2016**, *40*, 455–467. [\[CrossRef\]](#)
9. Zhu, Y.; Li, G.; Wang, R.; Tang, S.; Su, H.; Cao, K. Intelligent fault diagnosis of hydraulic piston pump combining improved LeNet-5 and PSO hyperparameter optimization. *Appl. Acoust.* **2021**, *183*, 108336. [\[CrossRef\]](#)

10. Deng, W.; Yao, R.; Zhao, H.; Yang, X.; Li, G. A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. *Soft Comput.* **2019**, *23*, 2445–2462. [\[CrossRef\]](#)
11. Yadav, R.K. PSO-GA based hybrid with Adam Optimization for ANN training with application in Medical Diagnosis. *Cogn. Syst. Res.* **2020**, *64*, 191–199. [\[CrossRef\]](#)
12. Zhang, K.; Su, J.; Sun, S.; Liu, Z.; Wang, J.; Du, M.; Zhang, Q. Compressor fault diagnosis system based on PCA-PSO-LSSVM algorithm. *Sci. Prog.* **2021**, *104*, 368504211026110. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Raja, J.B.; Pandian, S.C. PSO-FCM based data mining model to predict diabetic disease. *Comput. Methods Programs Biomed.* **2020**, *196*, 105659. [\[CrossRef\]](#)
14. Al-Andoli, M.N.; Tan, S.C.; Sim, K.S.; Lim, C.P.; Goh, P.Y. Parallel Deep Learning with a hybrid BP-PSO framework for feature extraction and malware classification. *Appl. Soft Comput.* **2022**, *131*, 109756. [\[CrossRef\]](#)
15. Huang, C.; Zhang, Z.; Wang, L. PSOPruner: PSO-based deep convolutional neural network pruning method for PV module defects classification. *IEEE J. Photovolt.* **2022**, *12*, 1550–1558. [\[CrossRef\]](#)
16. George, S.T.; Subathra, M.S.P.; Sairamya, N.J.; Susmitha, L.; Premkumar, M.J. Classification of epileptic EEG signals using PSO based artificial neural network and tunable-Q wavelet transform. *Biocybern. Biomed. Eng.* **2020**, *40*, 709–728. [\[CrossRef\]](#)
17. Issa, M.; Helmi, A.M.; Elsheikh, A.H.; Abd Elaziz, M. A biological sub-sequences detection using integrated BA-PSO based on infection propagation mechanism: Case study COVID-19. *Expert Syst. Appl.* **2022**, *189*, 116063. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Dixit, A.; Mani, A.; Bansal, R. CoV2-Detect-Net: Design of COVID-19 prediction model based on hybrid DE-PSO with SVM using chest X-ray images. *Inf. Sci.* **2021**, *571*, 676–692. [\[CrossRef\]](#)
19. Sharif, M.; Amin, J.; Raza, M.; Yasmin, M.; Satapathy, S.C. An integrated design of particle swarm optimization (PSO) with fusion of features for detection of brain tumor. *Pattern Recognit. Lett.* **2020**, *129*, 150–157. [\[CrossRef\]](#)
20. Verma, H.; Verma, D.; Tiwari, P.K. A population based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain image. *Expert Syst. Appl.* **2021**, *167*, 114121. [\[CrossRef\]](#)
21. Muduli, D.; Dash, R.; Majhi, B. Fast discrete curvelet transform and modified PSO based improved evolutionary extreme learning machine for breast cancer detection. *Biomed. Signal Process. Control* **2021**, *70*, 102919. [\[CrossRef\]](#)
22. Luo, X.; Chen, J.; Yuan, Y.; Wang, Z. Pseudo Gradient-Adjusted Particle Swarm Optimization for Accurate Adaptive Latent Factor Analysis. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 2213–2226. [\[CrossRef\]](#)
23. Gupta, A.; Nahar, P. Sandpiper optimization algorithm with cosine similarity based cross-layer routing protocol for smart agriculture in wireless sensor network assisted internet of things systems. *Int. J. Commun. Syst.* **2023**, *36*, e5514. [\[CrossRef\]](#)
24. Kwak, B.I.; Han, M.L.; Kim, H.K. Cosine similarity based anomaly detection methodology for the CAN bus. *Expert Syst. Appl.* **2021**, *166*, 114066. [\[CrossRef\]](#)
25. Zeng, N.; Wang, Z.; Liu, W.; Zhang, H.; Hone, K.; Liu, X. A dynamic neighborhood-based switching particle swarm optimization algorithm. *IEEE Trans. Cybern.* **2020**, *52*, 9290–9301. [\[CrossRef\]](#)
26. Zaman, H.R.R.; Gharehchopogh, F.S. An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Eng. Comput.* **2022**, *38* (Suppl. S4), 2797–2831. [\[CrossRef\]](#)
27. Xu, H.Q.; Gu, S.; Fan, Y.C.; Li, X.S.; Zhao, Y.F.; Zhao, J.; Wang, J.J. A strategy learning framework for particle swarm optimization algorithm. *Inf. Sci.* **2023**, *619*, 126–152. [\[CrossRef\]](#)
28. Chen, X.; Tianfield, H.; Du, W. Bee-foraging learning particle swarm optimization. *Appl. Soft Comput.* **2021**, *102*, 107134. [\[CrossRef\]](#)
29. Khan, T.A.; Ling, S.H. A novel hybrid gravitational search particle swarm optimization algorithm. *Eng. Appl. Artif. Intell.* **2021**, *102*, 104263. [\[CrossRef\]](#)
30. Molaei, S.; Moazen, H.; Najjar-Ghabel, S.; Farzinvash, L. Particle swarm optimization with an enhanced learning strategy and crossover operator. *Knowl.-Based Syst.* **2021**, *215*, 106768. [\[CrossRef\]](#)
31. Ali, A.F.; Tawhid, M.A. A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Eng. J.* **2017**, *8*, 191–206. [\[CrossRef\]](#)
32. Xia, X.; Gui, L.; Yu, F.; Wu, H.; Wei, B.; Zhang, Y.L.; Zhan, Z.H. Triple archives particle swarm optimization. *IEEE Trans. Cybern.* **2019**, *50*, 4862–4875. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Zhang, Y.; Liu, X.; Bao, F.; Chi, J.; Zhang, C.; Liu, P. Particle swarm optimization with adaptive learning strategy. *Knowl.-Based Syst.* **2020**, *196*, 105789. [\[CrossRef\]](#)
34. Li, H.; Li, J.; Wu, P.; You, Y.; Zeng, N. A ranking-system-based switching particle swarm optimizer with dynamic learning strategies. *Neurocomputing* **2022**, *494*, 356–367. [\[CrossRef\]](#)
35. Sheng, M.; Wang, Z.; Liu, W.; Wang, X.; Chen, S.; Liu, X. A particle swarm optimizer with multi-level population sampling and dynamic p-learning mechanisms for large-scale optimization. *Knowl.-Based Syst.* **2022**, *242*, 108382. [\[CrossRef\]](#)
36. Lu, J.; Zhang, J.; Sheng, J. Enhanced multi-swarm cooperative particle swarm optimizer. *Swarm Evol. Comput.* **2022**, *69*, 100989. [\[CrossRef\]](#)
37. Liu, W.; Wang, Z.; Yuan, Y.; Zeng, N.; Hone, K.; Liu, X. A novel sigmoid-function-based adaptive weighted particle swarm optimizer. *IEEE Trans. Cybern.* **2019**, *51*, 1085–1093. [\[CrossRef\]](#)
38. Liu, H.; Zhang, X.W.; Tu, L.P. A modified particle swarm optimization using adaptive strategy. *Expert Syst. Appl.* **2020**, *152*, 113353. [\[CrossRef\]](#)
39. Li, Y.; Yu, K.; Liang, J.; Yue, C.; Qiao, K. A landscape-aware particle swarm optimization for parameter identification of photovoltaic models. *Appl. Soft Comput.* **2022**, *131*, 109793. [\[CrossRef\]](#)

40. Zhang, Y.; Kong, X. A particle swarm optimization algorithm with empirical balance strategy. *Chaos Solitons Fractals X* **2023**, *10*, 100089. [\[CrossRef\]](#)
41. Wang, R.; Hao, K.; Chen, L.; Wang, T.; Jiang, C. A novel hybrid particle swarm optimization using adaptive strategy. *Inf. Sci.* **2021**, *579*, 231–250. [\[CrossRef\]](#)
42. Li, X.; Mao, K.; Lin, F.; Zhang, X. Particle swarm optimization with state-based adaptive velocity limit strategy. *Neurocomputing* **2021**, *447*, 64–79. [\[CrossRef\]](#)
43. Wang, F.; Wang, X.; Sun, S. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. *Inf. Sci.* **2022**, *602*, 298–312. [\[CrossRef\]](#)
44. Tian, J.; Hou, M.; Bian, H.; Li, J. Variable surrogate model-based particle swarm optimization for high-dimensional expensive problems. *Complex Intell. Syst.* **2023**, *9*, 3887–3935. [\[CrossRef\]](#)
45. Liu, Z.; Nishi, T. Strategy dynamics particle swarm optimizer. *Inf. Sci.* **2022**, *582*, 665–703. [\[CrossRef\]](#)
46. Liu, W.; Wang, Z.; Zeng, N.; Yuan, Y.; Alsaadi, F.E.; Liu, X. A novel randomised particle swarm optimizer. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 529–540. [\[CrossRef\]](#)
47. Machado, J.T.; Pahnkekolaei, S.M.A.; Alfi, A. Complex-order particle swarm optimization. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *92*, 105448. [\[CrossRef\]](#)
48. Yang, Q.; Song, G.W.; Chen, W.N.; Jia, Y.H.; Gao, X.D.; Lu, Z.Y.; Zhang, J. Random Contrastive Interaction for Particle Swarm Optimization in High-Dimensional Environment. *IEEE Trans. Evol. Comput.* **2023**; *in press*.
49. Ye, J.; Xie, L.; Wang, H. A water cycle algorithm based on quadratic interpolation for high-dimensional global optimization problems. *Appl. Intell.* **2023**, *53*, 2825–2849. [\[CrossRef\]](#)
50. Qaraad, M.; Amjad, S.; Hussein, N.K.; Farag, M.A.; Mirjalili, S.; Elhosseini, M.A. Quadratic interpolation and a new local search approach to improve particle swarm optimization: Solar photovoltaic parameter estimation. *Expert Syst. Appl.* **2024**, *236*, 121417. [\[CrossRef\]](#)
51. Ratnaweera, A.; Halgamuge, S.K.; Watson, H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [\[CrossRef\]](#)
52. Zhang, Q.; Zou, D.; Duan, N. An improved differential evolution using self-adaptable cosine similarity for economic emission dispatch. *Energy* **2023**, *283*, 128421. [\[CrossRef\]](#)
53. Alazab, M.; Khurma, R.A.; Awajan, A.; Camacho, D. A new intrusion detection system based on Moth–Flame Optimizer algorithm. *Expert Syst. Appl.* **2022**, *210*, 118439. [\[CrossRef\]](#)
54. Li, W.; Liang, P.; Sun, B.; Sun, Y.; Huang, Y. Reinforcement learning-based particle swarm optimization with neighborhood differential mutation strategy. *Swarm Evol. Comput.* **2023**, *78*, 101274. [\[CrossRef\]](#)
55. Yang, J.; Yu, J.; Huang, C. Adaptive multistrategy ensemble particle swarm optimization with Signal-to-Noise ratio distance metric. *Inf. Sci.* **2022**, *612*, 1066–1094. [\[CrossRef\]](#)
56. Zhang, Y. Elite archives-driven particle swarm optimization for large scale numerical optimization and its engineering applications. *Swarm Evol. Comput.* **2023**, *76*, 101212. [\[CrossRef\]](#)
57. Duman, S.; Kahraman, H.T.; Korkmaz, B.; Bakir, H.; Guvenc, U.; Yilmaz, C. Improved Phasor Particle Swarm Optimization with Fitness Distance Balance for Optimal Power Flow Problem of Hybrid AC/DC Power Grids. In Proceedings of the International Conference on Artificial Intelligence and Applied Mathematics in Engineering, Belek, Turkey, 1–3 October 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 307–336.
58. Shami, T.M.; Mirjalili, S.; Al-Eryani, Y.; Daoudi, K.; Izadi, S.; Abualigah, L. Velocity pausing particle swarm optimization: A novel variant for global optimization. *Neural Comput. Appl.* **2023**, *35*, 9193–9223. [\[CrossRef\]](#)
59. Meng, Z.; Zhong, Y.; Mao, G.; Liang, Y. PSO-sono: A novel PSO variant for single-objective numerical optimization. *Inf. Sci.* **2022**, *586*, 176–191. [\[CrossRef\]](#)
60. Li, T.; Shi, J.; Deng, W.; Hu, Z. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* **2022**, *121*, 108731. [\[CrossRef\]](#)
61. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
62. Mohamed, A.W.; Hadi, A.A.; Agrawal, P.; Sallam, K.M.; Mohamed, A.K. Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with IMODE algorithm for solving CEC 2021 benchmark problems. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: New York, NY, USA, 2021; pp. 841–848.
63. Bujok, P.; Kolenovsky, P. Eigen crossover in cooperative model of evolutionary algorithms applied to CEC 2022 single objective numerical optimization. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2022; IEEE: New York, NY, USA, 2022; pp. 1–8.
64. Sallam, K.M.; Elsayed, S.M.; Chakraborty, R.K.; Ryan, M.J. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2020; IEEE: New York, NY, USA, 2021; pp. 1–8.
65. dos Santos Coelho, L.; Ayala, H.V.; Freire, R.Z. Population’s variance-based adaptive differential evolution for real parameter optimization. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; IEEE: New York, NY, USA, 2013; pp. 1672–1677.

66. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K.; Awad, N.H. Evaluating the performance of adaptive gaining sharing knowledge based algorithm on CEC 2020 benchmark problems. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; IEEE: New York, NY, USA, 2020; pp. 1–8.
67. Da Silva, G.L.F.; Valente, T.L.A.; Silva, A.C.; De Paiva, A.C.; Gattass, M. Convolutional neural network-based PSO for lung nodule false positive reduction on CT images. *Comput. Methods Programs Biomed.* **2018**, *162*, 109–118. [[CrossRef](#)]
68. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
69. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
70. El-Kenawy, E.S.M.; Mirjalili, S.; Ibrahim, A.; Alrahmawy, M.; El-Said, M.; Zaki, R.M.; Eid, M.M. Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient COVID-19 X-ray chest image classification. *IEEE Access* **2021**, *9*, 36019–36037. [[CrossRef](#)] [[PubMed](#)]
71. Goel, T.; Murugan, R.; Mirjalili, S.; Chakrabartty, D.K. OptCoNet: An optimized convolutional neural network for an automatic diagnosis of COVID-19. *Appl. Intell.* **2021**, *51*, 1351–1366. [[CrossRef](#)] [[PubMed](#)]

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.