



ARTICLE

Enhanced Particle Swarm Optimization Algorithm Based on SVM Classifier for Feature Selection

Xing Wang^{1,*}, Huazhen Liu¹, Abdelazim G. Hussien², Gang Hu¹ and Li Zhang³

¹School of Science, Xi'an University of Technology, Xi'an, 710054, China

²Department of Computer and Information Science, Linköping University, Linköping, 58183, Sweden

³School of Science, Chang'an University, Xi'an, 710064, China

*Corresponding Author: Xing Wang, Email: wangxing@xaut.edu.cn

Received: 13 September 2024; Accepted: 06 January 2025; Published: 03 March 2025

ABSTRACT: Feature selection (FS) is essential in machine learning (ML) and data mapping by its ability to preprocess high-dimensional data. By selecting a subset of relevant features, feature selection cuts down on the dimension of the data. It excludes irrelevant or surplus features, thus boosting the performance and efficiency of the model. Particle Swarm Optimization (PSO) boasts a streamlined algorithmic framework and exhibits rapid convergence traits. Compared with other algorithms, it incurs reduced computational expenses when tackling high-dimensional datasets. However, PSO faces challenges like inadequate convergence precision. Therefore, regarding FS problems, this paper presents a binary version enhanced PSO based on the Support Vector Machines (SVM) classifier. First, the Sand Cat Swarm Optimization (SCSO) is added to enhance the global search capability of PSO and improve the accuracy of the solution. Secondly, the Latin hypercube sampling strategy initializes populations more uniformly and helps to increase population diversity. The last is the roundup search strategy introducing the grey wolf hierarchy idea to help improve convergence speed. To verify the capability of Self-adaptive Cooperative Particle Swarm Optimization (SCPSO), the CEC2020 test suite and CEC2022 test suite are selected for experiments and applied to three engineering problems. Compared with the standard PSO algorithm, SCPSO converges faster, and the convergence accuracy is significantly improved. Moreover, SCPSO's comprehensive performance far exceeds that of other algorithms. Six datasets from the University of California, Irvine (UCI) database were selected to evaluate SCPSO's effectiveness in solving feature selection problems. The results indicate that SCPSO has significant potential for addressing these problems.

KEYWORDS: Feature selection; SVM; particle swarm optimization; sand cat swarm optimization; engineering problems

1 Introduction

With the rapid progress of society and the economy, the era of big data is advancing steadily. Data collection and storage technology are widely utilized, and a massive amount of data has been accumulated. Against this background, extracting helpful information has proven challenging due to the sheer volume of data, its varying quality, and the diversity of data sources. Traditional data analysis methods, like hypothesis testing and regression analysis, can no longer handle large data sets. Even if some datasets are not significant in quantity, they cannot be analyzed using these methods due to the unique properties of the data. Therefore, to overcome these challenges, new techniques must be developed. Data mapping integrates classic data analysis tools with sophisticated algorithms to effectively process vast volumes of data and discover valuable knowledge within them. By applying data mapping techniques, hidden and helpful information and patterns



can be extracted from massive databases. Thus, critical conclusions and trends can be drawn, helping people understand the data and make meaningful analyses and applications.

Feature selection (FS) is an instrumental part of Machine Learning (ML) and data mapping [1]. It can extract a valid set of features from a substantial quantity of redundant, noisy, and irrelevant information to raise the performance of learning algorithms, reduce computational expenses, improve the interpretability of models, and reduce the risk of over-fitting [2]. FS attempts to optimize the system for a particular metric by selecting N features from the available M features, where $N < M$.

Generally, FS methods have three categories [3]: filtering approach, embedding approach, and wrapping approach. First, independent of the specific learning algorithm, the filtering approach selects the most significant subset of features by assessing and ranking the features. The more common filtering approaches [4] are variance selection, mutual information method, chi-square test, etc. They all choose the top-ranked features by some law, and these guidelines can be based on statistical indicators, information theory, correlation coefficients, and so on. Filtering approaches are computationally simple and efficient, but they have several limitations. Due to their independence, they may not be able to capture the complex interactions between features; furthermore, focusing only on the attributes of the features themselves and the relationship with the target variables without considering the correlation between the features makes it impossible to discover the optimal subset of features for a given problem and data set. In practical applications, combining other FS approaches to obtain better results is frequently essential. Second, embedding approaches [5] perform feature selection simultaneously with model training and decide whether to select a feature or not by evaluating the importance or weight of the feature during the model training process. Embedding approaches have become familiar and effective in FS due to their advantages of automation, making full use of data, avoiding over-fitting, learning task adaptation, and reducing dimension catastrophe.

Finally, the core idea of the wrapping approach [6] is to convert the FS task into a subset search problem and to assess the effectiveness of various feature subsets using an evaluation metric. These evaluation criteria can be specific performance metrics of the learning algorithm or classifier, such as accuracy, mean square error, etc. Moreover, this approach can be combined with diverse classifiers, including the K-Nearest Neighbor method (KNN) [7], Decision Tree (DT) [8], Support Vector Machines (SVM) [9], and other similar techniques. Although these methods are computationally more expensive, they provide a more accurate measure of the impact of features on model performance and have more vital generalization ability. The wrapping approach generally utilizes optimization techniques to deal with the FS problem. The FS may be viewed as an optimization problem solved using existing optimization algorithms and techniques [10] and finding the near-optimal subset of features.

Meta-heuristic algorithms are extensively applied to solve FS problems by their extreme optimization search capability. In general, they randomly generate a group of initial solutions and then evaluate this group of solutions using an evaluation function to measure their performance [11] in the FS problem. Next, the algorithm undergoes a series of search strategies to refine the current solution through iteration. During each iteration, they adjust the state of the solution according to some rules and ways that approximate the optimal solution until the termination condition is reached or a predetermined result is satisfied. However, although each evaluated algorithm can perform well when considering some specific optimization problems, the No Free Lunch theorem [12] shows that no super-optimization technique can tackle all optimization problems. Hence, refining existing algorithms or inventing more capable ones is essential.

Particle Swarm Optimization (PSO) is a conventional meta-heuristic algorithm that takes inspiration from the foraging and social behavior of bird flocks [13]. Due to its relatively simple core thinking, PSO finds broad application in optimization problems. However, several challenges and limitations exist, such as being vulnerable to local optimization. Therefore, an enhanced particle swarm optimization is proposed in

this paper. Meanwhile, the proposed algorithm is tested on two challenging test sets and applied to three engineering applications and six feature selection problems. The primary achievements of this paper are outlined as follows:

(1) Integrate the benefits of the Sand Cat Swarm Optimization (SCSO) into PSO and integrate two enhanced methodologies.

- The incorporation of the SCSO tackles the problem of restricted search breadth present in the original method, thereby greatly enhancing the algorithm's ability to explore and reducing the likelihood of becoming ensnared in local optima.
- In the algorithm's initial phase, the initial population's random generation is discarded in favor of a more high-level Latin hypercube sampling strategy that permits particles to explore a broader range of the entire search space.
- Roundup search strategy is introduced so particles can more quickly determine the target's location during the search process, significantly reducing the computational cost and increasing the optimization efficiency.

(2) Compare the suggested algorithm with other enhanced PSO and various algorithms on the CEC2020 and CEC2022 test suites.

(3) Self-adaptive Cooperative Particle Swarm Optimization (SCPSO) is tested against other excellent algorithms on three engineering application problems with satisfactory results.

(4) SCPSO has demonstrated its effectiveness in addressing six feature selection problems, producing subsets that exhibit high accuracy while containing only a limited number of features.

The following chapters are outlined: [Section 2](#) introduces the improvement and optimization of PSO in recent years and the researchers' settlement and study of the feature selection problem. [Section 3](#) briefly describes the basic framework and optimization process of PSO and SCSO. In [Section 4](#), the contents of the two enhanced strategies and the process of SCPSO are described in detail. [Section 5](#) concentrates on verifying the performance of SCPSO on two test sets. Three real-world problems are chosen in [Section 6](#) to test SCPSO's ability to address the optimization problem further. The study of SCPSO to solve six feature selection problems is presented in [Section 7](#). [Section 8](#) offers a recapitulation and prospects.

2 Relevant Research

2.1 Research on PSO

Recently, continuous work has been done to improve PSO. In 2015, an SL-PSO proposed by Cheng et al. [14] incorporated a social learning mechanism into PSO and performed well on 40 low-dimensional issues and seven high-dimensional functions. HFPSO, a hybrid algorithm combining Firefly and Particle Swarm Optimization, proposed by Aydilek [15] in 2018, was tested on the CEC2015 and CEC2017 test suites in different dimensions as well as several engineering and mechanical design benchmark issues. The results showed that HFPSO provided fast and reliable optimized solutions with single peaks for computationally expensive numerical functions, simple multi-peak, mixed, and combined classes outperforming other algorithms. Moreover, researchers have also employed PSO in many real-world applications. The new hybrid genetic PSO proposed by Ha et al. [16] in 2020 obtains the advantages of both algorithms and achieves satisfactory results in allocating distributed generation in distribution grids. To enhance the operational efficacy of micro gas turbines under a wide range of operating conditions, Yang et al. [17], in 2021, developed an algorithm named HIPSO_C.S. by combining PSO with the Cuckoo Search Algorithm.

2.2 Research on Feature Selection

As the optimization power of the meta-heuristic algorithms increases, so does the number of examples of their adoption for the FS algorithm. In the following, we will explore some of the studies in this domain. Zarita et al. [18] in 2016 proposed a wrapper's Harmony Search Algorithm that successfully addressed the detection and prediction of epileptic seizures and enabled HS to discover a more reasonable solution within a limited time frame by varying the initialization process of HS and the improvisation of the solution. Desbordes et al. [19] applied a Genetic Algorithm to tackle the issue of the survival rate of patients after esophageal cancer treatment. The ultimate forecast outcomes were obtained as a subset of nine features, and the comparative findings indicated that the utilization of GA outperformed alternative approaches. Tu et al. [20] added the enhanced global optimal guidance strategy, adaptive cooperation strategy, and dispersed foraging strategy to Gray Wolf Optimization Algorithm (GWO) to obtain the multi-strategy ensemble GWO (MEGWO), which not only increases the local and global search ability of GWO but also enriches the species diversity. They also applied the MEGWO to 12 FS issues, which proves its reliability and utility in solving real-world issues. Meta-heuristic algorithms hold substantial significance in cancer prediction, and the enhanced Cuckoo search algorithm proposed by Malek et al. [21] can identify a reduced set of features while attaining superior classification accuracy. Besides, the binary teaching-based optimization algorithm [22], binary dandelion algorithm [23], and binary capuchin monkey search algorithm [24] came into being. It was adopted to solve the FS problem better.

3 PSO Overview

Particle Swarm Optimization [25], recognized as a leading meta-heuristic algorithm, was initially introduced by Eberhart and Kennedy in 1995. PSO utilizes the biological flock model proposed by biologist Heppner to simulate birds' feeding behavior. Assuming that there is one and only one piece of food in a defined area, a community of birds randomly seeks the food without knowing the exact location of the food, and the simplest and easiest to implement approach is to explore the vicinity of the bird nearest to the food. Consider each bird as a weightless and volume-less point mass in space, and each point mass is assigned a fitness value based on a particular function. Besides, velocity governs both the direction and distance of flight.

First, a uniform probability distribution function is deployed to randomly distribute each particle $Pa_i = \{pa_{i1}, pa_{i2}, \dots, pa_{iDim}\}$, ($1 \leq i \leq Nu$) throughout the exploration space, and all the particles in the exploration space form a particle population $POP = \{Pa_1, Pa_2, \dots, Pa_{Nu}\}$ composed of Nu particles, with each particle represented as a Dim dimension vector; furthermore, the velocity $Ve_i = \{ve_{i1}, ve_{i2}, \dots, ve_{iDim}\}$ of each particle is randomly initialized with the specific formula shown below:

$$Pa_i = Lb + (Ub - Lb) \cdot rand(0, 1) \quad (1)$$

$$Ve_i = Ve_{min} + 2 \cdot Ve_{max} \cdot rand(0, 1) \quad (2)$$

where Ub and Lb are the upper and lower bounds of all the population variables, Ve_{max} , Ve_{min} are the maximum and minimum values of the velocity of each particle. The fitness of each individual in the population is calculated using the function Fit , and the historical optimal position $Pabest_i$ of particle Pa_i , which can also be called the local optimal position, is given by the following formula:

$$Pabest_i(t+1) = \begin{cases} Pabest_i(t) & F(Pa_i(t+1)) \geq F(Pabest_i(t+1)) \\ Pa_i(t+1) & F(Pa_i(t+1)) < F(Pabest_i(t+1)) \end{cases} \quad (3)$$

where t is the current iteration number. Finally, the location of the global best individual $GPabest$ is determined by evaluating the fitness values of all particles in the population, i.e.,

$$F(GPabest) = \min \{F(Pabest_1(t)), F(Pabest_2(t)), \dots, F(Pabest_{Nu}(t))\}$$

$$GPabest \in \{Pabest_1(t), Pabest_2(t), \dots, Pabest_{Nu}(t)\} \quad (4)$$

In the optimization process, the combination of two optimal values is applied to optimize the flight speed of each particle, which is then employed to adjust the velocity of each particle. The specific formula is presented below:

$$Ve_i(t+1) = \omega \cdot Ve_i(t) + a_1 \cdot r_1 \cdot (Pabest_i(t) - Pa_i(t)) + a_2 \cdot r_2 \cdot (GPabest_i(t) - Pa_i(t)) \quad (5)$$

$$Pa_i(t+1) = Pa_i(t) + Ve_i(t) \quad (6)$$

where $a_1 = a_2 = 0.5$ are self-learning factor and group learning factor correspondingly, and r_1, r_2 are arbitrary numbers obeying a uniform distribution.

4 SCPSO Overview

The PSO algorithm's principle is relatively simple, its structure is concise, and it can achieve fast convergence. This can significantly save computational costs for feature selection problems requiring high-dimensional data processing. However, the fast convergence characteristic also makes the algorithm prone to getting stuck in local optima, resulting in poor algorithm accuracy. Therefore, it is necessary to adopt enhancement strategies to improve the algorithm's performance.

4.1 Latin Hypercube Sample Strategy

Most meta-heuristic algorithms adopt the technique of random initialization to obtain an initial population. However, this method is relatively low-level and highly prone to producing an unevenly distributed population, affecting the algorithm's convergence rate and precision. The Latin hypercube sample technique is a statistical method of approximate random sampling from a multivariate distribution that produces as many uniformly distributed sample points as possible. He et al. added it to the Firework Algorithm to better solve the mixed polarity Reed-Muller (MPRM) logic loop area optimization problem [26]. The enhanced Atom Search Algorithm using a Latin hypercube sample is more stable and reliable in optimizing the extreme learning machine model [27]. So SCPSO uses the Latin hypercube sample strategy to make individuals among the starting population span across the whole search space to the greatest extent possible so that the distribution of individuals can be more rationalized, and the diversity of the population can be improved. Then, the specific processes for initializing SCPSO are outlined as follows:

- (1) Determine the population size Nu and the dimension Dim .
- (2) Divide the variable interval into Nu equally spaced non-overlapping subintervals.
- (3) Randomly select one point in each dimension for each subinterval separately.
- (4) Form an initial population from the randomly selected points.

Fig. 1 illustrates the comparison of the effects of the two initialization methods. From the figure, it is evident that the population obtained by the initialization of the Latin hypercube sample strategy has a more robust distribution, which makes it more beneficial to choose the sample points uniformly in all dimensions, making the dispersion of the initial population throughout the space more balanced and comprehensive. That initialized populations more uniformly and helps to increase population diversity. In contrast, random initialization possesses a distinctly short-periodic state with a less even distribution, which means that

populations may cluster along specific directions in the search space while ignoring other possible search directions. This situation may limit the global search capability of the algorithm and lead to failure to achieve better optimization results.

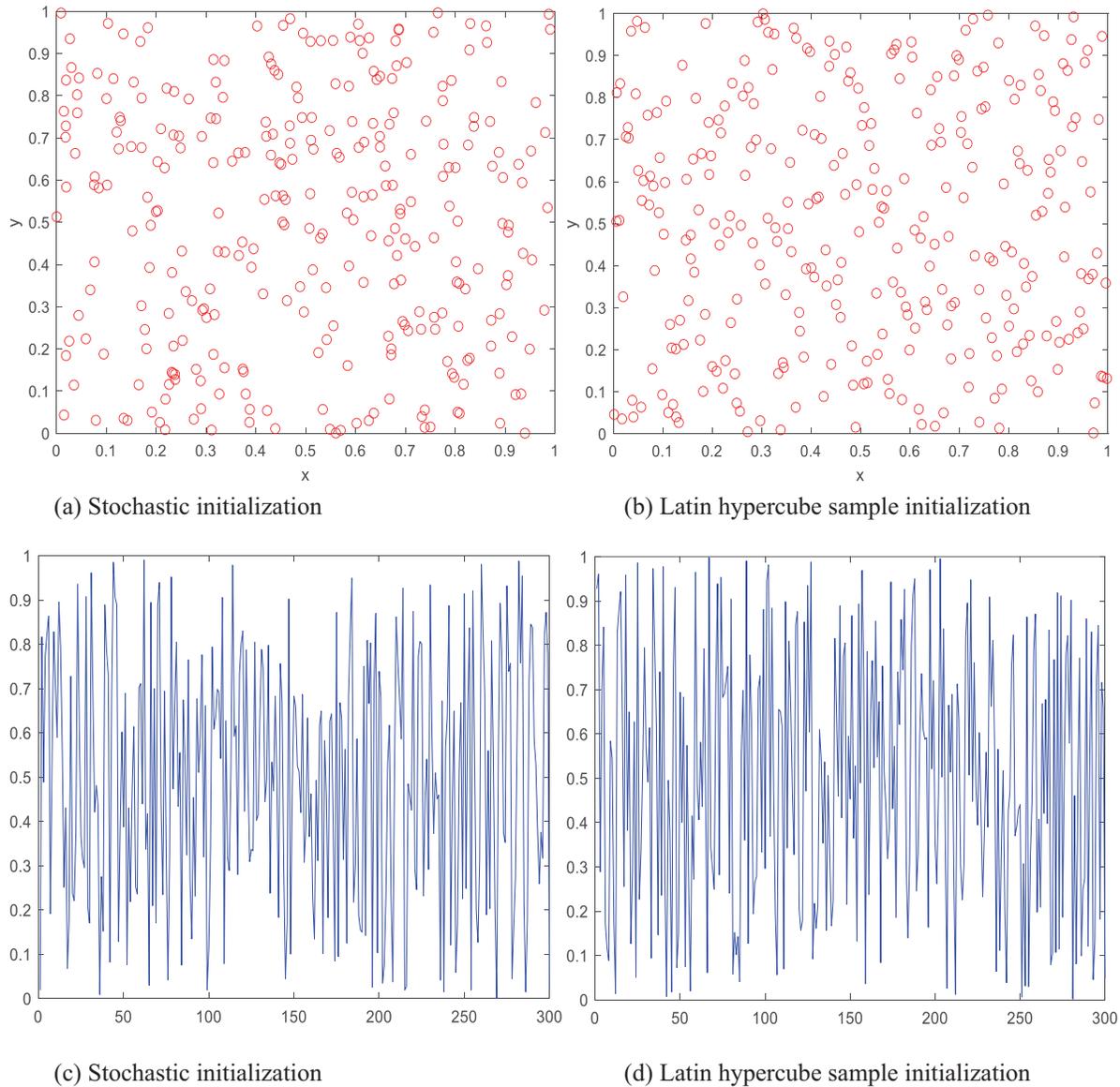


Figure 1: Comparison of the two initialization techniques

4.2 SCSO Strategy

Inspired by the foraging behavior of sand cats in desert environments, Amir Seyyedabbasi et al. introduced the Sand Cat Swarm Optimization [28]. Sand cats can efficiently locate and capture prey by exploiting their skill in detecting low-frequency noise. Like PSO, SCSO treats the prey position as the global optimal position and is divided into two mechanisms: search and attack.

4.2.1 Search Mechanism

Each sand cat utilizes its acute auditory sense to detect low-frequency vibrations below 2 kHz and locate prey throughout the exploration space. The specific modeling is as follows:

$$Pa_i(t+1) = r_a \cdot (GPabest(t) - rand(0,1) \cdot Pa_i(t)) \quad (7)$$

$$r_a = R_a \times rand(0,1) \quad (8)$$

$$R_a = S_M - \left(\frac{S_M \times t}{T} \right) \quad (9)$$

where r_a is the auditory sensitivity range of each sand cat. S_M is a constant with a value of 2, T is the upper limit on the count of iterations for the entire search process, and R_a denotes the general sensitivity scope of the sand cat's hearing, which linearly diminishes in value from 2 to 0 as the iterative process advances. Especially when $R_a = 0$, $r_a = 0$, at which point the most recent position of the sand cat will also be adjusted to 0, is also in the search space. Furthermore, an adaptive balance factor R is proposed, which ranges from $[0, 2]$ and whose value is given by the following formula:

$$R = 2 \times R_a \times rand(0,1) - R_a \quad (10)$$

4.2.2 Attack Mechanism

As the optimization procedure progresses, the sand cat in the exploration region automatically approaches the prey slowly and launches an attack on the targeted prey with the following formula:

$$Dist = |rand(0,1) \cdot GPabest(t) - Pa_i(t)| \quad (11)$$

$$Pa_i(t+1) = Pa_i(t) - r_a \cdot Dist \cdot \cos(\beta) \quad (12)$$

where $Dist$ is the magnitude of displacement between the sand cat's ideal position $GPabest(t)$ and the current position $Pa_i(t)$, β is an arbitrary angle that is between 0° and 360° . Through this mechanism, the orientation of the sand cat's movement is determined by β . At this point, each member of the colony moves in a different circular put-up search in space.

4.2.3 Transition Mechanism

The implementation and transition of the process of SCSO search and attack on prey are determined by the parameter R . In the initial phase of the iteration, the value of R_a is large; when $|R| > 1$, SCSO is in the exploring stage, and the sand cat explores broadly for potential targets within the exploration area. As the count of iterations grows, the value of R_a converges to 0 during the later phase of the iteration, at which point $|R| > 1$, SCSO is in the exploitation phase, and the sand cat initiates an attack on the prey it is aiming at. The diagram of sand cat search and attack prey is displayed in Fig. 2. At this time, the specific formula is described below:

$$Pa(t+1) = \begin{cases} GPabest(t) - Dist \cdot \cos(\beta) \cdot r_a, & |R| \leq 1; \text{ exploitation} \\ r_a \cdot (GPabest(t) - rand(0,1) \cdot Pa(t)), & |R| \geq 1; \text{ exploration} \end{cases} \quad (13)$$

Once the algorithm finishes the PSO position update phase, it implements the SCSO position update formula. This crucial step effectively addresses the issue of limited search scope seen in the original approach, significantly improving the algorithm's exploration capabilities and preventing it from getting trapped in local optima.

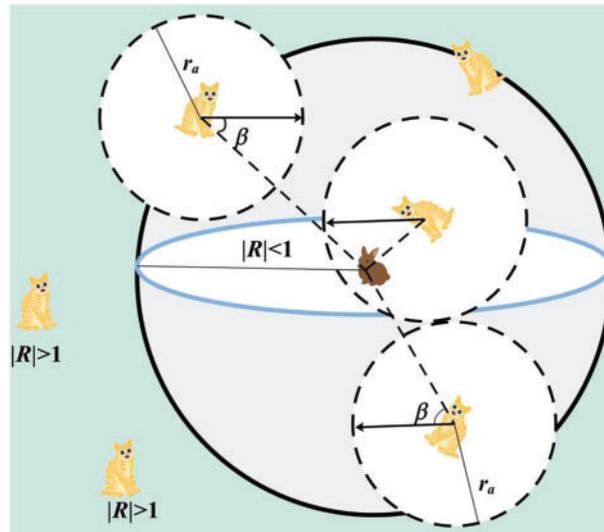


Figure 2: Diagram of sand cat search and attack prey

4.3 Roundup Search Strategy

Although the application of complex strategies enhances the accuracy of the solution, it may, to some extent, slow down the convergence speed of the algorithm. Drawing inspiration from how gray wolves round up food in the GWO [29], this paper refines the particle foraging exploration stage. Introducing the hierarchical idea of gray wolves, the leader chosen depending on the fitness size is nearer to the prey and is more capable of quickly recognizing the location of potential prey, and the leader leads the other particles to approach the target location gradually. That helps to improve exploration efficiency, thereby increasing the convergence speed of the algorithm. Usually, the number of leaders is kept in the top 10%. The specific modeling of individuals encircling prey is as follows:

$$Pa_m(t+1) = leader_m(t) - B_m \cdot dis_m, \quad m = 1, 2, \dots, Nu - leader \quad (14)$$

$$Pa(t+1) = \frac{\sum_{m=1}^{Nu-leader} pa_m(t+1)}{Nu - leader} \quad (15)$$

where *leader* refers to the leader in the population, and *Nu-leader* refers to the number of leaders. *dis* is the spatial separation between the leader and the individual's current. The specific equation is:

$$dis_m = |C_m \cdot leader_m(t) - pa(t+1)| \quad (16)$$

B and *C* are the two synergy coefficients, which are denoted as:

$$B_m = 2b \cdot r_3 - b \quad (17)$$

$$C_m = 2r_4 \quad (18)$$

$$b = 2 - l \cdot \left(\frac{t}{T}\right) \quad (19)$$

where *B* is designed to model the aggressive behavior of the particle toward its prey and its value is influenced by *b*. *b* exhibits a linear decline from 2 to 0 as the count of the iteration increases. *r*₃, *r*₄ are two random numbers in the range [0, 1]. Fig. 3 visualizes the particle updating its position according to the leader.

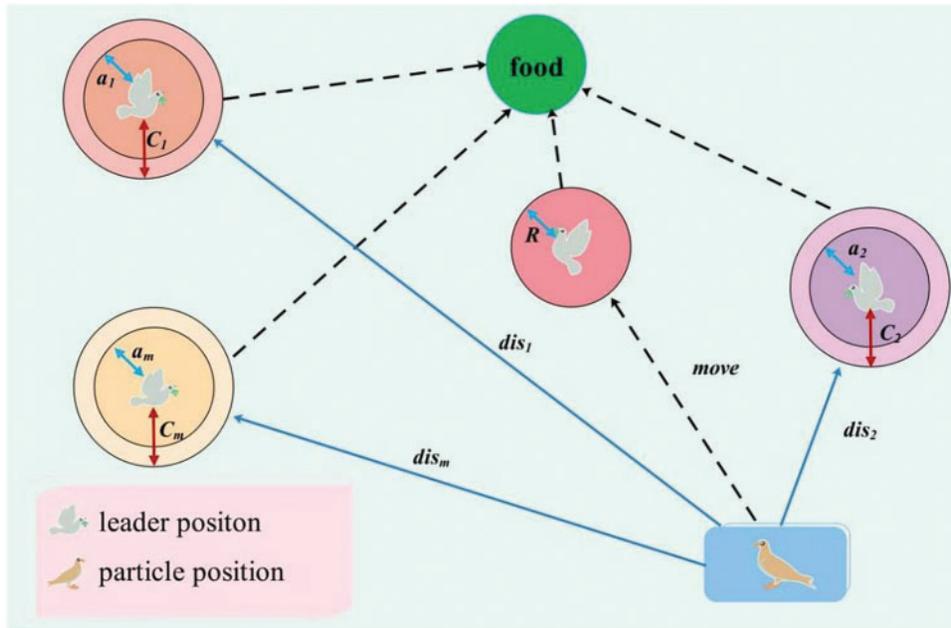


Figure 3: Diagram of the roundup search strategy

4.4 SCPSO

This section outlines the basic framework of SCPSO, where the PSO’s structure is reconstructed by adding the Latin hypercube sample strategy and the SCSO based on the roundup search strategy. Firstly, the Latin hypercube sampling strategy generates a population, and then the algorithm executes the PSO position update formula. Subsequently, the historical best position in PSO is passed as the initial point to SCSO, and SCSO’s position update is performed. Finally, the roundup search strategy will be integrated into the exploration phase of SCSO to improve the convergence speed of the algorithm.

The pseudo-code of SCPSO is also rendered in Algorithm 1. In addition, the flowchart in Fig. 4 visualizes more vividly the execution process of SCPSO.

Algorithm 1: SCPSO

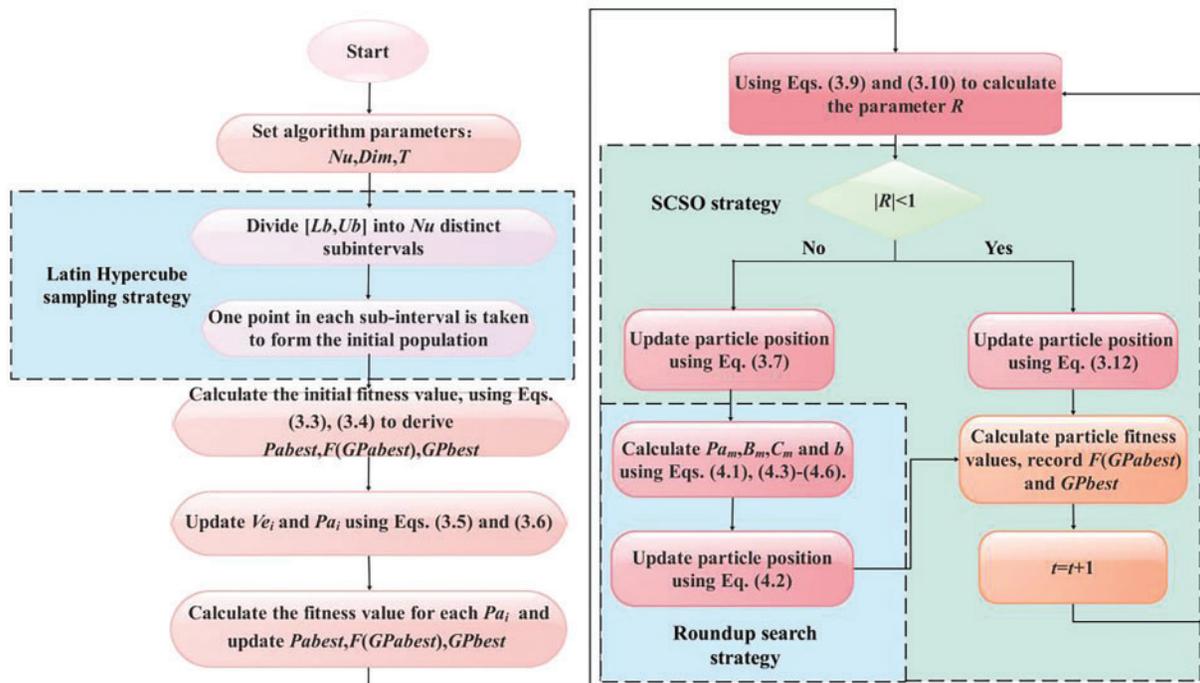
Set parameters: population scale (Nu), dimension (Dim), maximum iteration times (T), etc.
 Produce an initial population with Latin hypercube sampling strategy and initialize flight velocity.
 Compute the initial fitness value of particles, record the historical best position of particles $Pabest_i$, best fitness value $F(GPabest)$, global best position $GPabest$.

- 1: **While** ($t < T$)
 - 2: adjust the parameters like R_a, r_a, R, b .
 - 3: **for** $i = 1:Nu$
 - 4: $Ve_i(t+1) = \omega \cdot Ve_i(t) + a_1 \cdot r_1 \cdot (Pabest_i(t) - Pa_i(t)) + a_2 \cdot r_2 \cdot (GPabest_i(t) - Pa_i(t))$ // Updated flight velocity
 - 5: $Pa_i(t+1) = Pa_i(t) + Ve_i(t)$
 - 6: Calculate the particle fitness value, update $Pabest_i$, record $F(GPabest), GPabest$.
 - 7: **if** $|R| > 1$ //
 - 8: $Pa_i(t+1) = r_a \cdot (GPabest(t) - rand(0,1) \cdot Pa_i(t))$
-

(Continued)

Algorithm 1 (continued)

-
9. $Pa_m(t+1) = leader_m(t) - B_m \cdot dis_m, \quad m = 1, 2, \dots, Nu - leader // \text{Round-up}$
search strategy
 10. $Pa(t+1) = \frac{\sum_{m=1}^{Nu-leader} pa_m(t+1)}{Nu - leader}$
 11. **else**
 12. $Pa_i(t+1) = Pa_i(t) - r_a \cdot Dist \cdot \cos(\beta)$
 13. **end**
 14. Calculate per-particle fitness value $F(Pa_i)$
 15. **if** $F(Pa_i) < F(GPabest)$
 16. $F(GPabest) = F(Pa_i)$
 17. $GPabest = Pa_i$
 18. **end**
 19. **end**
 20. $t = t + 1$
 21. **end**
-

**Figure 4:** Flow chart of SCPSO**5 Trials and Analysis**

Within this part, a series of numerical trials are conducted on SCPSO to evaluate its performance fairly. Firstly, we conduct a performance analysis on various strategies using nine benchmark functions. Secondly, SCPSO is evaluated against other modified PSO algorithms on the 30-dimension CEC2020 test suite [30]

to test the efficiency of the strategies incorporated in this paper. Then, SCPSO is tested compared to other competitive algorithms on 10 and 20-dimension CEC2022 test suites [31]. Remarkably, the test functions in both the CEC2020 and CEC2022 test suites comprise a portion of the challenging test functions in the CEC2014 [32] and a portion of the CEC2017 [33] test suites. Moreover, they are all categorized into four parts: single-peak, fundamental, hybrid, and combined functions, encompassing a domain of -100 to 100 for all functions. Trials on such test suites are more adept at evaluating the proficiency of SCPSO in resolving diverse optimization issues.

To reflect the impartiality of the trials and eliminate the interference of chance, the population size was set to 50 and the maximum limit of iterations was 1000. All algorithms were run 20 times. Table 1 shows the built-in parameters of the algorithms. Each trial was executed in a standardized environment.

Table 1: Parameter settings for each algorithm

Algorithms	Parameters name	Parameters values
PSOGSA	Gravitational constant G	$[0, 100]$
GQPSO	Shrinkage-expansion coefficient β	$[0, 1]$
	Random parameter u, k	$[0, 1]$
	Gaussian distributed random number g	$[0, 1]$
	Positive constant q	5000
LFPSO	Offset parameter μ	$[0, \infty]$
	Scale parameter γ	$[0, \infty]$
	Skewness factor α	$[-1, 1]$
	Levy exponent β	$[0, 2]$
COSCSO	Cauchy mutation calculator a	$[0, 1]$
RSA	Exploring sensitive parameters β	0.1
	Evolutionary factor ES	Decrease randomly from 2 to -2
	Random number r_1	$[1, N]$
	Sensitive parameters α	0.1
WOA	Control factor m	Decreases linearly from 2 to 0
	Constant n	1
TLCO	scale parameter ξ	$[-1, 1]$
SCA	Constant a	2
	Control parameter r_1	$[0, 2]$
	Random parameter r_2	$[0, 2\pi]$
	Random parameter r_3	$[0, 2]$
HHO	Initial energy E_0	$[-1, 1]$
GJO	Runaway energy E	Reduction from 1.5 to 0
	Static parameter C_1	1.5
COA	Random integer I	$\{1, 2\}$

Furthermore, to effectively compare the efficacy of distinct algorithms, it is measured in terms of four common quantitative metrics [34]:

(1) Best value

$$Best = \{F_1(GPabest), F_2(GPabest), \dots, F_{Runs}(GPabest)\}$$

(2) Worst value

$$Worst = \max\{F_1(GPaworst), F_2(GPaworst), \dots, F_{Runs}(GPaworst)\}$$

(3) Average value

$$Average = \frac{\sum_{k=1}^{Runs} F_k(GPabest)}{Runs}$$

(4) Standard deviation

$$Std = \sqrt{\frac{1}{Runs - 1} \sum_{k=1}^{Runs} (F_k(GPabest) - Average)^2}$$

where $F_k(GPabest)$, $F_k(GPaworst)$ are the optimal and worst values derived from one run of the algorithm, respectively. These two metrics demonstrate the accuracy of the algorithm. Taking the average value provides an overall indication of algorithm competence and mitigates the influence of random factors. The standard deviation, on the other hand, mirrors the degree of discretization of the data set. The robustness of the algorithm can be measured accordingly.

5.1 Performance Analysis of Strategies

The performance analysis of strategies is of great significance in studying optimization algorithms. By quantifying a strategy's optimization effect, we can effectively evaluate its performance in terms of convergence speed, accuracy, and global search capability. In addition, the performance analysis can also evaluate the algorithm's applicability and provide a basis for selecting appropriate strategies in practical application scenarios. Building on the foundation of Particle Swarm Optimization (PSO), we introduce a single strategy: PSO + LHS, which integrates the Latin hypercube sampling technique. Additionally, PSO + RSA denotes the version of PSO that employs the Roundup Search Strategy. Lastly, PSO + SCSO refers to the variant of PSO enhanced with the SCSO methodology. In addition, PSO and SCPSO are to compare with them. Table 2 records the statistics of the mean and standard deviation of PSO and its variants. Fig. 5 shows their convergence curves. Comprehensive figures and tables show that compared with the original PSO, PSO + LHS, and PSO + SCSO have a noticeable improvement in convergence accuracy and stability, which indicates that the Latin hypercube sampling technique and SCSO effectively improve the solution accuracy of PSO. However, it is worth noting that compared with the fast convergence of PSO, they are slightly slower in convergence. That indicates that the algorithm sacrifices part of the solution speed while jumping out of the local optimal solution and actively searching for the global optimal solution. PSO + RSA has a limited improvement of the accuracy of PSO in most functions and even has a specific inhibiting effect in individual functions. However, PSO + RSA retains the advantage of PSO's fast convergence speed. Overall, SCPSO integrating three strategies is significantly better than the PSO variant with only a single strategy in most algorithms, both in terms of convergence accuracy and stability, and the convergence speed of SCPSO is moderate. That is mainly due to the cooperation between the Latin hypercube sampling technique and SCSO to improve the algorithm's accuracy. Adding the Roundup Search Strategy also improves the algorithm's overall convergence speed. However, the addition of the Roundup Search Strategy has a slight inhibiting

effect on the overall accuracy of the algorithm for individual functions. Therefore, SCPSO is generally the best choice, but for practical problems requiring higher accuracy, readers can also consider PSO + LHS + SCPSO.

Table 2: Performance analysis results

		PSO	PSO + LHS	PSO + RSA	PSO + SCPSO	SCPSO
F1	Average	3.015E + 03	1.157E - 04	1.001E + 03	1.508E - 03	2.619E - 07
	Std	4.844E + 03	1.247E - 04	3.162E + 03	2.432E - 03	2.415E - 07
F2	Average	1.103E + 02	2.253E + 00	1.215E + 01	3.859E - 02	3.765E - 03
	Std	6.945E + 01	1.087E + 00	7.610E + 00	3.953E - 02	4.338E - 03
F3	Average	1.785E + 04	3.840E + 01	7.037E + 03	1.598E - 01	1.801E - 04
	Std	1.344E + 04	2.586E + 01	4.878E + 03	3.234E - 01	3.270E - 04
F4	Average	1.811E + 01	3.163E - 01	1.658E + 01	4.178E - 02	1.186E - 04
	Std	4.744E + 00	5.837E - 02	3.060E + 00	2.915E - 02	7.414E - 05
F5	Average	8.012E + 06	6.227E + 01	9.347E + 03	4.933E + 01	4.871E + 01
	Std	2.531E + 07	2.426E + 01	2.836E + 04	4.882E - 01	3.873E - 02
F6	Average	6.318E + 03	8.697E - 02	1.044E + 03	1.200E + 01	6.693E + 00
	Std	7.225E + 03	1.336E - 01	3.615E + 03	4.275E - 01	6.148E + 00
F7	Average	1.259E + 01	3.053E + 00	2.144E + 00	4.372E - 03	2.259E - 03
	Std	2.005E + 01	1.959E + 00	8.605E - 01	2.808E - 03	1.708E - 03
F8	Average	-1.197E + 04	-1.149E + 04	-1.112E + 04	-1.279E + 04	-1.227E + 04
	Std	9.361E + 02	1.216E + 03	1.086E + 03	1.892E + 03	2.211E + 03
F9	Average	2.408E + 02	4.110E + 01	1.726E + 02	4.517E - 03	1.304E - 04
	Std	5.217E + 01	1.568E + 01	3.879E + 01	4.577E - 03	2.224E - 04

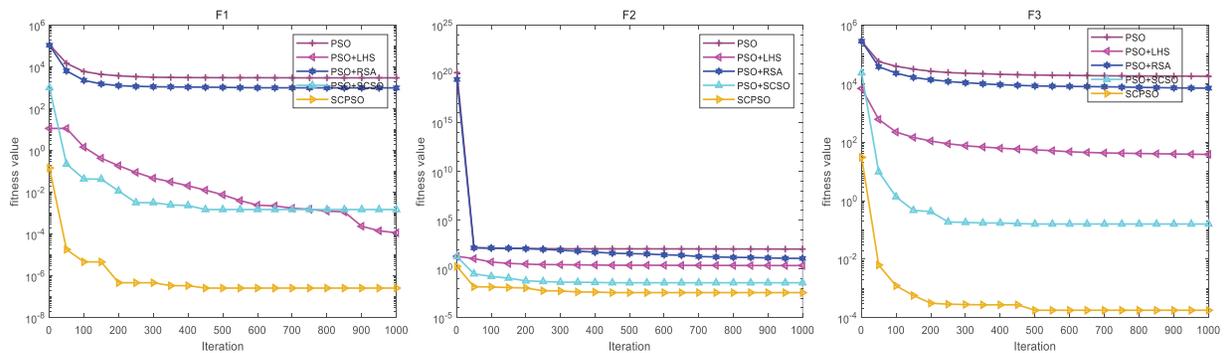


Figure 5: (Continued)

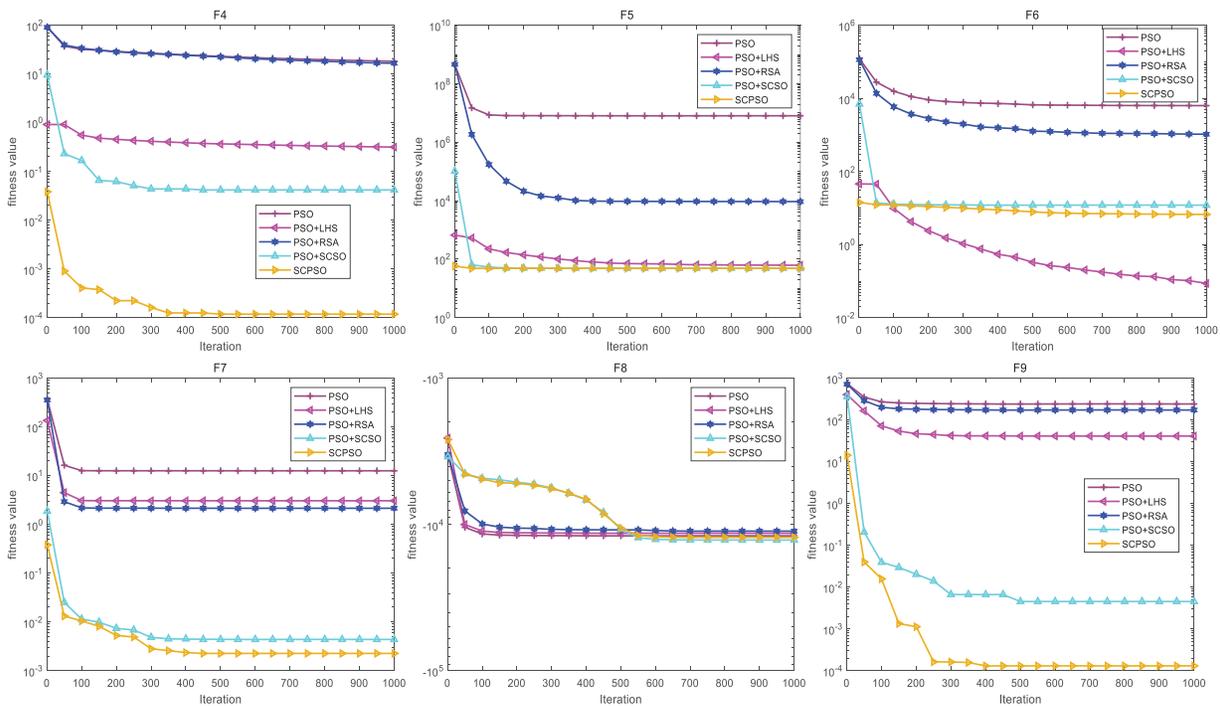


Figure 5: Convergence plot of SCPSO and PSO with a single strategy

5.2 Comparison and Analysis of SCPSO with Other Modified PSO

To assess the efficiency of SCPSO and investigate the effects of SCSO, the Latin hypercube sample strategy and roundup search strategy on PSO were used. In this part, SCPSO is tested against six other competitively modified versions of PSO and two improved versions of SCSO, in addition to comparing it with PSO on the 30-dimension CEC2020 test suite. The chosen algorithms are PSO_{GSA} [35], GQPSO [36], PSO_{DLS} [37], RDPSO [38], LFPSO [39], ESPSO [40], MSCSO [41], COSCSO [42].

In Table 3, the results of the ten algorithms on the 30-dimensional CEC2020 test suite are meticulously compiled. SCPSO stands out as the front-runner, securing the top position on seven test functions. While it doesn't quite outperform PSO_{GSA} on F5 and F7, SCPSO still shows remarkable consistency and strong performance compared to the other algorithms. Notably, SCPSO demonstrates a clear edge on F1, F3, F4, F6, F8, and F9, where its results are consistently smaller than those of the other algorithms, reflecting its exceptional stability. The table further presents the algorithms' average rankings and their final overall ranks, which are as follows: SCPSO > ESPSO > COSCSO > PSO_{GSA} > RDPSO > PSO > GQPSO > MSCSO > PSO_{DLS} > LFPSO. SCPSO maintains an impressive lead with an average rank of 1.3, while ESPSO follows closely in second place with an average rank of 3.0. PSO lags in sixth place with an average rank of 4.9, demonstrating that the enhancements brought by SCSO hybridization give SCPSO a substantial edge in performance. This stark contrast in rankings highlights the effectiveness of SCPSO in delivering superior optimization results, establishing it as the top performer in this suite.

Table 3: Comparison of findings between SCPSO and other modified PSO (CEC2020 30 dimension)

F	Results	Algorithm									
		PSOGSA	GQPSO	PSO_DLS	RDPSO	LFPSO	ESPSO	MCSO	COSCSO	PSO	SCPSO
F1	Average	7.212E + 07	2.666E + 10	6.215E + 10	7.775E + 08	5.074E + 10	5.135E + 07	1.666E + 10	2.614E + 05	1.208E + 10	1.776E + 04
	Std	7.094E + 07	1.068E + 09	6.571E + 09	8.976E + 08	9.955E + 09	2.296E + 08	6.338E + 09	4.957E + 05	9.302E + 09	4.745E + 04
	Best	8.251E + 06	2.441E + 10	4.639E + 10	6.433E + 07	3.343E + 10	3.887E + 02	9.133E + 09	2.131E + 03	9.838E + 03	5.875E + 02
	Worst	2.623E + 08	2.874E + 10	7.082E + 10	3.295E + 09	6.953E + 10	1.027E + 09	3.116E + 10	2.231E + 06	2.869E + 10	2.170E + 05
	Rank	4	8	10	5	9	3	7	2	6	1
F2	Average	5.538E + 03	8.028E + 03	8.751E + 03	5.623E + 03	9.751E + 03	5.267E + 03	7.512E + 03	5.531E + 03	5.378E + 03	5.005E + 03
	Std	7.760E + 02	2.917E + 02	3.000E + 02	6.009E + 02	5.041E + 02	7.193E + 02	1.175E + 03	7.719E + 02	8.986E + 02	8.501E + 02
	Best	4.410E + 03	7.397E + 03	8.190E + 03	4.761E + 03	8.787E + 03	3.468E + 03	5.333E + 03	4.297E + 03	3.688E + 03	3.549E + 03
	Worst	7.308E + 03	8.373E + 03	9.238E + 03	7.037E + 03	1.081E + 04	6.480E + 03	9.178E + 03	7.060E + 03	6.904E + 03	6.681E + 03
	Rank	5	8	9	6	10	2	7	4	3	1
F3	Average	1.122E + 03	1.189E + 03	2.369E + 03	1.033E + 03	1.504E + 03	8.421E + 02	1.278E + 03	1.129E + 03	1.098E + 03	8.099E + 02
	Std	1.182E + 02	2.064E + 01	1.272E + 02	7.236E + 01	5.817E + 01	3.142E + 01	6.449E + 01	9.161E + 01	1.403E + 02	1.790E + 01
	Best	9.295E + 02	1.156E + 03	2.094E + 03	9.195E + 02	1.393E + 03	7.737E + 02	1.162E + 03	9.616E + 02	9.284E + 02	7.753E + 02
	Worst	1.312E + 03	1.235E + 03	2.600E + 03	1.184E + 03	1.614E + 03	8.903E + 02	1.380E + 03	1.298E + 03	1.409E + 03	8.347E + 02
	Rank	5	7	10	3	9	2	8	6	4	1
F4	Average	1.934E + 03	1.900E + 03	1.243E + 05	1.976E + 03	1.906E + 03	1.906E + 03	1.900E + 03	1.900E + 03	1.925E + 03	1.900E + 03
	Std	1.030E + 01	0	7.092E + 04	8.117E + 01	8.526E + 00	1.914E + 00	0	0	9.731E + 00	0
	Best	1.917E + 03	1.900E + 03	9.426E + 03	1.921E + 03	1.900E + 03	1.903E + 03	1.900E + 03	1.900E + 03	1.914E + 03	1.900E + 03
	Worst	1.955E + 03	1.900E + 03	2.444E + 05	2.253E + 03	1.922E + 03	1.911E + 03	1.900E + 03	1.900E + 03	1.950E + 03	1.900E + 03
	Rank	4	1	7	5	3	2	1	1	6	1
F5	Average	2.766E + 05	3.704E + 07	3.432E + 07	4.073E + 06	9.106E + 07	8.777E + 05	1.104E + 08	1.047E + 06	3.236E + 06	6.104E + 05
	Std	2.533E + 05	1.011E + 07	1.339E + 07	2.958E + 06	5.663E + 07	9.092E + 05	1.170E + 08	6.192E + 05	6.753E + 06	1.165E + 06
	Best	6.889E + 04	1.941E + 07	6.977E + 06	3.604E + 05	2.184E + 07	2.034E + 04	9.429E + 06	1.828E + 05	6.852E + 04	5.412E + 04
	Worst	1.038E + 06	6.282E + 07	5.628E + 07	1.141E + 07	2.424E + 08	3.476E + 06	3.871E + 08	2.335E + 06	3.076E + 07	5.266E + 06
	Rank	1	8	7	6	9	3	10	4	5	2
F6	Average	2.680E + 03	3.417E + 03	4.613E + 03	2.487E + 03	5.116E + 03	2.392E + 03	4.625E + 03	2.749E + 03	2.726E + 03	2.104E + 03
	Std	4.262E + 02	2.132E + 02	1.798E + 02	3.435E + 02	8.201E + 02	3.151E + 02	1.564E + 03	3.643E + 02	4.812E + 02	1.683E + 02
	Best	2.029E + 03	2.765E + 03	4.239E + 03	1.895E + 03	4.045E + 03	1.932E + 03	2.771E + 03	2.119E + 03	2.048E + 03	1.833E + 03
	Worst	3.889E + 03	3.630E + 03	4.918E + 03	3.254E + 03	6.917E + 03	3.024E + 03	7.594E + 03	3.486E + 03	3.627E + 03	2.476E + 03
	Rank	4	7	8	3	10	2	9	6	5	1
F7	Average	9.730E + 04	9.959E + 06	9.191E + 06	7.051E + 05	3.737E + 07	9.839E + 05	6.068E + 07	4.423E + 05	6.377E + 05	1.943E + 05
	Std	4.814E + 04	3.356E + 06	3.246E + 06	5.496E + 05	2.260E + 07	2.256E + 06	6.759E + 07	3.673E + 05	1.239E + 06	1.868E + 05
	Best	2.584E + 04	3.514E + 06	2.873E + 06	8.220E + 04	3.207E + 06	1.957E + 04	1.724E + 06	9.441E + 04	1.785E + 04	2.669E + 04
	Worst	1.919E + 05	1.534E + 07	1.600E + 07	1.886E + 06	9.097E + 07	8.980E + 06	2.046E + 08	1.253E + 06	5.640E + 06	8.059E + 05
	Rank	1	8	7	5	9	6	10	3	4	2

(Continued)

Table 3 (continued)

F	Results	Algorithm									
		PSOGSA	GQPSO	PSO_DLS	RDPSO	LFPSO	ESPSO	MSCSO	COSCSO	PSO	SCPSO
F8	Average	6.877E + 03	5.462E + 03	8.832E + 03	6.112E + 03	1.030E + 04	6.094E + 03	7.830E + 03	3.733E + 03	6.249E + 03	2.316E + 03
	Std	1.287E + 03	2.416E + 02	6.734E + 02	1.981E + 03	1.424E + 03	1.225E + 03	2.644E + 03	2.253E + 03	1.579E + 03	3.502E + 01
	Best	2.427E + 03	4.979E + 03	7.738E + 03	2.496E + 03	6.574E + 03	2.300E + 03	3.108E + 03	2.301E + 03	2.463E + 03	2.300E + 03
	Worst	8.917E + 03	5.876E + 03	9.923E + 03	8.404E + 03	1.160E + 04	7.719E + 03	1.152E + 04	7.816E + 03	8.170E + 03	2.418E + 03
	Rank	7	3	9	5	10	4	8	2	6	1
F9	Average	3.185E + 03	3.498E + 03	3.560E + 03	3.071E + 03	3.801E + 03	3.159E + 03	4.328E + 03	3.079E + 03	3.113E + 03	2.928E + 03
	Std	1.235E + 02	3.481E + 01	3.550E + 01	7.371E + 01	3.453E + 02	6.814E + 01	2.550E + 02	7.999E + 01	7.142E + 01	3.461E + 01
	Best	3.003E + 03	3.424E + 03	3.492E + 03	2.944E + 03	3.262E + 03	3.014E + 03	3.930E + 03	2.953E + 03	2.975E + 03	2.858E + 03
	Worst	3.423E + 03	3.540E + 03	3.637E + 03	3.195E + 03	4.429E + 03	3.316E + 03	4.775E + 03	3.314E + 03	3.246E + 03	2.991E + 03
	Rank	6	7	8	2	9	5	10	3	4	1
F10	Average	2.964E + 03	3.433E + 03	7.599E + 03	3.052E + 03	5.437E + 03	2.894E + 03	3.388E + 03	2.919E + 03	3.117E + 03	2.910E + 03
	Std	4.065E + 01	5.049E + 01	6.192E + 02	7.877E + 01	6.827E + 02	1.154E + 01	2.654E + 02	2.680E + 01	2.455E + 02	1.839E + 01
	Best	2.898E + 03	3.314E + 03	5.893E + 03	2.937E + 03	3.994E + 03	2.884E + 03	3.124E + 03	2.884E + 03	2.890E + 03	2.885E + 03
	Worst	3.082E + 03	3.523E + 03	8.561E + 03	3.245E + 03	6.530E + 03	2.931E + 03	3.986E + 03	2.962E + 03	3.743E + 03	2.956E + 03
	Rank	4	8	10	5	9	1	7	3	6	2
Average rank		4.1	6.5	8.5	4.5	8.7	3.0	7.7	3.4	4.9	1.3
Final rank		4	7	9	5	10	2	8	3	6	1

The Wilcoxon rank sum test [43], also known as the Wilcoxon signed rank test or Wilcoxon-Mann-Whitney test, is employed to compare two independent samples to see if there is a difference in their median. The Wilcoxon rank sum test p -value at the 95% significance level ($\alpha = 0.05$) with SCPSO as the control is tabulated in Table 4. Synthesizing the four significant indicators mentioned above, PSOGSA was significantly more potent than PSO on F5 and F7, and COSCSO was significantly superior to SCPSO on F10. ESPSO and PSO do not differ significantly from SCPSO at F2 and F5, respectively. Based on the data, SCPSO exhibits a significant superiority over the six modified versions of PSO and demonstrates a remarkable difference compared to the two enhanced SCSOs.

Table 4: Wilcoxon rank sum test p -values for other modified PSO (CEC2020 30 dimension)

F	Algorithm								
	PSOGSA	GQPSO	PSO_DLS	RDPSO	LFPSO	ESPSO	MSCSO	COSCSO	PSO
F1	3.187E - 09	1.596E - 10	3.187E - 09	7.577E - 06	1.918E - 07				
F2	3.372E - 02	3.187E - 09	3.187E - 09	1.548E - 02	3.187E - 09	2.977E - 01	1.201E - 06	4.986E - 02	3.277E - 05
F3	3.187E - 09	1.349E - 03	3.187E - 09	3.187E - 09	3.187E - 09				
F4	8.007E - 09	NAN	8.007E - 09	8.007E - 09	8.007E - 09	8.007E - 09	NAN	NAN	8.007E - 09
F5	2.390E - 02	3.187E - 09	3.187E - 09	6.610E - 05	3.187E - 09	3.187E - 09	9.173E - 08	9.173E - 08	7.557E - 01
F6	3.987E - 06	3.187E - 09	3.187E - 09	3.048E - 04	3.187E - 09	1.782E - 03	3.187E - 09	5.227E - 07	6.674E - 06
F7	4.112E - 02	3.187E - 09	3.187E - 09	6.868E - 04	3.187E - 09	3.187E - 09	3.187E - 09	1.667E - 02	1.349E - 03
F8	3.187E - 09	1.201E - 06	3.187E - 09	8.782E - 03	3.187E - 09				
F9	3.187E - 09	3.187E - 09	3.187E - 09	2.960E - 07	3.187E - 09	3.187E - 09	3.187E - 09	1.431E - 07	1.235E - 07
F10	3.987E - 06	3.187E - 09	3.187E - 09	7.898E - 08	3.187E - 09	4.601E - 04	3.187E - 09	5.250E - 01	5.255E - 05
+ / = / -	2/0/8	0/0/9	0/0/10	0/0/10	0/0/10	0/1/9	0/0/9	1/0/8	0/1/9

To better visualize the convergence performance of SCPSO during the iteration process, ten algorithms are qualitatively analyzed by the curve convergence plot given in Fig. 6. On observing the plot, the red curves on F1, F2, F3, F6, F7, F8, and F9 have distinctly lower minimums than those of the other modified algorithms. Moreover, except for F4 and F10, the convergence speed of SCPSO is rapidly accelerated when the iteration count reaches 400, and the precision is improved, which further suggests that it is wise to add SCSO to PSO.

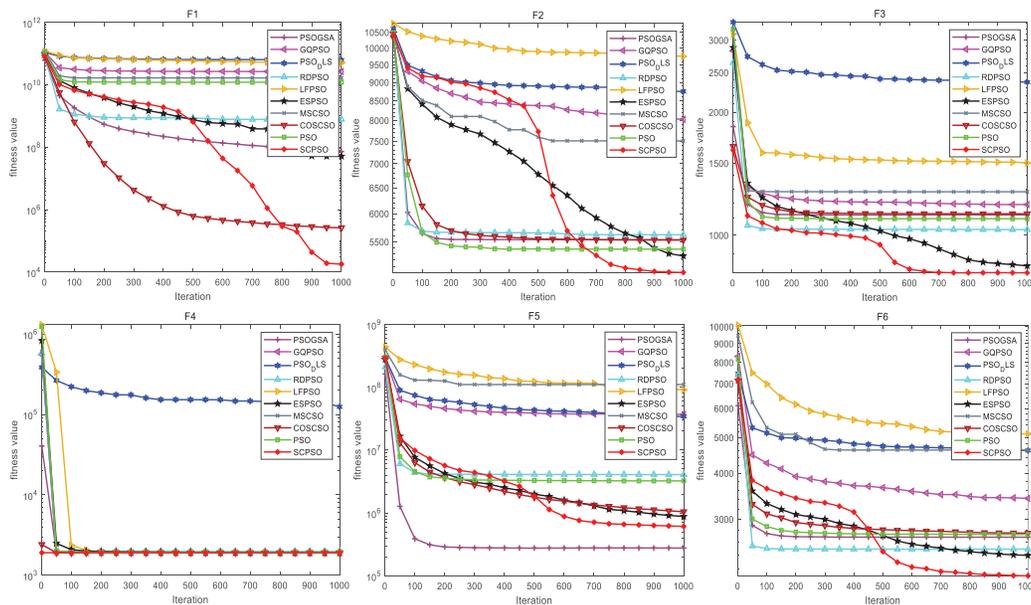


Figure 6: (Continued)

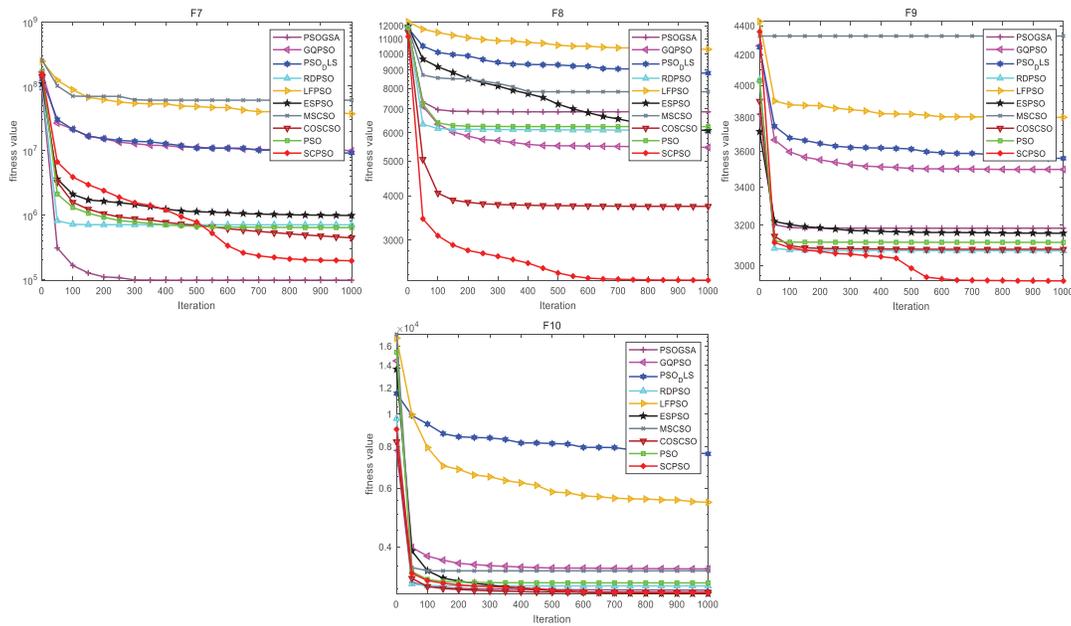


Figure 6: Convergence plot of SCPSO with other modified PSO (CEC2020 30 dimension)

Boxplot is a commonly available data visualization tool to display a data collection’s distribution and statistical characteristics. Fig. 7 exhibits a boxplot of SCPSO with other algorithms. Upon observation, except for F2, SCPSO has the smallest width of a rectangular box compared to other algorithms. That implies that SCPSO has a more centralized data distribution with a smaller range of outliers. In addition, SCPSO has the fewest outliers on all the test functions, which reveals that SCPSO performs more consistently in terms of performance and has stronger robustness compared to other modified algorithms. These findings provide us with more information about the advantages of SCPSO in data processing and optimization problems and give us more confidence to apply it in real-world scenarios.

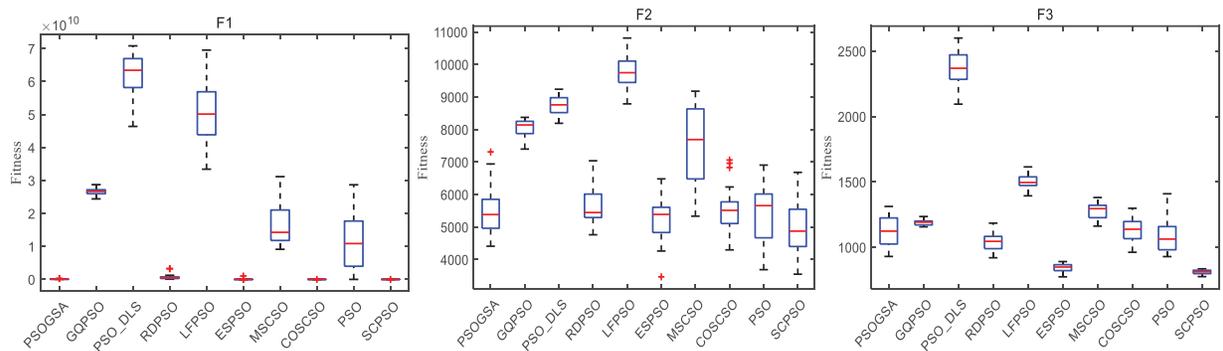


Figure 7: (Continued)

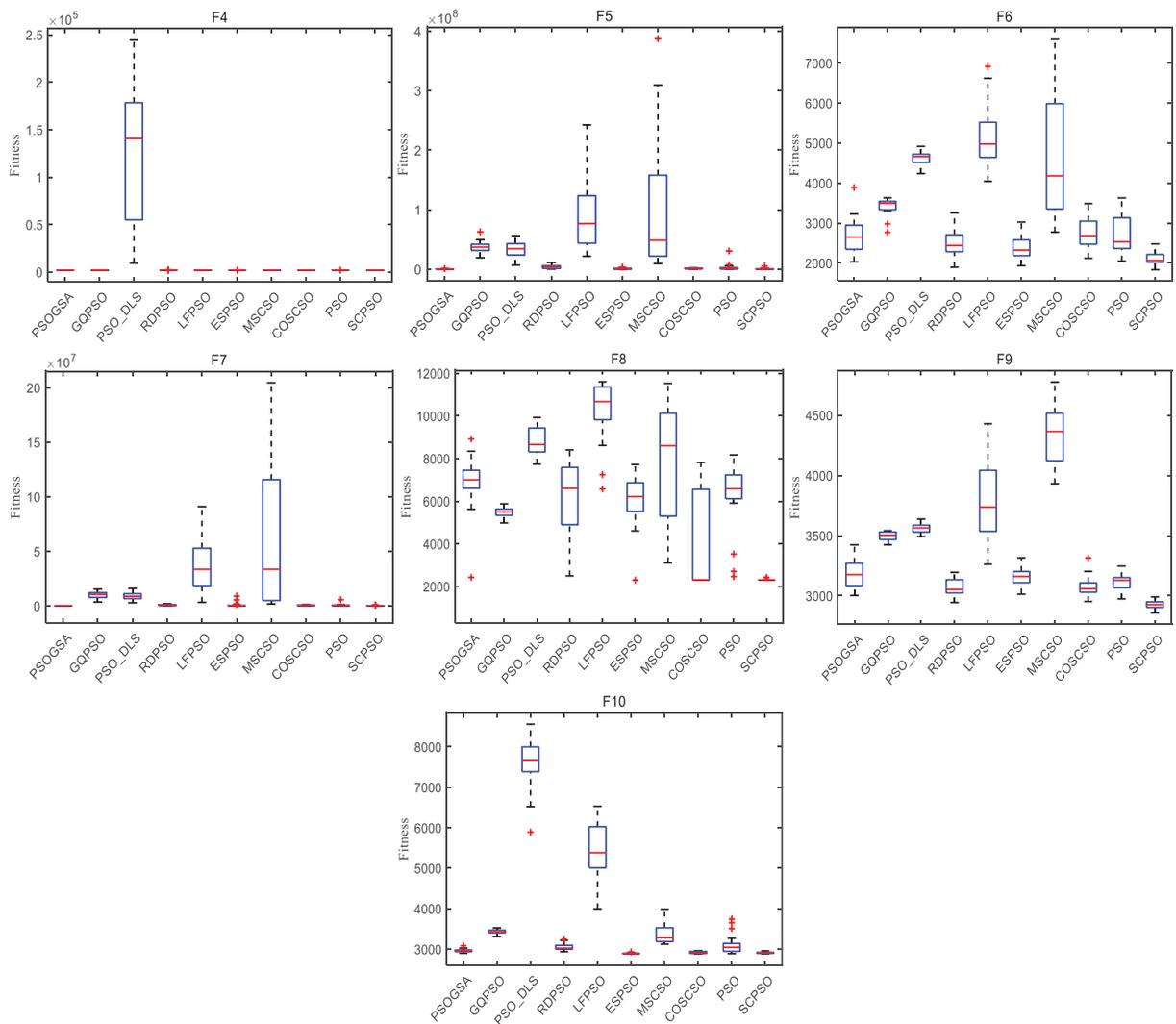
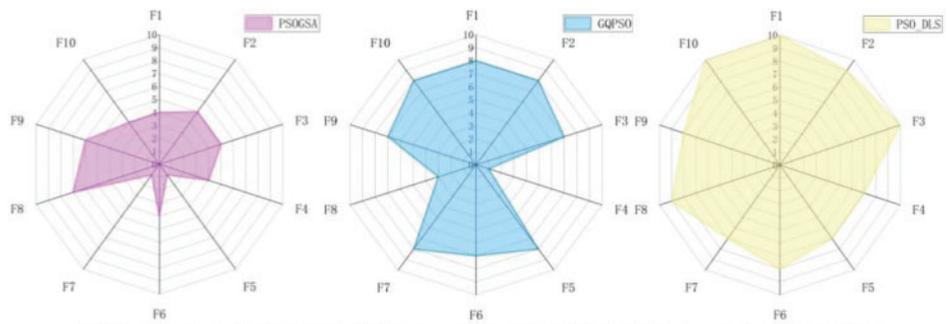
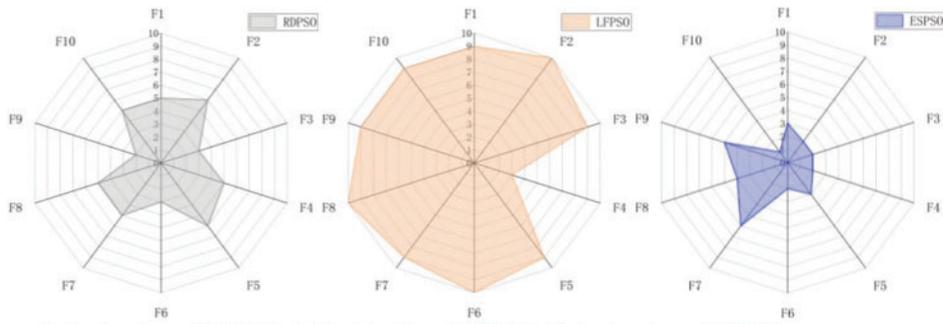


Figure 7: Boxplot of SCPSO with other modified PSO (CEC2020 30 dimension)

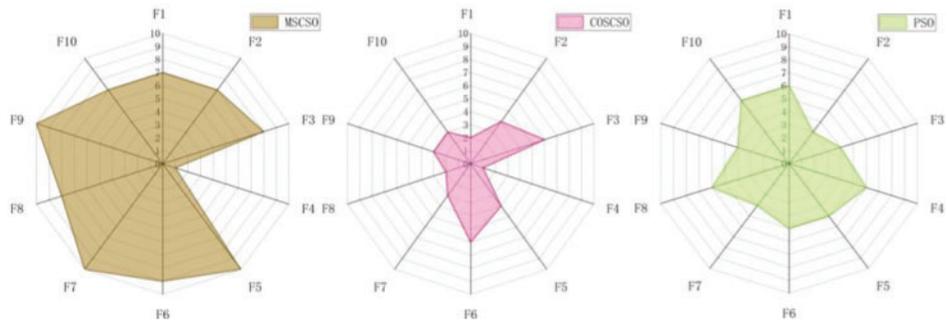
The following additions and summaries can be drawn by viewing the radar plot in Fig. 8 based on the rank of SCPSO vs. other modified PSO algorithms in the 30-dimension CEC2020 test suite. In the first place, it is evident from the radar plot that SCPSO encloses the smallest area, demonstrating that SCPSO performs best relative to other algorithms when evaluating the performance of multiple variables comprehensively. Secondly, it can be observed in the radar plot that the shaded area enclosed by the ESPSO is ranked second, but there is still a gap compared with the SCPSO.



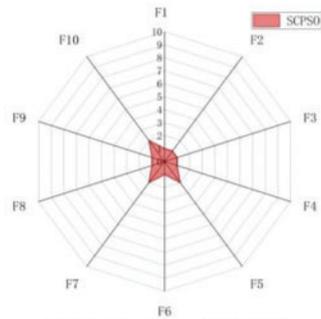
(a) Radar plot of PSOGSA (b) Radar plot of GQPSO (c) Radar plot of PSO_DLS



(d) Radar plot of RDPSO (e) Radar plot of LFPSO (f) Radar plot of ESPSO



(g) Radar plot of MSCSO (h) Radar plot of COSCSO (i) Radar plot of PSO



(j) Radar plot of SCPSO

Figure 8: Radar plot of SCPSO with other modified PSO (CEC2020 30 dimension)

In summary, compared with other PSO variants, SCPSO has better convergence accuracy and stability, which is its most significant advantage. The rank sum test results also show that SCPSO outperforms other variants in accuracy. PSOGSA and GQPSO also performed well in multiple functions, achieving the second and third final rankings. The performance of LFPSO, PSO-DLS, and MSCSO could be improved, resulting in a lower final ranking.

5.3 Comparison and Analysis of SCPSO with Other Modified Algorithms

To more extensively evaluate the effectiveness of SCPSO in addressing intricate issues, researchers executed experiments in the newer and more challenging CEC2022 test suite involving two dimensions. In addition, SCPSO was compared with several competitive experimental algorithms besides PSO, including the Reptile search algorithm [44], Whale optimization algorithm (WOA) [45], Termite life cycle optimizer (TLCO) [46], Sine Cosine Algorithm (SCA) [47], Sand Cat Swarm Optimization (SCSO), Harris Hawk Optimizer (HHO) [48], Golden Jackal Optimizer (GJO) [49], and Coati Optimization Algorithm (COA) [50]. These algorithms represent the current cutting-edge methods in the field, so comparing them can reveal the advantages and characteristics of SCPSO.

Table 5 contains a collection of statistical findings for SCPSO and other algorithms on 10 dimensions. SCPSO ranks highest on 75% of the functions, which means that it performs excellently in solving these problems. However, on F1, although SCPSO and PSO have the same average, PSO has a minor standard deviation, implying that PSO slightly outperforms SCPSO regarding stability. Interestingly, on F6 and F10, SCPSO acts differently from the other functions, ranking 6th and 5th, respectively, and not dominating in standard deviation; there may be scope for improvement. Further research and analysis are necessary to determine the causes and find ways to improve.

Table 5: Comparison of findings of SCPSO with other algorithms (CEC2022 10 dimension)

F	Results	Algorithm									
		RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO	SCPSO
F1	Average	8.383E + 03	1.441E + 04	1.657E + 03	1.200E + 03	1.357E + 03	3.010E + 02	2.447E + 03	6.449E + 03	3.000E + 02	3.000E + 02
	Std	2.842E + 03	6.113E + 03	1.532E + 03	7.779E + 02	1.977E + 03	5.300E - 01	1.917E + 03	1.535E + 03	2.916E - 14	5.377E - 14
	Best	4.714E + 03	4.659E + 03	5.531E + 02	5.796E + 02	3.423E + 02	3.003E + 02	4.563E + 02	2.555E + 03	3.000E + 02	3.000E + 02
	Worst	1.449E + 04	3.011E + 04	6.392E + 03	3.198E + 03	6.389E + 03	3.023E + 02	6.337E + 03	8.593E + 03	3.000E + 02	3.000E + 02
	Rank	10	6	7	4	5	3	8	9	1	2
F2	Average	7.150E + 02	4.446E + 02	5.057E + 02	4.594E + 02	4.318E + 02	4.330E + 02	4.421E + 02	1.218E + 03	4.319E + 02	4.148E + 02
	Std	2.710E + 02	7.501E + 01	1.056E + 02	1.477E + 01	3.650E + 01	3.389E + 01	2.453E + 01	4.537E + 02	2.710E + 01	1.890E + 01
	Best	5.060E + 02	4.003E + 02	4.122E + 02	4.240E + 02	4.001E + 02	4.006E + 02	4.083E + 02	6.821E + 02	4.080E + 02	4.082E + 02
	Worst	1.632E + 03	7.364E + 02	9.207E + 02	4.931E + 02	5.015E + 02	4.902E + 02	4.946E + 02	2.287E + 03	4.876E + 02	4.701E + 02
	Rank	9	6	8	7	2	4	5	10	3	1
F3	Average	6.463E + 02	6.349E + 02	6.227E + 02	6.190E + 02	6.195E + 02	6.261E + 02	6.070E + 02	6.442E + 02	6.088E + 02	6.005E + 02
	Std	7.274E + 00	1.309E + 01	5.247E + 00	3.992E + 00	1.356E + 01	1.182E + 01	4.743E + 00	8.074E + 00	4.871E + 00	3.340E - 01
	Best	6.292E + 02	6.158E + 02	6.130E + 02	6.136E + 02	6.040E + 02	6.086E + 02	6.002E + 02	6.296E + 02	6.022E + 02	6.000E + 02
	Worst	6.628E + 02	6.581E + 02	6.326E + 02	6.273E + 02	6.505E + 02	6.522E + 02	6.161E + 02	6.586E + 02	6.170E + 02	6.012E + 02
	Rank	10	8	6	4	5	7	2	9	3	1
F4	Average	8.490E + 02	8.289E + 02	8.349E + 02	8.403E + 02	8.266E + 02	8.249E + 02	8.257E + 02	8.453E + 02	8.278E + 02	8.135E + 02
	Std	6.274E + 00	9.061E + 00	4.561E + 00	7.519E + 00	8.913E + 00	6.984E + 00	8.147E + 00	7.604E + 00	9.971E + 00	5.595E + 00
	Best	8.380E + 02	8.130E + 02	8.271E + 02	8.220E + 02	8.080E + 02	8.140E + 02	8.107E + 02	8.319E + 02	8.099E + 02	8.040E + 02
	Worst	8.614E + 02	8.468E + 02	8.430E + 02	8.525E + 02	8.366E + 02	8.419E + 02	8.431E + 02	8.605E + 02	8.490E + 02	8.249E + 02
	Rank	10	6	7	8	4	2	3	9	5	1
F5	Average	1.478E + 03	1.614E + 03	1.033E + 03	1.010E + 03	1.002E + 03	1.290E + 03	9.586E + 02	1.355E + 03	9.791E + 02	9.009E + 02
	Std	1.250E + 02	3.641E + 02	1.258E + 02	7.480E + 01	9.998E + 01	1.756E + 02	8.739E + 01	1.840E + 02	1.200E + 02	7.226E - 01
	Best	1.273E + 03	1.028E + 03	9.325E + 02	9.509E + 02	9.031E + 02	9.808E + 02	9.018E + 02	1.095E + 03	9.000E + 02	9.001E + 02
	Worst	1.713E + 03	2.205E + 03	1.427E + 03	1.255E + 03	1.245E + 03	1.536E + 03	1.305E + 03	1.708E + 03	1.292E + 03	9.033E + 02
	Rank	9	10	6	5	4	7	2	8	3	1
F6	Average	5.449E + 07	3.653E + 03	1.196E + 06	1.914E + 06	4.419E + 03	4.209E + 03	7.396E + 03	5.019E + 06	5.010E + 03	6.035E + 03
	Std	2.268E + 07	1.721E + 03	9.297E + 05	1.540E + 06	2.029E + 03	1.783E + 03	2.093E + 03	7.500E + 06	2.287E + 03	2.404E + 03
	Best	1.708E + 07	1.867E + 03	4.302E + 04	3.538E + 05	2.005E + 03	2.203E + 03	3.347E + 03	1.624E + 04	2.001E + 03	1.905E + 03
	Worst	1.146E + 08	7.343E + 03	4.280E + 06	6.629E + 06	8.111E + 03	8.090E + 03	1.080E + 04	2.827E + 07	8.143E + 03	8.169E + 03
	Rank	10	1	8	9	4	3	7	5	4	6
F7	Average	2.124E + 03	2.063E + 03	2.064E + 03	2.057E + 03	2.043E + 03	2.047E + 03	2.045E + 03	2.078E + 03	2.036E + 03	2.023E + 03
	Std	2.329E + 01	2.386E + 01	1.054E + 01	7.203E + 00	1.970E + 01	1.703E + 01	2.219E + 01	1.341E + 01	1.378E + 01	2.833E + 00
	Best	2.086E + 03	2.026E + 03	2.045E + 03	2.045E + 03	2.009E + 03	2.023E + 03	2.002E + 03	2.045E + 03	2.021E + 03	2.020E + 03
	Worst	2.168E + 03	2.119E + 03	2.077E + 03	2.072E + 03	2.099E + 03	2.085E + 03	2.093E + 03	2.102E + 03	2.067E + 03	2.031E + 03
	Rank	10	7	8	6	3	5	4	9	2	1

(Continued)

Table 5 (continued)

F	Results	Algorithm									
		RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO	SCPSO
F8	Average	2.252E + 03	2.232E + 03	2.231E + 03	2.232E + 03	2.227E + 03	2.231E + 03	2.227E + 03	2.238E + 03	2.230E + 03	2.219E + 03
	Std	1.298E + 01	5.141E + 00	2.814E + 00	2.963E + 00	2.709E + 00	8.931E + 00	4.273E + 00	1.225E + 01	2.756E + 01	7.271E + 00
	Best	2.234E + 03	2.224E + 03	2.222E + 03	2.227E + 03	2.223E + 03	2.223E + 03	2.215E + 03	2.226E + 03	2.221E + 03	2.202E + 03
	Worst	2.280E + 03	2.242E + 03	2.234E + 03	2.238E + 03	2.233E + 03	2.264E + 03	2.235E + 03	2.275E + 03	2.346E + 03	2.229E + 03
	Rank	10	8	5	7	2	6	3	9	4	1
F9	Average	2.718E + 03	2.562E + 03	2.626E + 03	2.568E + 03	2.569E + 03	2.551E + 03	2.588E + 03	2.727E + 03	2.555E + 03	2.529E + 03
	Std	5.038E + 01	5.002E + 01	3.530E + 01	1.898E + 01	3.951E + 01	4.438E + 01	3.672E + 01	3.064E + 01	4.789E + 01	2.721E - 01
	Best	2.631E + 03	2.530E + 03	2.566E + 03	2.538E + 03	2.529E + 03	2.529E + 03	2.529E + 03	2.670E + 03	2.529E + 03	2.529E + 03
	Worst	2.807E + 03	2.697E + 03	2.695E + 03	2.622E + 03	2.640E + 03	2.676E + 03	2.643E + 03	2.792E + 03	2.697E + 03	2.530E + 03
	Rank	9	4	8	5	6	2	7	10	3	1
F10	Average	2.617E + 03	2.601E + 03	2.539E + 03	2.502E + 03	2.551E + 03	2.591E + 03	2.556E + 03	2.622E + 03	2.576E + 03	2.567E + 03
	Std	9.053E + 01	2.914E + 02	6.296E + 01	7.793E - 01	6.376E + 01	6.912E + 01	6.364E + 01	1.283E + 02	1.048E + 02	5.594E + 01
	Best	2.511E + 03	2.500E + 03	2.502E + 03	2.501E + 03	2.500E + 03	2.501E + 03	2.500E + 03	2.516E + 03	2.501E + 03	2.500E + 03
	Worst	2.765E + 03	3.812E + 03	2.662E + 03	2.504E + 03	2.641E + 03	2.668E + 03	2.633E + 03	3.054E + 03	2.929E + 03	2.620E + 03
	Rank	9	8	2	1	3	7	4	10	6	5
F11	Average	3.196E + 03	2.838E + 03	2.793E + 03	2.770E + 03	2.754E + 03	2.730E + 03	2.861E + 03	3.380E + 03	2.943E + 03	2.666E + 03
	Std	3.649E + 02	1.814E + 02	3.067E + 01	6.455E + 00	1.685E + 02	8.859E + 01	1.905E + 02	3.270E + 02	2.453E + 02	1.700E + 02
	Best	2.841E + 03	2.606E + 03	2.757E + 03	2.756E + 03	2.601E + 03	2.604E + 03	2.722E + 03	2.920E + 03	2.600E + 03	2.600E + 03
	Worst	4.420E + 03	3.216E + 03	2.877E + 03	2.785E + 03	3.191E + 03	2.912E + 03	3.210E + 03	4.035E + 03	3.413E + 03	3.222E + 03
	Rank	9	6	5	4	3	2	7	10	8	1
F12	Average	2.946E + 03	2.888E + 03	2.916E + 03	2.869E + 03	2.868E + 03	2.900E + 03	2.869E + 03	2.922E + 03	2.869E + 03	2.863E + 03
	Std	6.525E + 01	2.666E + 01	2.092E + 01	1.969E + 00	1.036E + 01	3.900E + 01	7.989E + 00	2.201E + 01	5.142E + 00	7.766E - 01
	Best	2.876E + 03	2.864E + 03	2.884E + 03	2.866E + 03	2.859E + 03	2.865E + 03	2.861E + 03	2.887E + 03	2.864E + 03	2.862E + 03
	Worst	3.125E + 03	2.950E + 03	2.975E + 03	2.873E + 03	2.911E + 03	3.014E + 03	2.898E + 03	2.960E + 03	2.888E + 03	2.865E + 03
	Rank	10	6	8	3	2	7	5	9	4	1
Average rank		9.58	6.33	6.50	5.12	3.58	4.58	4.75	8.92	3.83	1.83
Result		10	7	8	6	2	4	5	9	3	1

What is more, the last part of [Table 5](#) also counts the overall average rank, and judging from the statistics, the rank is from high to low: SCPSO > SCSO > PSO > HHO > GJO > SCA > WOA > TLCO > COA > RSA. As a result, SCPSO takes the lead with an average rank of 1.83. SCSO takes second place with an average rank of 3.58.

In addition, [Table 6](#) provides the p -value results of the Wilcoxon rank sum test to explicitly demonstrate the statistical disparities between SCPSO and the other algorithms. The findings show that WOA, HHO, and COA significantly outperform SCPSO on F6. On F10, although the average values of TLCO, SCA, SCSO, and GJO are superior to those of SCPSO, the p -value reveals little variation between them and SCPSO. However, SCPSO excels on F1, F3, F4, F5, F7, F8, F9, F11, and F12. Therefore, SCPSO can solve the CEC2022 problem efficiently.

Table 6: Wilcoxon rank sum test p -value for other algorithms (CEC2022 10 dimension)

F	Algorithm								
	RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO
F1	5.061E-08	2.194E-02							
F2	5.296E-08	3.014E-02	3.157E-07	6.368E-06	5.216E-01	2.355E-01	3.597E-05	5.296E-08	2.938E-02
F3	3.187E-09	3.187E-09	3.187E-09	3.187E-09	3.187E-09	3.187E-09	7.948E-07	3.187E-09	3.187E-09
F4	3.187E-09	1.803E-06	3.187E-09	7.898E-08	4.680E-05	8.597E-06	3.293E-05	3.187E-09	1.597E-05
F5	6.682E-08	6.682E-08	6.682E-08	6.682E-08	7.767E-08	6.682E-08	7.767E-08	6.682E-08	3.459E-03
F6	3.187E-09	2.139E-03	3.187E-09	3.187E-09	3.152E-02	1.794E-02	2.227E-02	3.187E-09	2.393E-01
F7	3.187E-09	1.065E-07	3.187E-09	3.187E-09	8.292E-05	1.047E-06	3.293E-05	3.187E-09	1.227E-03
F8	3.187E-09	2.218E-07	2.218E-07	1.065E-07	1.376E-06	6.015E-07	1.251E-05	1.065E-07	7.712E-03
F9	3.670E-08	3.670E-08	3.670E-08	3.670E-08	3.272E-05	3.272E-05	3.061E-07	3.670E-08	1.701E-02
F10	3.152E-02	4.570E-01	5.979E-01	2.853E-01	3.104E-01	1.782E-03	1.806E-01	1.806E-01	7.203E-02
F11	1.387E-06	3.038E-05	1.362E-04	1.362E-04	7.757E-05	1.090E-04	3.857E-05	2.190E-07	4.608E-05
F12	6.672E-08	7.756E-08	6.672E-08	6.672E-08	2.652E-06	6.672E-08	2.323E-06	6.663E-08	1.303E-07
+/- = /-	0/0/10	1/1/8	0/1/9	0/1/9	1/2/7	1/1/8	0/1/9	1/1/8	0/1/9

To present the convergence performance of SCPSO on the 10-dimension CEC2022 test suite more clearly and explicitly, [Fig. 9](#) gives a convergence plot of SCPSO with other algorithms. We can draw the following conclusions by observing the plot's trend. First, SCPSO exhibits a speedy rate of descent and reaches much lower nadirs on the F2, F5, F8, F9, F11, and F12 functions. That suggests that SCPSO has exceptional convergence performance in addressing these functions. Secondly, for F3, F7, and F11, although SCPSO's rate of descent is not outstanding, it still obtains a superior solution after a certain level of convergence. That points to the fact that SCPSO may need more iterations on these functions to achieve the best results, but it still achieves satisfying convergence performance. Taken together, the superiority of SCPSO on the 10-dimension CEC2022 test suite is evident from the convergence curves in [Fig. 9](#). It demonstrates fast descent rates and better nadir points in several functions, thus proving its effectiveness in solving these problems.

By comparing the boxplot of SCPSO with the other algorithms in [Fig. 10](#), we can observe that on F1, F2, F3, F5, F7, F9, F11, and F12, SCPSO shows thinner boxes, signifying a more centralized distribution of fitness values for SCPSO. This more concentrated distribution implies that the performance of SCPSO is relatively stable on these functions. Simultaneously, we can also note the relatively small number of outliers in the SCPSO boxplot. Measurement errors, special conditions, or other factors may cause outliers. Thus, fewer outliers reflect a more stable and robust performance of SCPSO on these functions.

To investigate the dimension scaling capability of SCPSO further, SCPSO is measured on the 20dimension CEC2022 test suite. [Table 7](#) contains a collection of statistical. Encouragingly, compared to its performance on 10 dimensions, SCPSO is ranked first on 83.3% of the functions and is only ranked 4th and 2nd on F10 and F11, respectively. Furthermore, SCPSO is ranked first on eight functions under average

and standard deviation. That proves the excellence of SCPSO in terms of convergence and emphasizes its stability advantage. That implies that SCPSO can converge efficiently on high-dimensional problems and have excellent stability in outcomes. Depending on the statistical results, the rank of all the algorithms is SCPSO > HHO > SCSO = GJO > PSO > SCA > TLCO > WOA > RSA > COA. SCPSO holds the leading position with an average rank of 1.33. SCSO and GJO have the same average rank and are jointly listed in third place. In summary, SCPSO performs more prominently on higher dimensional problem.

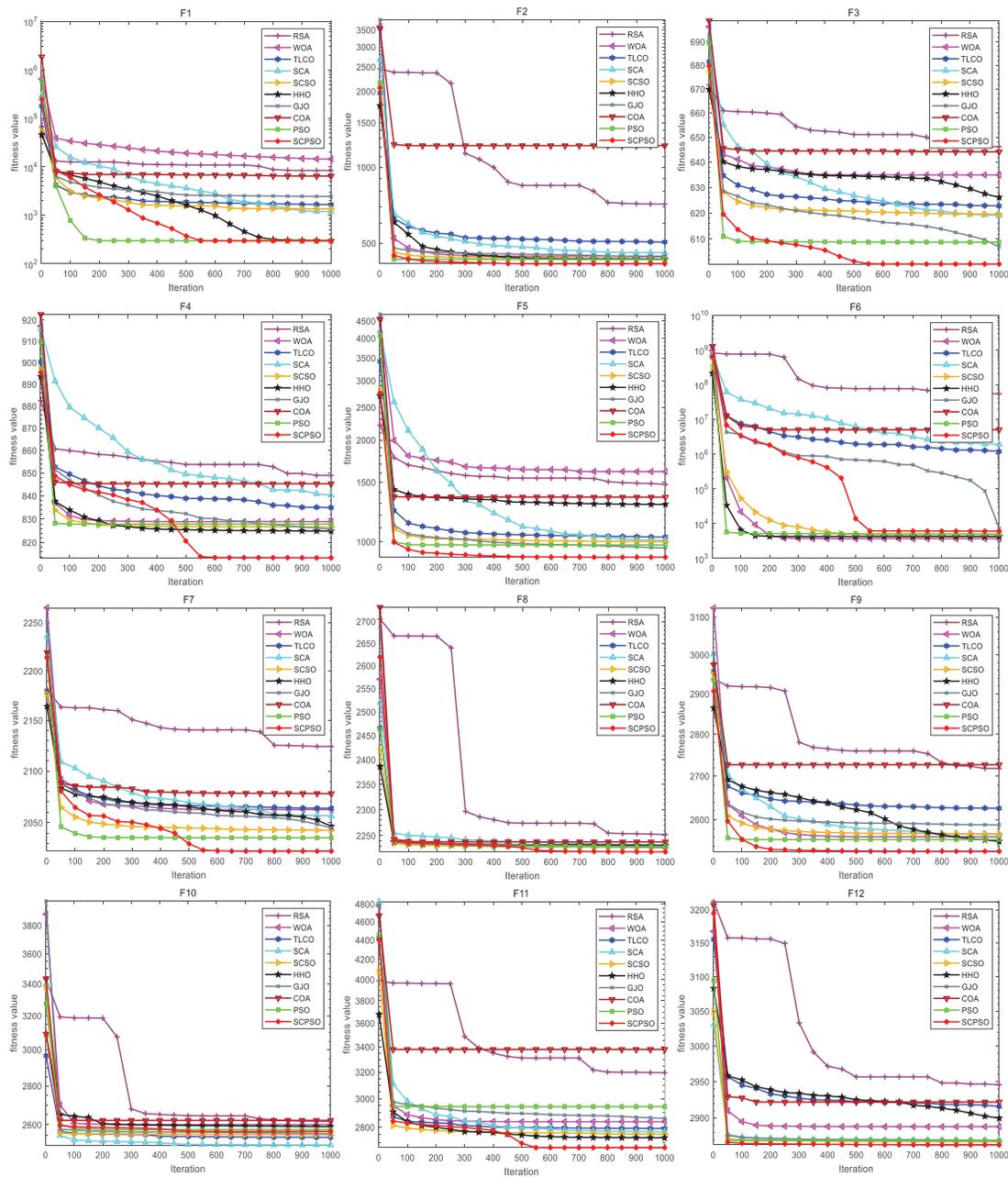


Figure 9: Convergence plot of SCPSO with other algorithms (CEC2022 10 dimension)

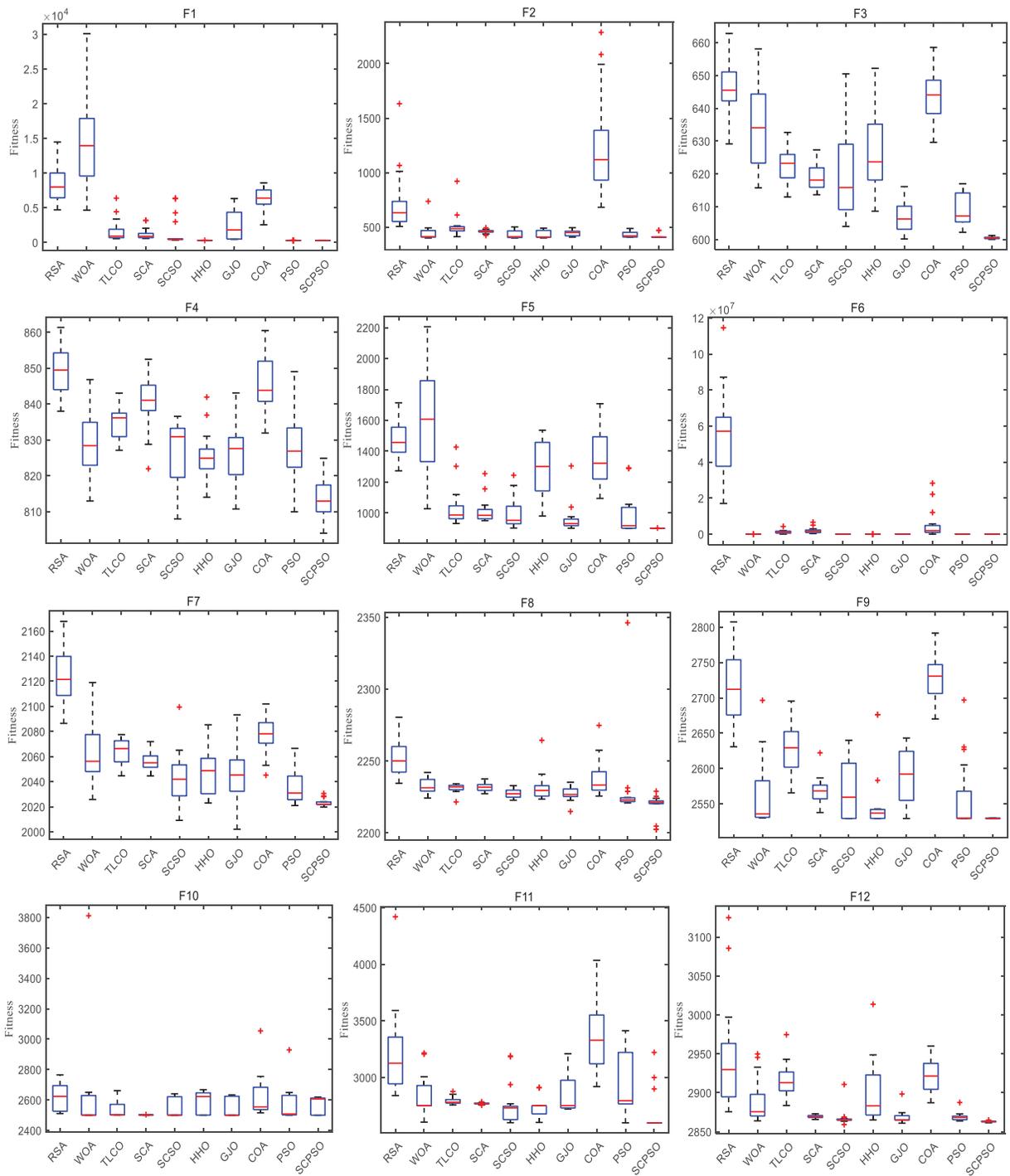


Figure 10: Boxplot of SCPSO with the other algorithms (CEC2022 10 dimension)

Table 7: Comparison of findings of SCPSO with other algorithms (CEC2022 20 dimension)

F	Results	Algorithm									
		RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO	SCPSO
F1	Average	3.149E + 04	1.581E + 04	1.103E + 04	1.182E + 04	1.115E + 04	8.589E + 02	1.311E + 04	4.217E + 04	3.333E + 03	7.026E + 02
	Std	5.611E + 03	5.060E + 03	3.628E + 03	3.074E + 03	6.905E + 03	4.709E + 02	5.945E + 03	9.877E + 03	2.519E + 03	5.301E + 02
	Best	2.130E + 04	8.000E + 03	6.537E + 03	5.928E + 03	1.630E + 03	3.760E + 02	5.054E + 03	2.761E + 04	3.131E + 02	3.043E + 02
	Worst	4.058E + 04	2.897E + 04	1.932E + 04	1.699E + 04	3.160E + 04	1.952E + 03	3.274E + 04	6.213E + 04	8.520E + 03	2.425E + 03
	Rank	9	8	4	6	5	2	7	10	3	1
F2	Average	2.152E + 03	5.564E + 02	8.115E + 02	7.291E + 02	5.427E + 02	4.805E + 02	5.819E + 02	2.340E + 03	5.846E + 02	4.548E + 02
	Std	6.992E + 02	5.103E + 01	1.128E + 02	8.299E + 01	5.677E + 01	3.820E + 01	6.876E + 01	6.290E + 02	1.473E + 02	2.005E + 01
	Best	1.329E + 03	4.850E + 02	6.275E + 02	6.225E + 02	4.733E + 02	4.494E + 02	4.876E + 02	1.459E + 03	4.067E + 02	4.065E + 02
	Worst	3.830E + 03	6.482E + 02	1.016E + 03	8.845E + 02	6.480E + 02	5.936E + 02	7.733E + 02	3.388E + 03	8.149E + 02	4.849E + 02
	Rank	9	4	8	7	3	2	5	10	6	1
F3	Average	6.757E + 02	6.599E + 02	6.562E + 02	6.424E + 02	6.405E + 02	6.537E + 02	6.198E + 02	6.769E + 02	6.314E + 02	6.057E + 02
	Std	8.592E + 00	9.835E + 00	5.802E + 00	4.264E + 00	1.315E + 01	7.879E + 00	7.381E + 00	6.603E + 00	1.112E + 01	2.424E + 00
	Best	6.631E + 02	6.418E + 02	6.455E + 02	6.354E + 02	6.169E + 02	6.412E + 02	6.098E + 02	6.663E + 02	6.133E + 02	6.021E + 02
	Worst	6.961E + 02	6.775E + 02	6.654E + 02	6.501E + 02	6.631E + 02	6.656E + 02	6.325E + 02	6.904E + 02	6.577E + 02	6.110E + 02
	Rank	9	8	7	5	4	6	2	10	3	1
F4	Average	9.719E + 02	9.233E + 02	9.366E + 02	9.437E + 02	8.877E + 02	8.837E + 02	8.946E + 02	9.640E + 02	8.812E + 02	8.430E + 02
	Std	9.323E + 00	3.348E + 01	1.188E + 01	1.176E + 01	1.688E + 01	1.441E + 01	3.004E + 01	1.730E + 01	2.414E + 01	1.104E + 01
	Best	9.563E + 02	8.817E + 02	9.081E + 02	9.170E + 02	8.548E + 02	8.477E + 02	8.342E + 02	9.104E + 02	8.448E + 02	8.249E + 02
	Worst	9.881E + 02	1.001E + 03	9.540E + 02	9.629E + 02	9.184E + 02	9.132E + 02	9.404E + 02	9.922E + 02	9.397E + 02	8.627E + 02
	Rank	10	6	7	8	4	3	5	9	2	1
F5	Average	3.258E + 03	3.654E + 03	2.306E + 03	2.444E + 03	2.159E + 03	2.517E + 03	1.822E + 03	3.375E + 03	2.041E + 03	9.630E + 02
	Std	2.712E + 02	1.141E + 03	4.180E + 02	5.357E + 02	4.311E + 02	2.350E + 02	3.401E + 02	4.127E + 02	7.363E + 02	4.725E + 01
	Best	2.789E + 03	2.475E + 03	1.585E + 03	1.702E + 03	1.405E + 03	2.081E + 03	1.241E + 03	2.459E + 03	9.465E + 02	9.136E + 02
	Worst	3.758E + 03	7.178E + 03	3.085E + 03	3.645E + 03	2.760E + 03	2.856E + 03	2.335E + 03	4.197E + 03	3.976E + 03	1.115E + 03
	Rank	8	10	5	6	4	7	2	9	3	1
F6	Average	8.358E + 08	1.057E + 05	6.736E + 07	1.242E + 08	3.612E + 06	9.357E + 04	6.215E + 06	1.729E + 09	1.774E + 06	3.507E + 03
	Std	1.005E + 09	1.232E + 05	5.246E + 07	8.722E + 07	1.415E + 07	5.226E + 04	1.325E + 07	9.579E + 08	6.518E + 06	2.780E + 03
	Best	1.999E + 08	7.604E + 03	2.839E + 07	1.945E + 07	2.117E + 03	1.962E + 04	9.628E + 03	1.003E + 08	2.029E + 03	1.930E + 03
	Worst	4.941E + 09	4.881E + 05	1.928E + 08	3.536E + 08	6.348E + 07	2.173E + 05	5.760E + 07	3.505E + 09	2.935E + 07	1.456E + 04
	Rank	9	3	7	8	5	2	6	10	4	1
F7	Average	2.231E + 03	2.205E + 03	2.149E + 03	2.137E + 03	2.128E + 03	2.145E + 03	2.108E + 03	2.198E + 03	2.142E + 03	2.057E + 03
	Std	2.576E + 01	7.605E + 01	2.403E + 01	2.022E + 01	3.933E + 01	4.757E + 01	3.448E + 01	2.863E + 01	6.381E + 01	1.913E + 01
	Best	2.183E + 03	2.101E + 03	2.099E + 03	2.102E + 03	2.061E + 03	2.087E + 03	2.052E + 03	2.163E + 03	2.054E + 03	2.027E + 03
	Worst	2.300E + 03	2.334E + 03	2.206E + 03	2.167E + 03	2.198E + 03	2.288E + 03	2.163E + 03	2.245E + 03	2.283E + 03	2.097E + 03
	Rank	10	9	7	4	3	6	2	8	5	1

(Continued)

Table 7 (continued)

F	Results	Algorithm									
		RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO	SCPSO
F8	Average	2.409E + 03	2.274E + 03	2.258E + 03	2.258E + 03	2.280E + 03	2.276E + 03	2.249E + 03	2.414E + 03	2.298E + 03	2.229E + 03
	Std	1.500E + 02	6.447E + 01	1.579E + 01	9.776E + 00	5.837E + 01	7.387E + 01	4.338E + 01	1.202E + 02	6.108E + 01	4.231E + 00
	Best	2.281E + 03	2.230E + 03	2.241E + 03	2.244E + 03	2.229E + 03	2.230E + 03	2.225E + 03	2.243E + 03	2.224E + 03	2.222E + 03
	Worst	2.876E + 03	2.230E + 03	2.299E + 03	2.276E + 03	2.373E + 03	2.458E + 03	2.351E + 03	2.620E + 03	2.416E + 03	2.238E + 03
	Rank	9	5	4	3	7	6	2	10	8	1
F9	Average	3.025E + 03	2.541E + 03	2.680E + 03	2.583E + 03	2.546E + 03	2.493E + 03	2.552E + 03	3.282E + 03	2.543E + 03	2.485E + 03
	Std	1.997E + 02	3.855E + 01	7.164E + 01	2.730E + 01	4.182E + 01	1.144E + 01	3.762E + 01	2.682E + 02	7.752E + 01	1.082E + 01
	Best	2.808E + 03	2.503E + 03	2.623E + 03	2.529E + 03	2.494E + 03	2.484E + 03	2.486E + 03	2.818E + 03	2.481E + 03	2.481E + 03
	Worst	3.593E + 03	2.648E + 03	2.956E + 03	2.650E + 03	2.626E + 03	2.533E + 03	2.645E + 03	3.827E + 03	2.762E + 03	2.510E + 03
	Rank	9	3	8	7	5	2	6	10	4	1
F10	Average	5.070E + 03	4.644E + 03	2.984E + 03	2.541E + 03	3.069E + 03	3.609E + 03	3.807E + 03	6.052E + 03	4.230E + 03	3.452E + 03
	Std	1.608E + 03	1.210E + 03	1.016E + 03	5.574E + 01	1.150E + 03	7.573E + 02	1.549E + 03	1.384E + 03	9.914E + 02	9.402E + 02
	Best	2.627E + 03	2.501E + 03	2.528E + 03	2.519E + 03	2.501E + 03	2.501E + 03	2.502E + 03	2.925E + 03	2.502E + 03	2.501E + 03
	Worst	6.666E + 03	6.090E + 03	5.934E + 03	2.776E + 03	5.750E + 03	4.874E + 03	6.524E + 03	7.087E + 03	5.742E + 03	4.825E + 03
	Rank	9	7	2	1	3	5	6	10	8	4
F11	Average	8.415E + 03	3.233E + 03	5.548E + 03	4.777E + 03	3.917E + 03	3.147E + 03	4.176E + 03	8.200E + 03	4.305E + 03	3.154E + 03
	Std	5.798E + 02	2.000E + 02	4.988E + 02	4.134E + 02	7.809E + 02	1.033E + 03	5.721E + 02	7.637E + 02	1.084E + 03	2.133E + 02
	Best	6.983E + 03	2.747E + 03	4.712E + 03	3.849E + 03	3.170E + 03	2.645E + 03	3.275E + 03	6.123E + 03	3.284E + 03	2.900E + 03
	Worst	9.141E + 03	3.542E + 03	6.823E + 03	5.422E + 03	6.275E + 03	7.483E + 03	5.441E + 03	9.466E + 03	6.309E + 03	3.370E + 03
	Rank	10	3	8	7	4	1	6	9	5	2
F12	Average	3.162E + 03	3.084E + 03	3.408E + 03	3.050E + 03	3.022E + 03	3.083E + 03	3.010E + 03	3.485E + 03	3.012E + 03	2.968E + 03
	Std	1.489E + 02	1.251E + 02	1.537E + 02	2.203E + 01	4.359E + 01	1.457E + 02	4.021E + 01	1.900E + 02	4.082E + 01	2.817E + 01
	Best	3.015E + 03	2.984E + 03	3.182E + 03	3.005E + 03	2.972E + 03	2.966E + 03	2.960E + 03	3.161E + 03	2.950E + 03	2.943E + 03
	Worst	3.619E + 03	3.452E + 03	3.889E + 03	3.086E + 03	3.140E + 03	3.563E + 03	3.111E + 03	3.896E + 03	3.082E + 03	3.046E + 03
	Rank	8	6	9	5	4	7	2	10	3	1
Average rank		9.08	6.00	6.33	5.58	4.25	4.08	4.25	9.58	4.5	1.33
Result		8	7	6	5	3	2	3	9	4	1

Table 8 presents the p -value results from the Wilcoxon rank sum test applied to the 20-dimensional CEC2022 test suite. A closer look at the table reveals that while TLCO, SCA, and SCSO show a slight average advantage over SCPSO on F10, the p -value differences are minimal. That suggests these algorithms may marginally outperform SCPSO on specific functions, but the disparities are not statistically significant. In contrast, SCPSO consistently outshines most other algorithms, securing top positions across most test functions. It leads in terms of performance and claims the top spot in both the mean rank and overall ranking, reinforcing its dominance in optimization tasks.

Table 8: Wilcoxon rank sum test p -values for other algorithms (CEC2022 20 dimension)

F	Algorithm								
	RSA	WOA	TLCO	SCA	SCSO	HHO	GJO	COA	PSO
F1	3.187E - 09	3.187E - 09	3.187E - 09	3.187E - 09	7.898E - 08	8.586E - 02	3.187E - 09	7.577E - 06	2.596E - 05
F2	3.187E - 09	3.187E - 09	3.187E - 09	3.187E - 09	9.173E - 08	3.057E - 03	3.187E - 09	3.187E - 09	1.794E - 02
F3	3.187E - 09	1.065E - 07	3.187E - 09	3.187E - 09					
F4	3.187E - 09	3.187E - 09	3.187E - 09	3.187E - 09	9.173E - 08	2.960E - 07	2.356E - 06	3.187E - 09	1.376E - 06
F5	3.187E - 09	3.939E - 07							
F6	3.187E - 09	9.173E - 08	3.187E - 09	3.187E - 09	9.278E - 05	3.187E - 09	7.898E - 08	7.898E - 08	7.406E - 05
F7	3.187E - 09	3.187E - 09	3.187E - 09	3.187E - 09	6.015E - 07	9.173E - 08	9.748E - 06	6.796E - 08	2.690E - 06
F8	9.748E - 06	2.341E - 03	1.227E - 03	1.116E - 03	6.040E - 03	2.564E - 02	5.428E - 01	3.987E - 06	5.561E - 03
F9	6.738E - 08	3.909E - 07	6.738E - 08	6.738E - 08	3.909E - 07	1.154E - 04	1.644E - 07	6.738E - 08	5.451E - 05
F10	9.209E - 04	2.222E - 04	7.972E - 01	2.977E - 01	8.817E - 01	4.408E - 01	1.333E - 01	6.674E - 06	6.040E - 03
F11	3.187E - 09	4.903E - 01	3.187E - 09	3.187E - 09	1.574E - 06	1.198E - 01	2.059E - 06	3.187E - 09	1.992E - 04
F12	1.235E - 07	2.062E - 06	3.187E - 09	3.416E - 07	2.041E - 05	1.415E - 05	1.610E - 04	6.796E - 08	6.220E - 04
+ / = / -	0/0/10	0/1/9	0/1/9	0/1/9	0/1/9	0/3/7	1/2/7	0/0/10	0/0/10

Figs. 11 and 12 contain convergence plots and boxplots of SCPSO and other algorithms on the 20-dimension CEC2022 test function. The convergence curve of SCPSO on the 20-dimension function decreases faster compared to that of the convergence plot in the 10-dimension case. Especially on F3, F4, F5, F6, and F7, SCPSO displays more notable advantages than other algorithms. Likewise, SCPSO has a relatively narrow box plot, which suggests it is relatively more stable and robust. A narrower box plot indicates that the algorithm has more consistent convergence performance across running instances and is more stable concerning parameter tuning and initial settings. Therefore, SCPSO performs better in solving these 20-dimensional function problems.

A radar plot according to the rank of SCPSO and other superior algorithms in the 10 and 20 dimensions of the CEC2022 test suite is plotted in Figs. 13 and 14. Observing the two sets of plots, SCPSO has the smallest area, and its image is more rounded in the 20 dimensions, demonstrating that SCPSO is more convergent and stable in the 20 dimensions of the test function. In addition, Fig. 15 visualizes the cumulative average ranks for the two dimensions. It again reveals that SCPSO provides an excellent output on both dimensions and the greater the dimension, the more enhanced the performance.

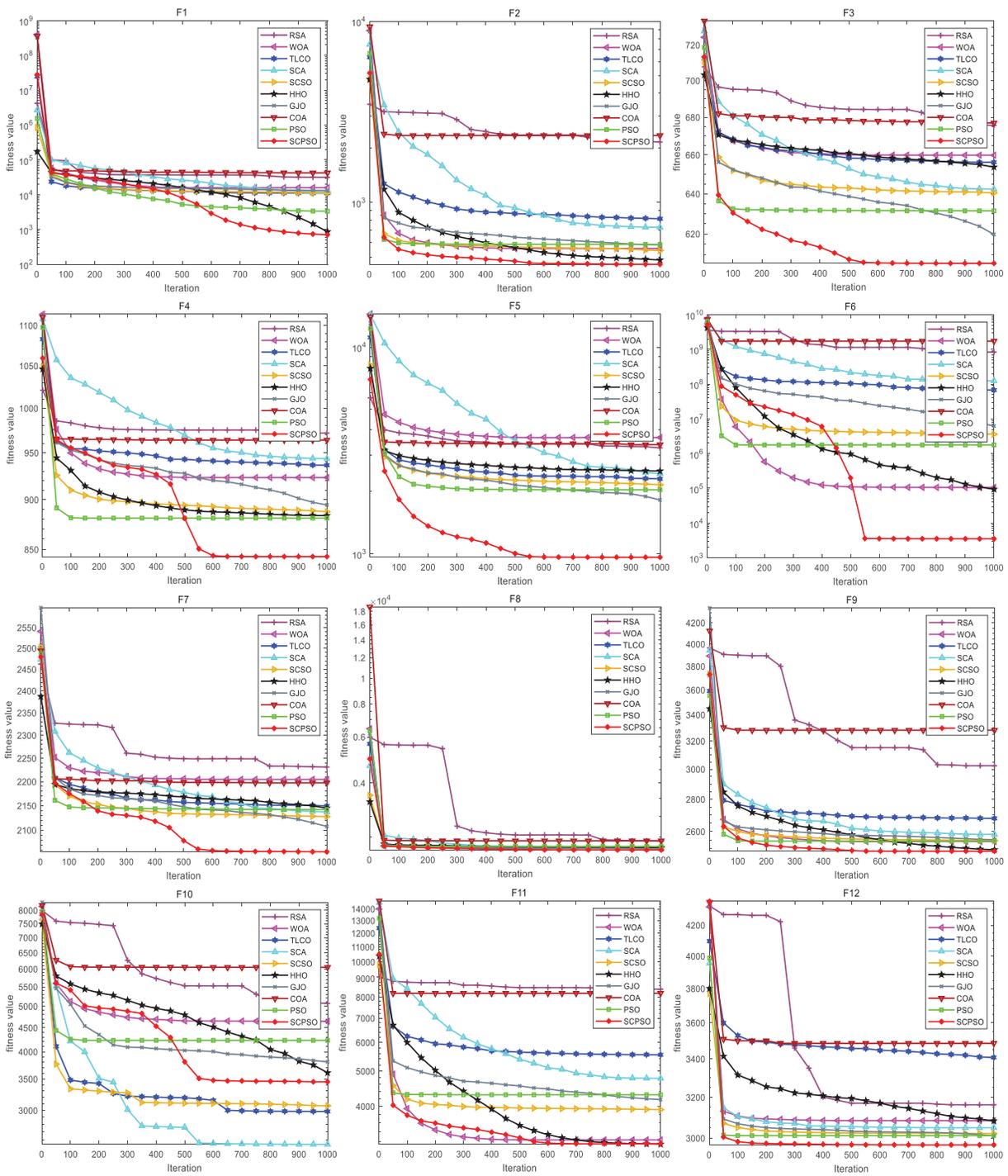


Figure 11: Convergence plot of SCPSO with other algorithms (CEC2022 20 dimension)

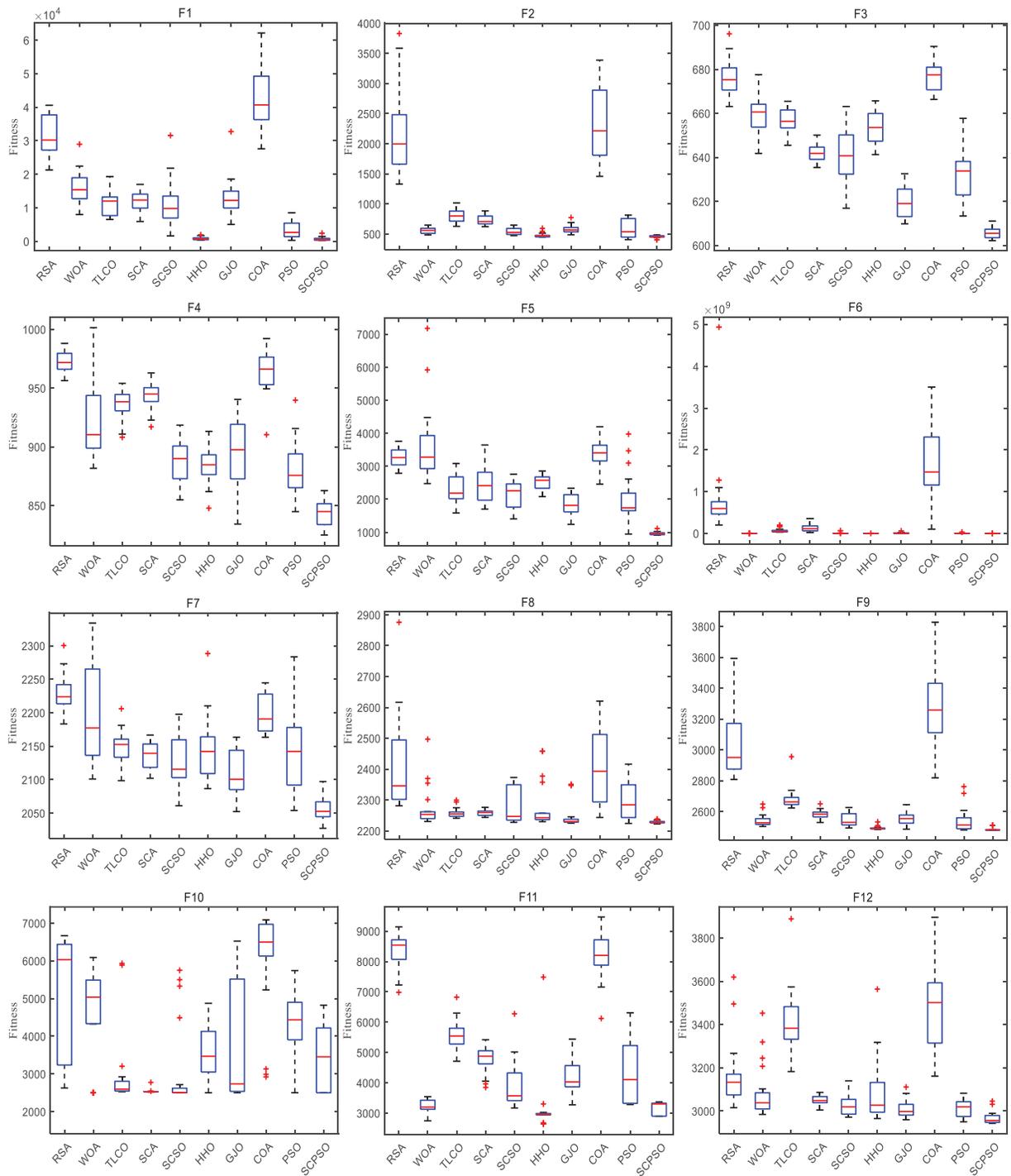


Figure 12: Boxplot of SCPSO with the other algorithms (CEC2022 20 dimension)

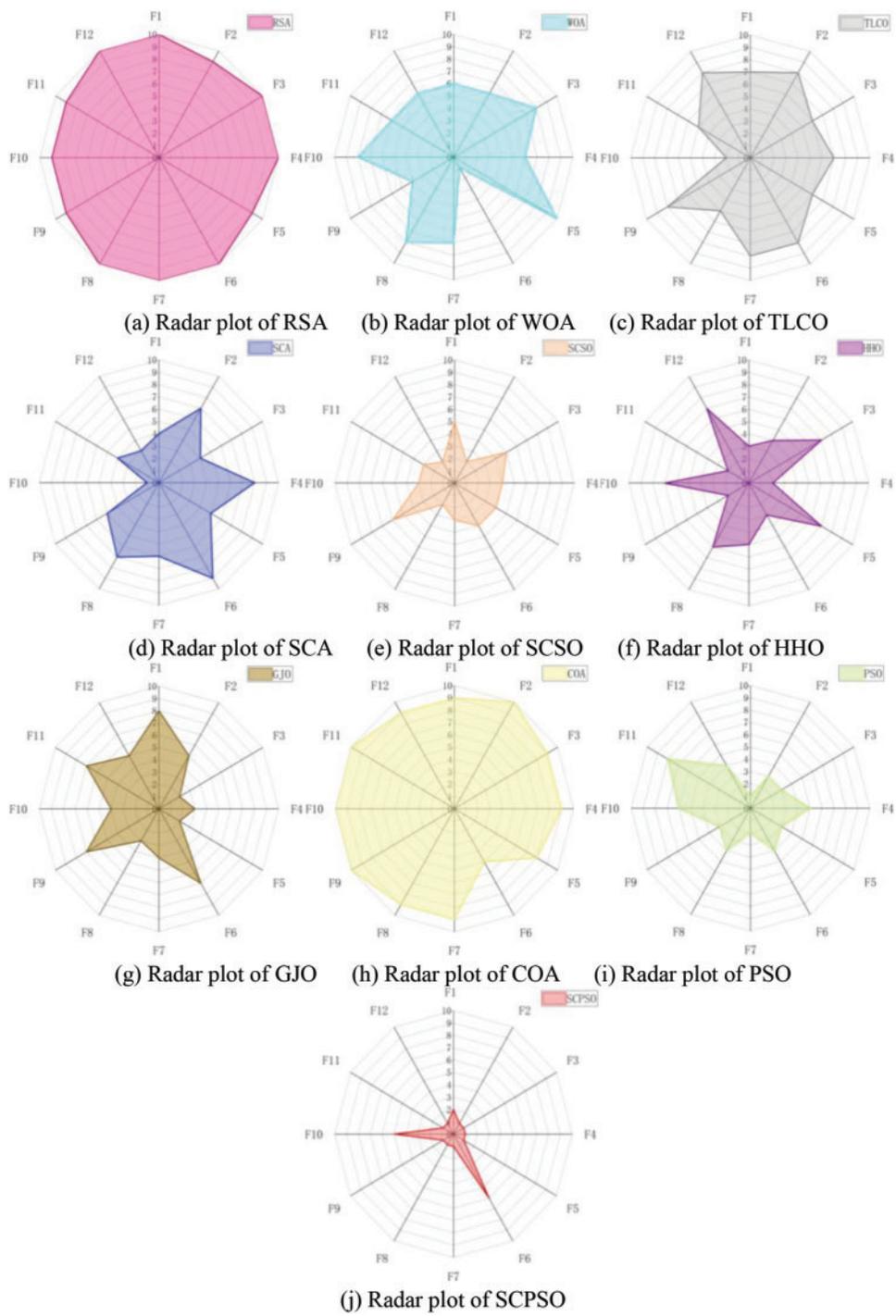


Figure 13: Radar plot of SCPSO with other algorithms (CEC2022 10 dimension)

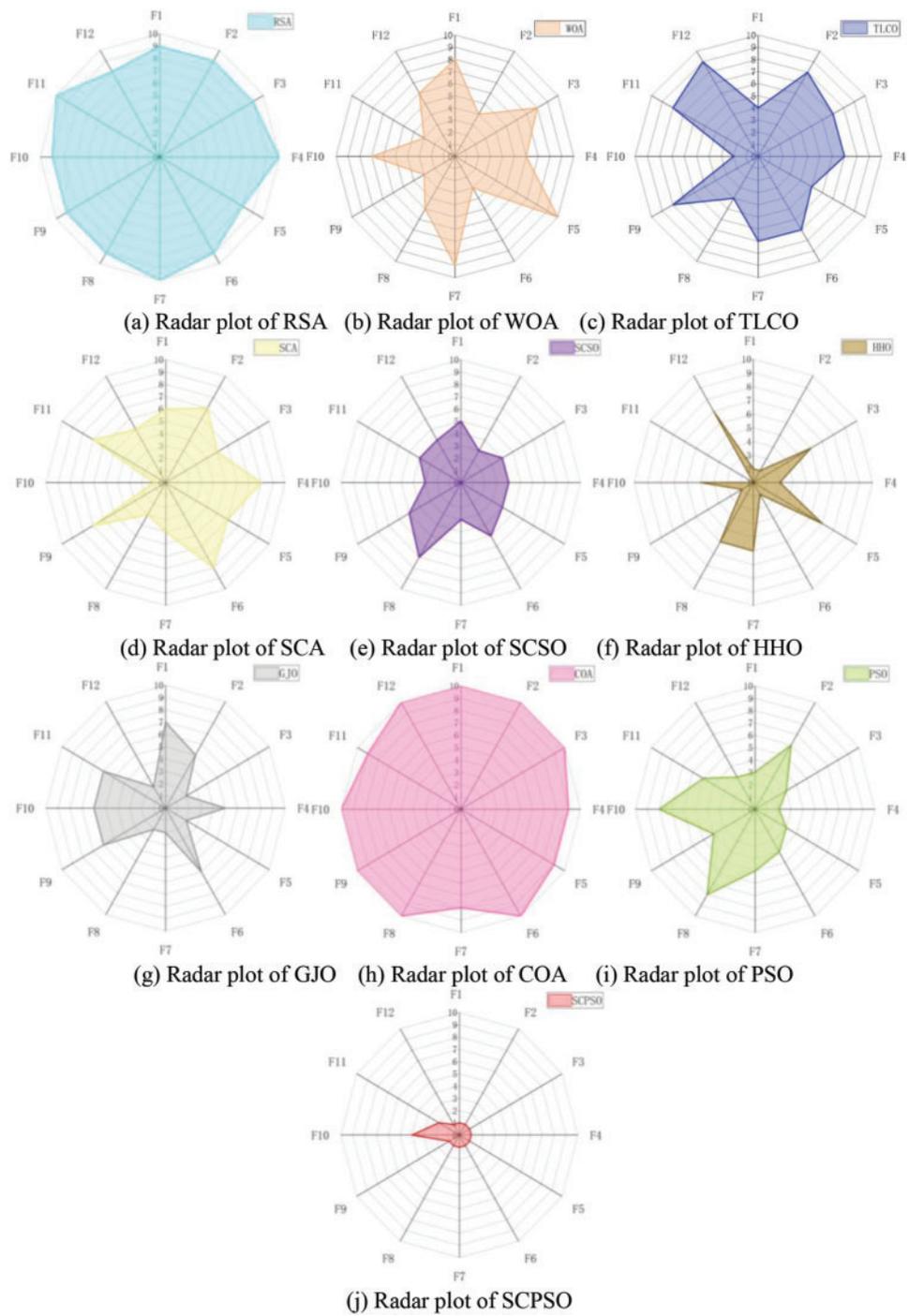


Figure 14: Radar plot of SCPSO with the other algorithms (CEC2022 20 dimension)

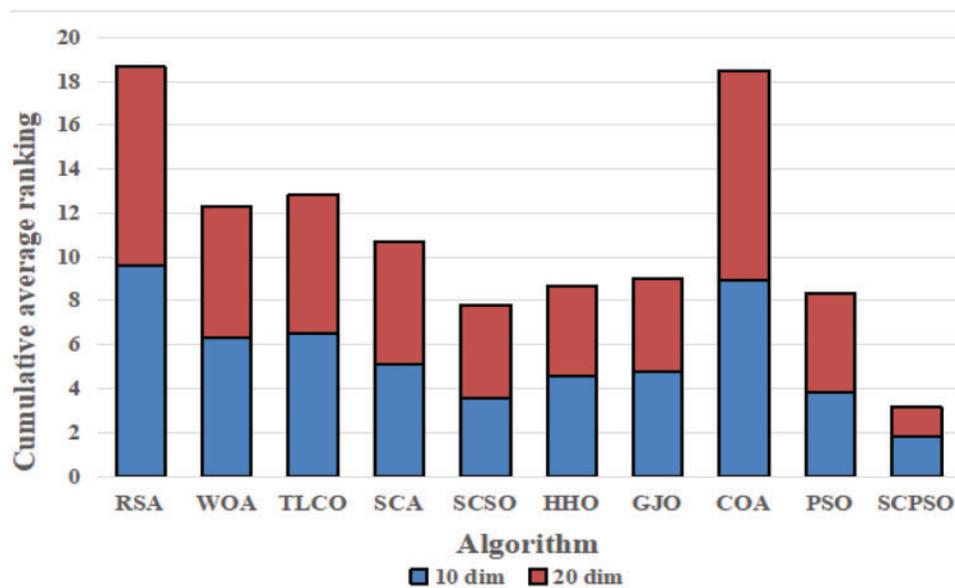


Figure 15: Cumulative average rank of the 12 algorithms on two different dimensions

6 Engineering Issues

6.1 Alkylation Unit Optimization Issue

To further assess the capability of SCPSO in addressing intricate constrained optimization issues, it is applied to three challenging problems. These problems are the optimization of the alkylation unit, the vehicle side impact design problem, and the industrial refrigeration system design problem. To deal with these nonlinearly constrained problems, we introduce the concept of penalty function [51]. By transforming the constraints into penalty terms of the objective function, we can optimize unconstrained problems in SCPSO. In this approach, the algorithm evaluates the solution in each iteration and adjusts to the value of the penalty function to approximate the optimal solution. In this case, the population size is 30, the maximum count of iterations is 500, and the count of executions is 30.

The alkylation process [52] is an essential reaction common in the petroleum industry, and it finds extensive utility in the synthesis of alkylated products across various domains. In this process, an olefin feed (usually pure butene) is introduced into a reactor for reaction, a pure isobutane recycles, a 100% isobutene supplemental stream, and an acid catalyst. The product stream in the reactor is then processed through a fractionator to separate the isobutene and alkylation products, while the exhausted acid is discharged from the reactor. The design of the alkylation unit is diagrammed in Fig. 16. In this problem, there are 14 inequality constraints, and 7 variables $Pa = [pa_1, pa_2, pa_3, pa_4, pa_5, pa_6, pa_7]$ are involved. The profit maximization of the reaction system is achieved by considering various constraints comprehensively. The detailed model is as described below:

$$\min f(Pa) = 1.715pa_1 + 0.035pa_1pa_6 + 4.0565pa_3 + 10pa_2 - pa_3pa_5 - 3000 \quad (20)$$

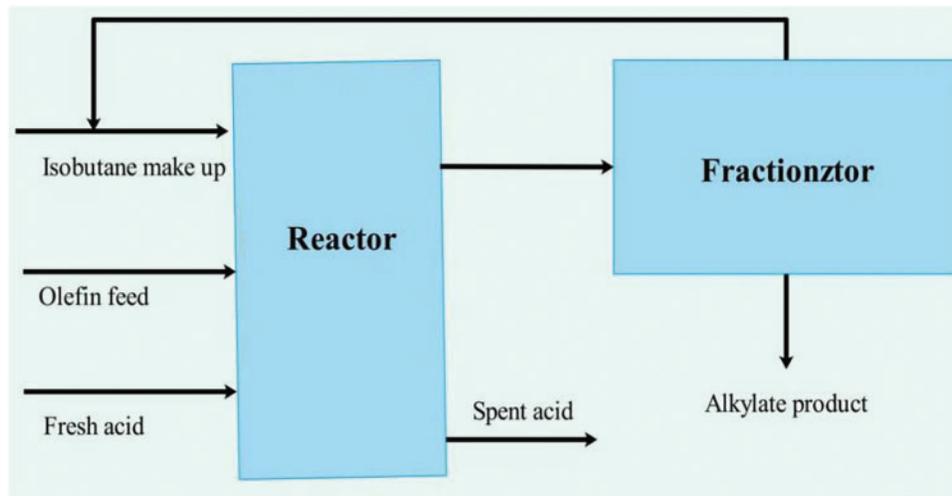


Figure 16: Design chart of the alkylation unit

Constraints:

$$g_1(Pa) = 0.0059553571pa_6^2 + 0.88392857\frac{pa_3}{pa_1} - 0.1175625pa_6 - 1,$$

$$g_2(Pa) = 1.1088\frac{pa_1}{pa_3} + 0.1303533\frac{pa_1pa_6}{pa_3} - 0.0066033\frac{pa_1pa_6^2}{pa_3} - 1,$$

$$g_3(Pa) = 0.00066173269pa_6^2 + 0.017239878pa_5 - 0.0056595559pa_4 - 0.19120592pa_6 - 1,$$

$$g_4(Pa) = \frac{56.85075}{pa_5} + \frac{1.08702pa_6}{pa_5} + \frac{0.32175pa_4}{pa_5} - \frac{0.32175pa_6^2}{pa_5} - 1,$$

$$g_5(Pa) = 0.006198pa_7 + \frac{2462.3121pa_2}{pa_3pa_4} - \frac{25.125634pa_2}{pa_3} - 1,$$

$$g_6(Pa) = \frac{0.0016118996}{pa_7} + \frac{5000pa_2}{pa_3pa_7} - \frac{489510pa_2}{pa_3pa_4pa_7} - 1,$$

$$g_7(Pa) = \frac{44.333333}{pa_5} + \frac{0.33pa_7}{pa_5} - 1,$$

$$g_8(Pa) = 0.022556pa_5 - 0.007595pa_7 - 1,$$

$$g_9(Pa) = 0.00061pa_3 - 0.0005pa_1 - 1,$$

$$g_{10}(Pa) = \frac{0.819672pa_1}{pa_3} + \frac{0.819672}{pa_3} - 1,$$

$$g_{11}(Pa) = \frac{24500pa_2}{pa_3pa_4} - \frac{250pa_2}{pa_3} - 1,$$

$$g_{12}(Pa) = 0.010204082pa_4 + \frac{0.000012244898pa_3pa_4}{pa_2} - 1,$$

$$g_{13}(Pa) = 0.0000625pa_1pa_6 + 0.0000625pa_1 - 0.00007625pa_3 - 1,$$

$$g_{14}(Pa) = \frac{1.22pa_3}{pa_1} - pa_6 + pa_1 - 1$$

Variables take values in the range:

$$1 \leq pa_1 \leq 2000, 1 \leq pa_2 \leq 120, 1 \leq pa_3 \leq 5000, 85 \leq pa_4 \leq 93,$$

$$90 \leq pa_5 \leq 95, 3 \leq pa_6 \leq 12, 145 \leq pa_7 \leq 162$$

The ten excellent algorithms used in this experiment are the Bird Search Algorithm [53], Aquila optimizer (AO) [54], SCSO, WOA, RSA, TLCO, SCA, Beluga whale optimization (BWO) [55], PSO, SCPSO. The optimal variables and outcomes of each algorithm for settling the alkylation unit problem are demonstrated in Table 9, from which we can see that SCPSO achieves the maximum profit. In addition, observing the statistical outcomes presented in Table 10, we can perceive that SCPSO has a smaller average and a minor standard deviation, suggesting that its results are more stable. That implies that SCPSO can achieve the optimal solution more reliably.

Table 9: Optimal outcomes of the alkylation unit optimization issue

Algorithm	Optimal variables							Optimal outcomes
	pa_1	pa_2	pa_3	pa_4	pa_5	pa_6	pa_7	
BSA	215.8513	1.3775	5000	85	90	3	155.3877	-4468.472691
AO	169.7608	57.8484	4952.1053	91.0867	90	7.4683	149.1759	-3780.575170
SCSO	215.8312	1.3769	5000	85	90	3	155.6344	-4467.477685
WOA	215.8515	1.3774	5000	85	90	3	162	-4468.476520
RSA	272.9829	1.4766	5000	92.9996	90	3.0038	162	-4435.927422
TLCO	213.3441	2.6151	4998.2872	86.7330	90	4.6998	151.4518	-4447.871453
SCA	215.1958	1.3510	5000	85	90	3	145	-4468.460454
BWO	214.3297	1	5000	85	90	3	145	-4466.002916
PSO	215.8519	1.3774	5000	85	90	3	155.5748	-4467.477612
SCPSO	215.8412	1.3787	5000	85	90	3	155.0522	-4468.477663

Table 10: Statistical outcomes of optimization problems for alkylation unit

Algorithm	Best	Worst	Average	Std
BSA	-4468.472691	-3288.898966	-4423.826364	214.9057773
AO	-3780.575170	-1493.944319	-3039.009522	554.3926668
SCSO	-4467.477685	-2649.450355	-4125.345270	621.9668770
WOA	-4468.476520	-4466.890332	-4468.051777	0.7123301
RSA	-4435.927422	-2045.949163	-3530.728638	569.0514241
TLCO	-4447.871453	-4362.136759	-4397.728715	23.4395622
SCA	-4468.460454	-4467.356721	-4468.071897	0.3186941
BWO	-4466.002916	-4367.374081	-4442.271801	28.0633036
PSO	-4467.477612	-4466.033154	-4467.012516	1.2164211
SCPSO	-4468.477663	-4468.476833	-4468.477411	0.0002109

6.2 Automobile Lateral Crash Design Issue

The automobile lateral crash issue [56] is a constrained optimization problem. It aims to minimize the extent of passenger injuries in a collision by optimizing the crash resistance of the vehicle to meet vehicle market and regulatory side-impact standards and to provide adequate side-impact protection. According to the European Enhanced Vehicle Safety Council (EEVC) European vehicle side impact procedure, this paper performs optimization under eight constraints to minimize the vehicle weight. Among the variables is the thickness of the inside of the B-pillar (pa_1), the B-pillar strengthening member (pa_2), the inside of the floor (pa_3), the cross member (pa_4), the door beam (pa_5), the door seat belt reinforcement (pa_6), the roof truss beam (pa_7), the material of the inside of the B-pillar (pa_8), the inside of the floor (pa_9), the stature of the impediment (pa_{10}), and the batting slotting (pa_{11}). The equation for this issue is:

$$\min f(Pa) = 1.98 + 4.9pa_1 + 6.67pa_2 + 6.98pa_3 + 4.01pa_4 + 1.78pa_5 + 2.73pa_7 \quad (21)$$

Constraints:

$$g_1(Pa) = 1.16 - 0.3717 pa_2 pa_4 - 0.00391 pa_2 pa_{10} - 0.484 pa_3 pa_9 + 0.01343 pa_6 pa_{10} - 1,$$

$$g_2(Pa) = 0.261 - 0.0159 pa_1 pa_2 - 0.188 pa_1 pa_8 - 0.019 pa_2 pa_7 + 0.0144 pa_3 pa_5 + 0.0008757 pa_5 pa_{10} \\ + 0.08045 pa_6 pa_9 + 0.00139 pa_8 pa_{11} + 0.00001575 pa_{10} pa_{11} - 0.32,$$

$$g_3(Pa) = 0.214 + 0.00817 pa_5 - 0.131 pa_1 pa_8 - 0.0704 pa_1 pa_9 + 0.03099 pa_2 pa_6 - 0.018 pa_2 pa_7 \\ + 0.0208 pa_3 pa_8 + 0.121 pa_3 pa_9 - 0.00364 pa_5 pa_6 + 0.0007715 pa_5 pa_{10} - 0.0005354 pa_6 pa_{10} \\ + 0.00121 pa_8 pa_{11} - 0.32.$$

$$g_4(Pa) = 0.74 - 0.061 pa_2 - 0.163 pa_3 pa_8 + 0.001232 pa_3 pa_{10} - 0.166 pa_7 pa_9 + 0.227 pa_2^2 - 0.32,$$

$$g_5(Pa) = 28.98 + 3.818 pa_3 - 4.2 pa_1 pa_2 + 0.02070.001232 pa_5 pa_{10} \\ + 6.63 pa_6 pa_9 - 7.7 a_7 pa_8 + 0.32 pa_9 pa_{10} - 32,$$

$$g_6(Pa) = 33.86 + 2.95 pa_3 + 0.1792 pa_{10} - 5.057 pa_1 pa_2 - 11 pa_2 pa_8 - 0.0215 pa_5 pa_{10} - 9.98 a_7 pa_8 \\ + 22 pa_8 pa_9 - 32,$$

$$g_7(Pa) = 46.36 - 9.9 pa_2 - 12.9 pa_1 pa_8 + 0.1107 pa_3 pa_{10} - 32,$$

$$g_8(Pa) = 4.72 - 0.5 pa_4 - 0.19 pa_2 pa_3 - 0.0122 pa_4 pa_{10} + 0.009325 pa_6 pa_{10} + 0.000191 pa_{11}^2 - 4$$

Variables take values in the range:

$$0.5 \leq pa_1, pa_3, pa_4 \leq 1.5, 0.45 \leq pa_2 \leq 1.35, 0.875 \leq pa_5 \leq 2.625,$$

$$0.4 \leq pa_6, pa_7 \leq 1.2, 0.192 \leq pa_8, pa_9 \leq 0.345, -30 \leq pa_{10}, pa_{11} \leq 30$$

The following participated in this experiment: SCA, AO, Slime Mould Algorithm (SMA) [57], TLCO, Artificial Rabbits Optimization (ARO) [58], WOA, Seagull Optimization Algorithm (SOA) [59], Mountain Gazelle Optimizer (MGO) [60], PSO, and SCPSO. Based on the optimal results of the automobile lateral crash design issue presented in Table 11, SCPSO has performed better in solving this problem. Moreover, other statistical outcomes of each algorithm for solving the automobile lateral crash design issue are also provided in Table 12, which further demonstrates the superiority of SCPSO.

Table 11: Optimal outcomes for the automobile lateral crash design issue

Algorithm	Optimal variables											Optimal outcomes
	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	pa_1	
SCA	0.5	1.0436	0.5	0.5	0.5082	1.5	0.5	0.3450	0.3450	-30	12.1610	22.414455
AO	0.5457	0.8942	1.0158	0.5839	0.7474	0.9305	0.5680	0.3401	0.3298	-28.229	2.1109	25.182156
SMA	0.5	1.0526	0.5	0.5	0.5	1.5	0.5	0.3450	0.3450	-30	0.0073	22.238323
TLCO	0.8740	0.8962	0.6074	0.5839	0.5030	1.3039	0.6153	0.3177	0.2035	-29.064	-0.9041	23.997613
ARO	0.5101	1.0479	0.5	0.5	0.5	1.5	0.5	0.3450	0.3450	-29.998	0.1794	22.238922
WOA	0.5	1.0480	0.5124	0.5	0.5	1.5	0.5	0.3450	0.2657	-30	-2.3475	22.333979
SOA	0.5048	1.0491	0.5	0.5	0.5	1.5	0.5052	0.3450	0.3403	-30	6.7587	22.279301
MGO	0.5	1.0525	0.5	0.5	0.5	1.5	0.5	0.3450	0.3450	-30	0	22.238312
PSO	0.5	1.0525	0.5	0.5	0.5	1.5	0.5	0.3450	0.3450	-30	0	22.238312
SCPSO	0.8777	0.5120	0.0263	0.3399	0.3759	0.4833	0.4112	0.8199	0.0499	0.1373	0.6234	18.071202

Table 12: Statistical outcomes of automobile lateral crash design issue

Algorithm	Best	Worst	Average	Std
SCA	22.41445503	23.98014985	23.09158126	0.38403839
AO	25.18215649	31.34997360	28.23093593	1.32330282
SMA	22.23832343	22.25508511	22.24040018	0.00397588
TLCO	23.99761321	27.12801303	25.16787171	0.61397204
ARO	22.23892245	22.33920453	22.25908192	0.02900766
WOA	22.33397876	25.13155506	23.67883400	0.79935778
SOA	22.27930119	22.74159649	22.43994485	0.10735146
MGO	22.23831193	22.25800664	22.23980907	0.00413040
PSO	22.23831193	23.99172644	22.45021007	0.49275942
SCPSO	18.07120239	23.06415054	21.30212189	1.29661810

6.3 Industrial Refrigeration System Design Issue

Industrial refrigeration system [61] is a type of refrigeration equipment used in extensive industrial facilities and commercial buildings closely related to energy consumption. Therefore, the optimal design of an industrial refrigeration system plays a significant role in regulating energy consumption. This problem aims to find the minimum energy consumption under fifteen constraints and contains fourteen variables, which are formulated as follows:

$$\begin{aligned}
 \min f(Pa) = & 63098.88 pa_2 pa_4 pa_{12} + 5441.5 pa_2^2 pa_{12} + 115055.5 pa_2^{1.664} pa_6 + 6172.27 pa_2^2 pa_6 \\
 & + 63098.88 pa_1 pa_3 pa_{11} + 5441.5 pa_1^2 pa_{11} + 115055.5 pa_1^{1.664} pa_5 + 6172.27 pa_1^2 pa_5 \\
 & + 140.53 pa_1 pa_{11} + 70.26 pa_1^2 + 281.29 pa_1 pa_3 + 281.29 pa_3^2 \\
 & + 14437 pa_8^{1.8812} pa_{12}^{0.3424} pa_{10} pa_{14}^{-1} pa_1^2 pa_7 pa_9^{-1} + 20470.2 pa_7^{2.893} pa_{11}^{0.316} pa_1^2
 \end{aligned} \tag{22}$$

Constraints:

$$\begin{aligned}
 g_1(Pa) &= 1.524 pa_7^{-1} - 1, \quad g_2(Pa) = 1.524 pa_8^{-1} - 1, \\
 g_3(Pa) &= 0.07789 pa_1 - 2 pa_7^{-1} pa_9 - 1, \\
 g_4(Pa) &= 7.05305 pa_9^{-1} pa_1^2 pa_{10} pa_8^{-1} pa_2^{-1} pa_{14}^{-1} - 1,
 \end{aligned}$$

$$\begin{aligned}
 g_5(Pa) &= 0.0833pa_{13}^{-1}pa_{14} - 1, \\
 g_6(Pa) &= 47.136pa_2^{0.333}pa_{10}^{-1}pa_{12} - 1.333pa_8pa_{13}^{2.1195} \\
 &\quad 62.08pa_{13}^{2.1195}pa_{12}^{-1}pa_8^{0.2}pa_{10}^{-1} - 1, \\
 g_7(Pa) &= 0.04771pa_{10}pa_8^{1.8812}pa_{12}^{0.3424} - 1, \\
 g_8(Pa) &= 0.0488pa_9pa_7^{1.893}pa_{11}^{0.316} - 1, \\
 g_9(Pa) &= 0.0099pa_1pa_3^{-1} - 1, g_{10}(Pa) = 0.0193pa_2pa_4^{-1} - 1, \\
 g_{11}(Pa) &= 0.0298pa_1pa_5^{-1} - 1, g_{12}(Pa) = 0.056pa_2pa_6^{-1} - 1, \\
 g_{13}(Pa) &= 2pa_9^{-1} - 1, g_{14}(Pa) = 2pa_{10}^{-1} - 1, \\
 g_{15}(Pa) &= 2pa_{11}^{-1} - 1
 \end{aligned}$$

Variables take values in the range:

$$0.001 \leq pa_i \leq 5, i = 1, 2, \dots, 14$$

PSO and SCPSO that participated in the comparative experiments, as well as the COA, Ant Lion Optimizer (ALO) [62], Harmony Search Algorithm (HS) [63], AO, Arithmetic Optimization Algorithm (AOA) [64], TLCO, BSA, and RSA. The optimal results of the industrial refrigeration system design problem are provided in Table 13, from which it can be noted that the optimal value of SCPSO is closest to the desired result. Moreover, from the statistical outcomes of each algorithm for solving the industrial refrigeration system design issue in Table 14, SCPSO has the smallest mean value and is also in a distant second place regarding standard deviation.

Table 13: Optimal outcomes for the industrial refrigeration system design issue

Algorithm	Optimal variables														Optimal outcomes
	pa_1	pa_2	pa_3	pa_4	pa_5	pa_6	pa_7	pa_8	pa_9	pa_{10}	pa_{11}	pa_{12}	pa_{13}	pa_{14}	
COA	0	0	0	0	0	0	0.50	0.13	0.50	0.09	0.11	0.04	0	0	0.01200097
ALO	0	0	0	0	0	0	0.15	0.39	0.88	0.78	0.07	0	0.01	0.24	0.02649406
HS	0.01	0.01	0	0.02	0.14	0.02	0.61	2.65	3.01	0.78	0.69	0.67	0.95	4.14	15.30238649
AO	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	5.32695568
AOA	0	0	0	0	0	0	0.10	0.05	5.00	0.65	0.05	0	0	0.37	0.03972975
TLCO	0	0	0	0.13	0.44	0.11	3.59	4.01	0.83	1.28	2.60	0.13	0.27	5.00	8.39399418
BSA	0	0	0	0	0.04	0	0.04	0.17	0.38	0.08	0.08	0.01	0.02	0.06	0.09101996
RSA	0	0	0	0	0	0	0.15	1.25	3.93	1.69	0.07	0.94	0.74	2.45	0.09598978
PSO	0	0	0	0	0	0	0.25	0.41	5.00	5.00	0.07	0	0.02	0.62	0.02592395
SCPSO	0	0	0	0	0	0	0.26	0.45	5.00	1.18	0.07	0	0.01	0.23	0.02586357

Table 14: Statistical outcomes of industrial refrigeration system design issue

Algorithm	Best	Worst	Average	Std
COA	0.01200097	4.78285405	1.46993420	1.25630923
ALO	0.02649406	2.97715709	0.24298702	0.56443219
HS	15.30238649	5095.03060	491.72975	943.29833
AO	5.32695568	6.53870665	6.46798239	0.23926326
AOA	0.03972975	0.17823339	0.08453720	0.04020038

(Continued)

Table 14 (continued)

Algorithm	Best	Worst	Average	Std
TLCO	8.39399418	426.44383609	139.52977397	111.50725407
BSA	0.09101996	6.29963487	2.15737234	1.82540728
RSA	0.09598978	663.58908278	443.87324855	316.06062290
PSO	0.02592395	29.11048468	2.30408251	5.59493673
SCPSO	0.02586357	0.02657502	0.02593954	0.00013654

7 Feature Selection

Feature selection (FS) [65] is a critical component of machine learning and data analysis, aiming to identify the most relevant subset of features from the raw data to build accurate and efficient predictive models. This process not only enhances model performance but also simplifies interpretation and reduces computational costs, making it essential for high-dimensional data applications. In this study, we leverage the Self-adaptive Cooperative Particle Swarm Optimization (SCPSO) algorithm to tackle the multi-objective FS problem effectively.

SCPSO is applied to achieve two main goals simultaneously: maximizing classification accuracy and minimizing the number of selected features. By balancing these objectives, SCPSO intelligently narrows down the feature space, identifying feature subsets that both maintain high predictive power and enhance efficiency. Through this approach, SCPSO demonstrates its ability to outperform other algorithms by selecting the most informative features that contribute to precise, generalizable models. This dual objective underscores the robustness of SCPSO in adapting to complex datasets, highlighting its value as a tool for both performance-driven and efficiency-oriented feature selection in machine learning tasks.

7.1 Modeling

Feature selection (FS) exemplifies a classic binary optimization challenge. In this context, each feature is represented by binary variables: a value of 1 indicates the feature is selected, while 0 signifies it is excluded. This binary transformation is fundamental to refining the feature subset and enhances interpretability in the optimization process. To robustly assess the performance of the binary FS algorithm, we employ 5-fold cross-validation, which partitions the data into training and test sets in five cycles, ensuring that the model's reliability is consistently validated across varying data segments. This setup not only reinforces the robustness of the model but also minimizes overfitting. The core objective of FS is to strike a fine balance: minimizing the number of selected features while preserving, or even enhancing, classification accuracy. The fitness function is thus meticulously designed to reward minimal feature sets that maintain high predictive power. This approach emphasizes computational efficiency, reducing model complexity and enabling faster, more effective decision-making, especially in scenarios where high-dimensional data can obscure insights and overburden resources. In sum, this FS strategy allows the model to zoom in on the most informative features, stripping away the irrelevant ones and thus honing predictive accuracy with a lean, high-performing subset of data. To achieve this goal, the test function is set as:

$$Fit = \gamma_1 \cdot err + \gamma_2 \frac{|R|}{|N|} \quad (23)$$

where err represents the classification error rate of the selected classifier, $|R|$ is the count of picked features in the data set, and $|N|$ is the original count of features in the data set. γ_1 as a penalty factor for controlling

classification accuracy, γ_2 as a factor identifying the count of picked features, and $\gamma_1 \in [0, 1]$, $\gamma_2 = 1 - \gamma_1$. In this paper, set $\gamma_1 = 0.9$, $\gamma_2 = 0.1$.

7.2 Trials and Analysis

To showcase the superiority of SCPSO in settling specific feature selection issues. We use the SVM classifier for testing. Six standard test sets from the database published by the University of California, Irvine (UCI) for machine learning were used in the test experiment, and the main details of the datasets are summarized in Table 15. Breast Cancer: This dataset is usually used to classify breast cancer. The SPECT dataset is mostly used to study heart disease diagnosis. Hepatitis: This dataset is often used for the task of hepatitis-related classification. Sonar: This dataset is usually used to classify sonar signals. Congress: This dataset is usually used to study voting patterns. Australian: This dataset is mainly used for research related to financial or credit card applications. Comparisons with SCPSO were made with SMA, Differential evolution (DE) [66], Atom Search Optimization (ASO) [67], GWO, ACO, and Gravitational Search Algorithm (GSA) [68], besides PSO. In addition, the population size and the count of iterations were set to 20 and 100, and each data set was executed 20 times independently.

Table 15: Details of the datasets

Data sets	Feature	Classification	Precedents
Breast Cancer	9	2	699
Spect	22	2	267
Hepatitis	29	2	155
Sonar	60	2	208
Congress	16	2	435
Australian	101	1	606

First, the data results of each algorithm on six test sets (including Breast Cancer, Spect, hepatitis, Sonar, Congress, and Australian) are presented in Table 16. In particular, it is worth noting that SCPSO has lower mean values than the other algorithms on all test sets, which reveals its superior performance in solving these issues. Additionally, the standard deviation is relatively small, which indicates that SCPSO also performs well in terms of stability. As a whole, the overall ranks of the algorithms are SCPSO > GSA > ASO > ACO > PSO > SMA = DE > GWO. SCPSO is ranked first with an average rank of 1, while PSO is ranked fifth with an average rank of 4.33, again emphasizing the introduced strategies' importance and effectiveness. The excellent performance of SCPSO may be attributed to the efficiency and robustness of its optimization process, which enables it to find better feature combinations on complex datasets. The stable and high performance of GSA indicates its adaptability in handling different types of data. The relatively weak performance of PSO, SMA, DE, and GWO may be due to the lack of adaptability of these methods to different data feature distributions and complexity, resulting in poor results for some datasets.

Table 16: Data outcomes of each FS algorithm

Datasets	Indicators	SCPSO	PSO	SMA	DE	ASO	GWO	ACO	GSA
Breast cancer	Best	0.02647	0.02647	0.03000	0.02647	0.02647	0.03000	0.02647	0.02647
	Average	0.04434	0.04778	0.05364	0.05081	0.04863	0.05351	0.04499	0.04654
	Worst	0.06532	0.06885	0.07180	0.06885	0.07237	0.06885	0.06532	0.06885
	Std	0.01109	0.01137	0.00998	0.01319	0.01329	0.01026	0.01088	0.01244
	Rank	1	4	8	6	5	7	2	3
Spect	Best	0.07367	0.07487	0.11792	0.11003	0.09065	0.08731	0.09640	0.08731
	Average	0.11676	0.12979	0.15668	0.14493	0.12867	0.14866	0.13520	0.12872
	Worst	0.18010	0.18465	0.19134	0.19949	0.18010	0.21192	0.17556	0.19134
	Std	0.02717	0.02915	0.02247	0.02658	0.02399	0.02865	0.02391	0.02569
	Rank	1	4	8	7	2	6	5	3
Hepatitis	Best	0.00526	0.01053	0.00526	0.07640	0.00526	0.01579	0.00526	0.00526
	Average	0.09813	0.11304	0.12243	0.14081	0.10829	0.14690	0.10856	0.10736
	Worst	0.16621	0.18472	0.17946	0.19253	0.17946	0.27963	0.17946	0.17946
	Std	0.04066	0.04317	0.04811	0.03550	0.04207	0.05819	0.04264	0.04308
	Rank	1	5	6	7	3	8	4	2
Sonar	Best	0.03362	0.04500	0.08947	0.06862	0.04862	0.04833	0.05057	0.06195
	Average	0.07748	0.10006	0.14993	0.12544	0.10553	0.13069	0.11688	0.10694
	Worst	0.14004	0.15476	0.20756	0.18142	0.17004	0.25423	0.18199	0.18866
	Std	0.03339	0.03202	0.03490	0.03300	0.03309	0.04693	0.03380	0.02908
	Rank	1	2	8	6	3	7	5	4
Congress	Best	0.02694	0.02694	0.02694	0.03944	0.02694	0.03750	0.02694	0.02694
	Average	0.04857	0.05034	0.05014	0.06203	0.04960	0.06068	0.04908	0.04888
	Worst	0.06832	0.06832	0.06832	0.09332	0.06832	0.08707	0.06832	0.06832
	Std	0.01119	0.01076	0.01095	0.01347	0.01086	0.01322	0.01078	0.01089
	Rank	1	6	5	8	4	7	3	2
Australian	Best	0.37990	0.39290	0.39478	0.40890	0.38990	0.41134	0.38734	0.39090
	Average	0.39655	0.41241	0.41007	0.42524	0.40570	0.43049	0.40401	0.40727
	Worst	0.41853	0.44153	0.43540	0.44953	0.43153	0.46297	0.42753	0.43553
	Std	0.01064	0.01427	0.01062	0.01237	0.01359	0.01418	0.01059	0.01271
	Rank	1	5	6	7	3	8	2	4
Mean rank		1.00	4.33	6.83	6.83	3.33	7.17	3.5	3.00
Final rank		1	5	6	7	3	7	4	2

In Fig. 17, we can view the convergence plot of each binary algorithm. Among them, the red curve represents the convergence trend of SCPSO. It is apparent from the observation that SCPSO converges more swiftly compared to other algorithms, and the minimum value reached is also the smallest. This observation further highlights the superiority of SCPSO in solving FS issues. Its fast convergence speed signifies that it can find the optimal subset of features more efficiently. At the same time, the minimum value reached indicates that it can find a more accurate prediction model. Not only that, the boxplot of SCPSO and the individual algorithms are also shown in Fig. 18, which also affirms the performance of SCPSO in terms of stability.

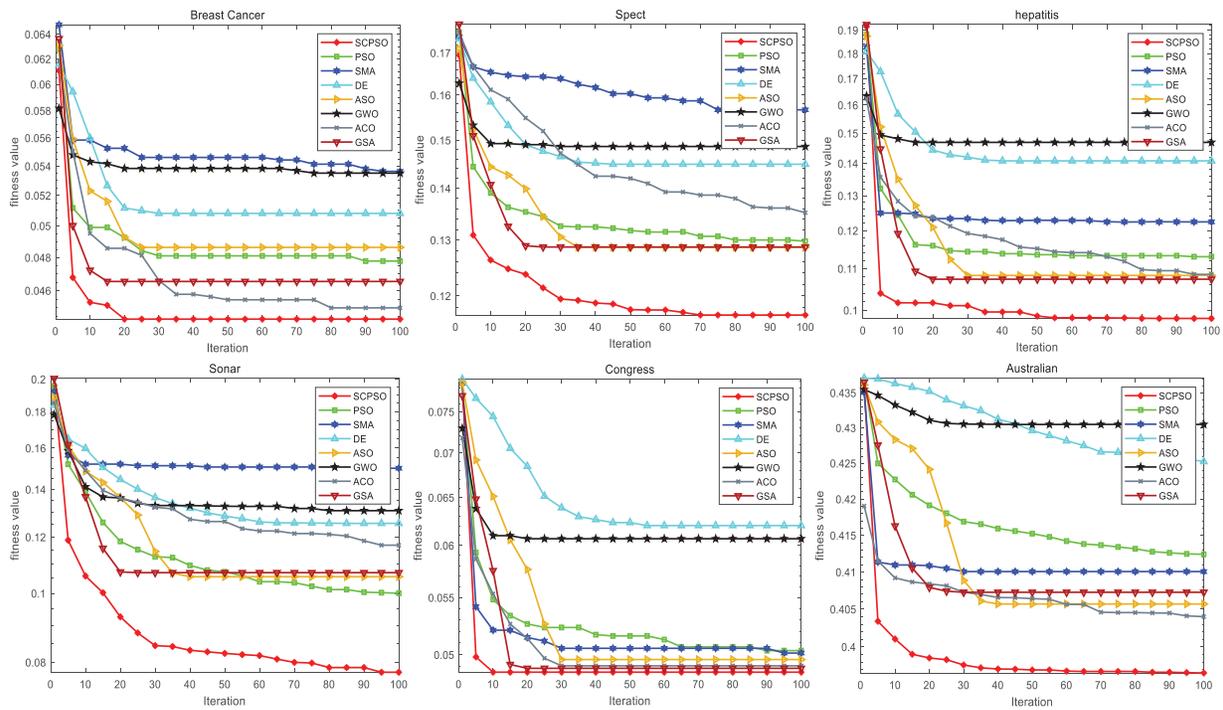


Figure 17: Convergence plot of each FS algorithm

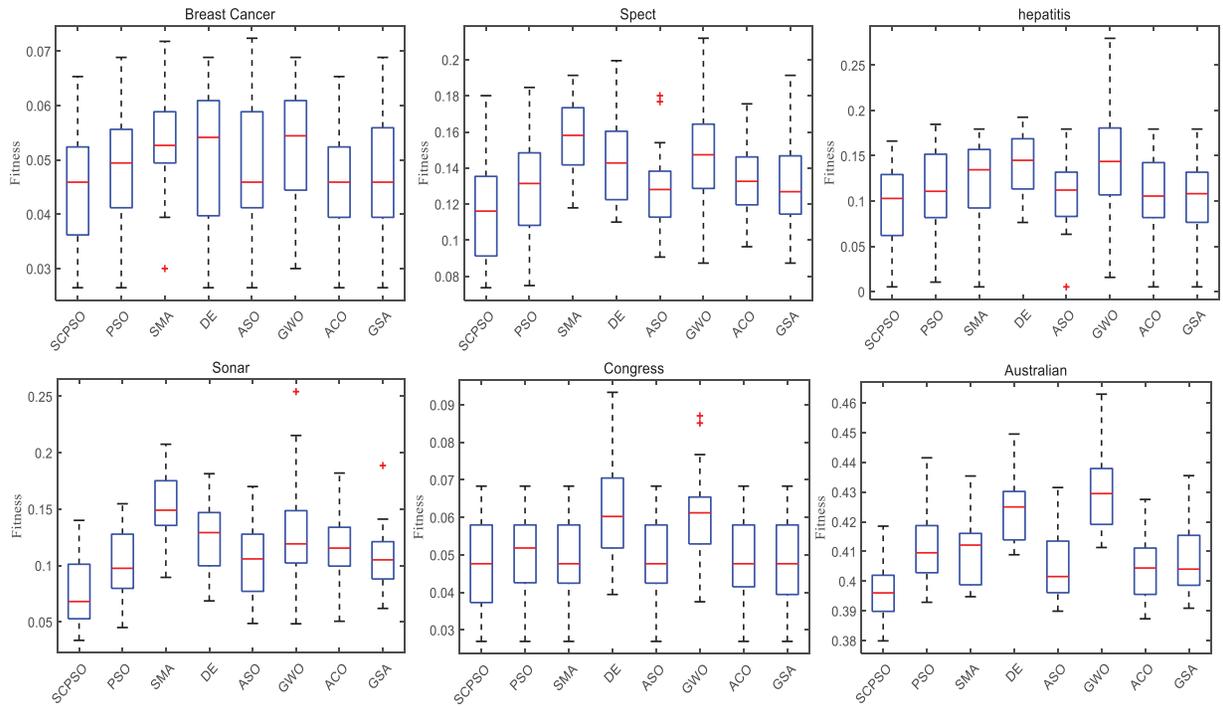


Figure 18: Boxplot of each FS algorithm

Feature selection (FS) primarily aims to maximize classification accuracy. Table 17 presents each algorithm's average classification accuracy and ranking across six datasets, with a bar plot in Fig. 19 visualizing

these averages. Notably, SCPSO (Self-adaptive Cooperative Particle Swarm Optimization) shows varying effects depending on the dataset, with standout performance on several. SCPSO ranks first in solving Spect, hepatitis, Sonar, and Australian datasets, showcasing its ability to handle these tasks effectively. This consistent top-ranking highlights SCPSO's robustness and adaptability, making it an ideal choice for complex feature selection problems in these domains. On the Breast Cancer dataset, SCPSO did not outperform SMA (Slime Mould Algorithm) and GSA (Gravitational Search Algorithm), but it still secured third place. This ranking indicates that SCPSO remains a competitive choice, even though other algorithms have a slight edge on this dataset. However, SCPSO ranks fifth on the Congress dataset, suggesting it faces challenges in addressing this data type. This result points to potential areas for SCPSO's improvement and to develop optimized strategies tailored for Congress-type data.

Table 17: Average classification accuracy of each algorithm

Datasets	SCPSO	PSO	SMA	DE	ASO	GWO	ACO	GSA
Breast cancer	97.5680 (3)	97.3022 (7)	96.7626 (1)	97.4101 (5)	97.3741 (6)	97.5540 (4)	88.4173 (8)	97.6619 (2)
Spect	89.9057 (1)	88.9623 (5)	87.6415 (6)	89.5283 (2)	89.3396 (3)	88.9623 (5)	78.5849 (7)	89.0566 (4)
Hepatitis	90.9677 (1)	89.5161 (4)	87.7419 (6)	89.3548 (5)	89.8387 (3)	87.7419 (6)	78.5484 (7)	90.3226 (2)
Sonar	93.7805 (1)	93.0488 (2)	85.6098 (6)	92.6829 (3)	92.5610 (4)	91.9512 (5)	69.5122 (7)	92.6829 (3)
Congress	95.4023 (5)	95.5172 (4)	95.5747 (3)	95.7471 (2)	95.2874 (6)	95.8621 (1)	68.2184 (7)	95.4023 (5)
Australian	57.0661 (1)	56.8595 (5)	55.5372 (7)	56.9008 (4)	56.9835 (2)	56.8182 (6)	52.5207 (8)	56.9421 (3)

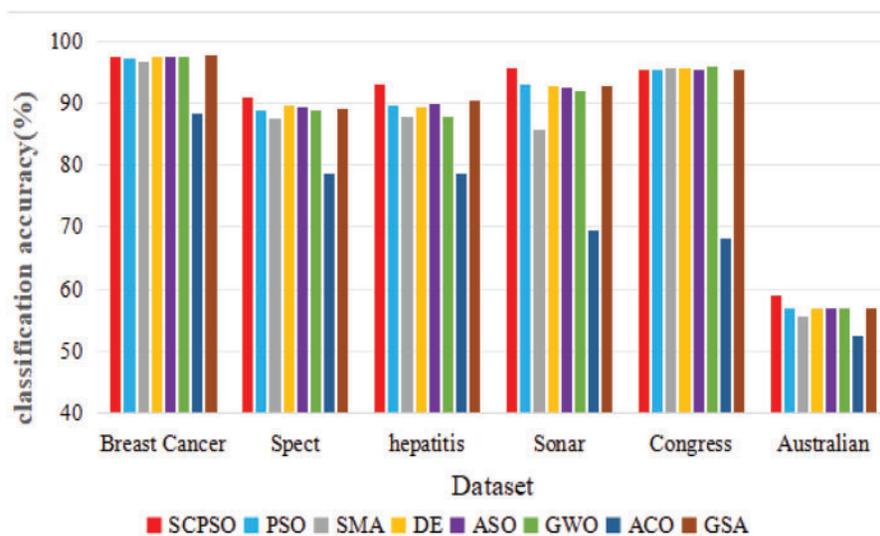


Figure 19: Bar plot of the average classification accuracy of each algorithm

SCPSO excels across most datasets, but its relatively weaker performance in Congress highlights an opportunity to refine the algorithm and expand its versatility in feature selection applications.

The second goal of the FS problem is to keep the number of features as small as possible. In Table 18, we collect the average size of the feature subset for the eight algorithms. These results show that all algorithms achieve some degree of reduction in the dimension of the original data set, i.e., successful feature selection. Specifically, with fewer features, SCPSO ranked first in Breast Cancer, Spect, Sonar, and Congress. Moreover, it is slightly inferior to SMA on hepatitis, Australian, and ranks second.

Table 18: Average size of feature subset for each algorithm

Datasets	SCPSO	PSO	SMA	DE	ASO	GWO	ACO	GSA
Breast Cancer	2.2 (1)	2.35 (3)	2.45 (4)	2.75 (7)	2.5 (5)	3.15 (8)	2.3 (2)	2.55 (6)
Spect	5.70 (1)	6.70 (3)	10.00 (6)	11.15 (8)	7.20 (4)	10.85 (7)	8.45 (5)	6.65 (2)
Hepatitis	3.2 (2)	3.55 (4)	2.3 (1)	8.55 (7)	3.2 (2)	6.95 (6)	3.25 (3)	3.85 (5)
Sonar	12.25 (1)	22.5 (4)	12.9 (2)	35.75 (8)	23.15 (5)	34.95 (7)	14.15 (3)	24.65 (6)
Congress	1.15 (1)	1.6 (3)	1.65 (4)	3.8 (6)	1.15 (1)	3.75 (5)	1.15 (1)	1.2 (2)
Australian	10.15 (2)	24.15 (6)	9.9 (1)	37.35 (7)	18.55 (5)	41.85 (8)	11.65 (3)	19.75 (4)

In summary, SCPSO iteratively selects features by evaluating the impact of feature subsets on model performance. To distinguish between related and unrelated features, the model reduces the probability of selecting features that contribute less to performance. That is achieved by strengthening particles that perform well (subsets with higher fitness) and punishing or abandoning particles that do not improve model performance. Therefore, as the iteration progresses, the algorithm will gradually converge to a feature set that contributes the most to prediction accuracy.

8 Conclusion and Future Direction

This paper presents an enhanced PSO algorithm called SCPSO. It demonstrates high convergence accuracy and stability on a complex CEC test suite and three engineering problems, proving its superb generalization ability and usefulness. He demonstrates its ability to reduce data dimensionality and improve model performance by selecting relevant features. The integration of SCSO into the traditional PSO framework is a key improvement that endows SCPSO with extended global search capabilities. This is attributed to SCSO's ability to mimic the hunting behavior of sand cats, which are adept at detecting low-frequency vibrations to locate their prey, thus exploring the solution space more efficiently and reducing the likelihood of getting stuck in local optima.

Employing a Latin hypercube sampling strategy during the initialization phase has proven to be a strategic move. This approach ensures a more even distribution of the initial population in the search space, enhances population diversity, and provides broader coverage from the outset. This initialization technique is particularly effective in high-dimensional problems where random initialization may lead to clustering

of solutions in some domains, ignoring others, and potentially missing the global optimum. In addition, inspired by the cooperative hunting behavior of Grey wolves, a roundup search strategy is introduced to accelerate the convergence of SCPSO. By selecting leaders based on fitness and allowing them to guide the group toward the goal, SCPSO demonstrates an improved ability to find optimal solutions quickly.

Improvements to the algorithm will continue in the future. On the one hand, although SCPSO has achieved some successes, the intake search strategy has been shown to improve the convergence speed but inhibit the accuracy of the solution on individual functions. Therefore, subsequent research can focus on finding better strategies. In addition, the transfer function helps to influence the way the algorithm navigates through the feature space, leading to changes in the quality and efficiency of feature selection. Therefore, more work needs to be done to measure the applicability of SCPSO in solving other FS problems using other transfer functions, such as the V-function. Finally, SCPSO can be applied to more problems, such as UAV path planning, time series prediction, etc.

Acknowledgement: None.

Funding Statement: This work was supported by the Fundamental Research Funds for the Central Universities of China (No. 300102122105), the Natural Science Basic Research Plan in Shaanxi Province of China (2023-JC-YB-023).

Author Contributions: Xing Wang: Methodology, Software, Investigation, Formal analysis, Funding acquisition, Writing—original draft. Huazhen Liu: Data curation, Resources, Writing—original draft. Abdelazim G. Hussien: Visualization, Analysis, Supervision, Investigation. Gang Hu: Writing—review & editing, Supervision. Li Zhang: Conceptualization, Writing—review & editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Ma X, Xu H, Liu Y, Zhang JZ. Class-specific feature selection using fuzzy information-theoretic metrics. *Eng Appl Artif Intell.* 2024;136(12):109035. doi:10.1016/j.engappai.2024.109035.
2. Fan Y, Liu J, Tang J, Liu P, Lin Y, Du Y. Learning correlation information for multi-label feature selection. *Pattern Recogn.* 2024;145(8):109899. doi:10.1016/j.patcog.2023.109899.
3. Mostafa RR, Ewees AA, Ghoniem RM, Abualigah L, Hashim FA. Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection. *Knowl-Based Syst.* 2022;246(9):108743. doi:10.1016/j.knosys.2022.108743.
4. Hu G, Zhong J, Wang X, Wei G. Multi-strategy assisted chaotic coot-inspired optimization algorithm for medical feature selection: a cervical cancer behavior risk study. *Comput Biol Med.* 2022;151(1):106239. doi:10.1016/j.compbimed.2022.106239.
5. Samieiyan B, MohammadiNasab P, Mollaei MA, Hajizadeh F, Kangavari M. Novel optimized crow search algorithm for feature selection. *Expert Syst Appl.* 2022;204(3):117486. doi:10.1016/j.eswa.2022.117486.
6. Braik MS, Hammouri AI, Awadallah MA, Al-Betar MA, Khatatneh K. An improved hybrid chameleon swarm algorithm for feature selection in medical diagnosis. *Biomed Signal Process Control.* 2023;85(8):105073. doi:10.1016/j.bspc.2023.105073.
7. Kamel SR, Yaghoubzadeh R. Feature selection using grasshopper optimization algorithm in diagnosis of diabetes disease. *Inform Med Unlocked.* 2021;26(12):100707. doi:10.1016/j.imu.2021.100707.

8. Zadsafar F, Tabrizchi H, Parvizpour S, Razmara J, Lotfi S. A model for Mesothelioma cancer diagnosis based on feature selection using Harris hawk optimization algorithm. *Comput Methods Programs Biomed Update*. 2022;2(4):2022–100078. doi:10.1016/j.cmpbup.2022.100078.
9. Yan H, Li Q, Tseng M, Guan X. Joint-optimized feature selection and classifier hyperparameters by salp swarm algorithm in piano score difficulty measurement problem. *Appl Soft Comput*. 2023;144:110464. doi:10.1016/j.asoc.2023.110464.
10. Abd El-Mageed AA, Abohany AA, Elashry A. Effective feature selection strategy for supervised classification based on an improved binary Aquila optimization algorithm. *Comput Ind Eng*. 2023;181(4):109300. doi:10.1016/j.cie.2023.109300.
11. Hu G, Yang R, Qin X, Wei G. MCSA: multi-strategy boosted chameleon-inspired optimization algorithm for engineering applications. *Comput Method Appl Mech Eng*. 2023;403(18):115676. doi:10.1016/j.cma.2022.115676.
12. Hu G, Zheng Y, Abualigah L, Hussien AG. DETDO: an adaptive hybrid dandelion optimizer for engineering optimization. *Adv Eng Inform*. 2023;57(4):102004. doi:10.1016/j.aei.2023.102004.
13. Chen Z, Cao H, Ye K, Zhu H, Li S. Improved particle swarm optimization-based form-finding method for suspension bridge installation analysis. *J Comput Civil Eng*. 2015;29(3):4014047 doi:10.1061/(ASCE)CP.1943-5487.0000354.
14. Cheng R, Jin Y. A social learning particle swarm optimization algorithm for scalable optimization. *Inform Sci*. 2015;291(2):43–60. doi:10.1016/j.ins.2014.08.039.
15. Aydilek IB. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Appl Soft Comput*. 2018;66(2):232–49. doi:10.1016/j.asoc.2018.02.025.
16. Ha MP, Nazari-Heris M, Mohammadi-Ivatloo B, Seyedi H. A hybrid genetic particle swarm optimization for distributed generation allocation in power distribution networks. *Energy*. 2020;209:118218. doi:10.1016/j.energy.2020.118218.
17. Yang R, Liu Y, Yu Y, He X, Li H. Hybrid improved particle swarm optimization-cuckoo search optimized fuzzy PID controller for micro gas turbine. *Energy Rep*. 2021;7(1):5446–54. doi:10.1016/j.egy.2021.08.120.
18. Zainuddin Z, Lai KH, Ong P. An enhanced harmony search based algorithm for feature selection: applications in epileptic seizure detection and prediction. *Comput Electr Eng*. 2016;53(2):143–62. doi:10.1016/j.compeleceng.2016.02.009.
19. Paul D, Su R, Romain M, Sébastien V, Pierre V, Isabelle G. Feature selection for outcome prediction in oesophageal cancer using genetic algorithm and random forest classifier. *Comput Med Imag Grap*. 2017;60(5):42–9. doi:10.1016/j.compmedimag.2016.12.002.
20. Tu Q, Chen X, Liu X. Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Appl Soft Comput*. 2019;76(99):16–30. doi:10.1016/j.asoc.2018.11.047.
21. Alzaqebah M, Briki K, Alrefai N, Brini S, Jawarneh S, Alsmadi MK, et al. Memory based cuckoo search algorithm for feature selection of gene expression dataset. *Inform Med Unlocked*. 2021;24(2):100572. doi:10.1016/j.imu.2021.100572.
22. Khorashadizade M, Hosseini S. An intelligent feature selection method using binary teaching-learning based optimization algorithm and ANN. *Chemometr Intell Lab Syst*. 2023;240(5):104880. doi:10.1016/j.chemolab.2023.104880.
23. Zhao Y, Dong J, Li X, Chen H, Li S. A binary dandelion algorithm using seeding and chaos population strategies for feature selection. *Appl Soft Comput*. 2022;125(5):109166. doi:10.1016/j.asoc.2022.109166.
24. Braik M, Hammouri A, Alzoubi H, Sheta A. Feature selection based nature inspired capuchin search algorithm for solving classification problems. *Expert Syst Appl*. 2024;235(1):121128. doi:10.1016/j.eswa.2023.121128.
25. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks; 1995; Perth, WA, Australia*. Vol. 4, p. 1942–8. doi:10.1109/ICNN.1995.488968.
26. He Z, Pan Y, Wang K, Xiao L, Wang X. Area optimization for MPRM logic circuits based on improved multiple disturbances fireworks algorithm. *Appl Math Comput*. 2021;399(9):126008. doi:10.1016/j.amc.2021.126008.

27. Hua L, Zhang C, Peng T, Ji C, Nazir MS. Integrated framework of extreme learning machine (ELM) based on improved atom search optimization for short-term wind speed prediction. *Energ Convers Manage.* 2022;252(30):115102. doi:10.1016/j.enconman.2021.115102.
28. Seyyedabbasi A, Kiani F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. *Eng Comput.* 2023;39(4):2627–51. doi:10.1007/s00366-022-01604-x.
29. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw.* 2014;69:46–61. doi:10.1016/j.advengsoft.2013.12.007.
30. Mohamed AW, Hadi AA, Mohamed AK, Awad NH. Evaluating the performance of adaptive gainingsharing knowledge based algorithm on CEC 2020 benchmark problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC); 2020; Glasgow, UK. p. 1–8. doi:10.1109/CEC48606.2020.9185901.
31. Yazdani D, Branke J, Omidvar MN, Li X, Li C, Mavrouniotis M, et al. IEEE CEC 2022 competition on dynamic optimization problems generated by generalized moving peaks benchmark. arXiv:2106.06174. 2021.
32. Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. In: Technical report. Zhengzhou, China: Computational Intelligence Laboratory, Zhengzhou University. Singapore: Nanyang Technological University; 2014.
33. Awad N, Ali M, Liang J, Qu B, Suganthan P. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. In: Technical report; 2016. p. 5–8.
34. Hu G, Guo Y, Zhong J, Wei G. IYDSE: ameliorated Young's double-slit experiment optimizer for applied mechanics and engineering. *Comput Method Appl Mech Eng.* 2023;412(1):116062. doi:10.1016/j.cma.2023.116062.
35. Mirjalili S, Wang G, Coelho LDS. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Comput Appl.* 2014;25(6):1423–35. doi:10.1007/s00521-014-1629-6.
36. Coelho LS. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst Appl.* 2010;37(2):1676–83. doi:10.1016/j.eswa.2009.06.044.
37. Ye W, Feng W, Fan S. A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Appl Soft Comput.* 2017;61(5):832–43. doi:10.1016/j.asoc.2017.08.051.
38. Sun J, Wu X, Palade V, Fang W, Shi Y. Random drift particle swarm optimization algorithm: convergence analysis and parameter selection. *Mach Learn.* 2015;101(1–3):345–76. doi:10.1007/s10994-015-5522-z.
39. Haklı H, Uğuz H. A novel particle swarm optimization algorithm with Levy flight. *Appl Soft Comput.* 2014;23(3):333–45. doi:10.1016/j.asoc.2014.06.034.
40. Yapıcı H, Çetinkaya N. An improved particle swarm optimization algorithm using eagle strategy for power loss minimization. *Math Probl Eng.* 2017;2017(1):1063045. doi:10.1155/2017/1063045.
41. Wu D, Rao H, Wen C, Jia H, Liu Q, Abualigah L. Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems. *Mathematics.* 2022;10(22):4350. doi:10.3390/math10224350.
42. Wang X, Liu Q, Zhang L. An adaptive sand cat swarm algorithm based on cauchy mutation and optimal neighborhood disturbance strategy. *Biomimetics.* 2023;8(2):191. doi:10.3390/biomimetics8020191.
43. Hu G, Zhong J, Zhao C, Wei G, Chang C. LCAHA: a hybrid artificial hummingbird algorithm with multi-strategy for engineering applications. *Comput Method Appl Mech Eng.* 2023;415(4):116238. doi:10.1016/j.cma.2023.116238.
44. Abualigah L, Elaziz MAbd, Sumari P, Geem ZW, Gandomi AH. Reptile Search Algorithm (RSA): a nature-inspired meta-heuristic optimizer. *Expert Syst Appl.* 2022;191(11):116158. doi:10.1016/j.eswa.2021.116158.
45. Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw.* 2016;95(12):51–67. doi:10.1016/j.advengsoft.2016.01.008.
46. Minh H, Sang-To T, Theraulaz G, Wahab MA, Cuong-Le T. Termite life cycle optimizer. *Expert Syst Appl.* 2023;213:119211. doi:10.1016/j.eswa.2022.119211.
47. Abualigah L, Diabat A. Advances in sine cosine algorithm: a comprehensive survey. *Artif Intell Rev.* 2021;54(4):2567–608. doi:10.1007/s10462-020-09909-3.
48. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst.* 2019;97:849–72. doi:10.1016/j.future.2019.02.028.

49. Chopra N, Ansari MM. Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. *Expert Syst Appl.* 2022;198(5):116924. doi:10.1016/j.eswa.2022.116924.
50. Dehghani M, Montazeri Z, Trojovská E, Trojovský P. Coati optimization algorithm: a new bio-inspired meta-heuristic algorithm for solving optimization problems. *Knowl-Based Syst.* 2023;259(1):110011. doi:10.1016/j.knosys.2022.110011.
51. Houssein EH, Hussain K, Abualigah L, Abd Elaziz M, Alomoush W, Dhiman G, et al. An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. *Knowl-Based Syst.* 2021;229(1):107348. doi:10.1016/j.knosys.2021.107348.
52. Kumar A, Wu G, Ali MZ, Mallipeddi R, Suganthan PN, Das S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol Comput.* 2020;56:100693. doi:10.1016/j.swevo.2020.100693.
53. Meng X, Gao XZ, Lu L, Liu Y, Zhang H. A new bio-inspired optimisation algorithm: bird Swarm Algorithm. *J Exp Theor Artif Intell.* 2016;28:673–87. doi:10.1080/0952813X.2015.1042530.
54. Abualigah L, Younsri D, Abd Elaziz M, Ewees AA, Al-Qaness MA, Gandomi AH. Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput Ind Eng.* 2021;157(11):107250. doi:10.1016/j.cie.2021.107250.
55. Zhong C, Li G, Meng Z. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm. *Knowl-Based Syst.* 2022;251(1):109215. doi:10.1016/j.knosys.2022.109215.
56. Gu L, Yang RJ, Tho C, Makowskit M, Faruquet O, Li YLY. Optimisation and robustness for crashworthiness of side impact. *Int J Vehicle Des.* 2001;26(4):348–60. doi:10.1504/IJVD.2001.005210.
57. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. *Future Gener Comput Syst.* 2020;111(Suppl C):300–23. doi:10.1016/j.future.2020.03.055.
58. Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng Appl Artif Intel.* 2022;114(4):105082. doi:10.1016/j.engappai.2022.105082.
59. Dhiman G, Kumar V. Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl-Based Syst.* 2019;165(25):169–96. doi:10.1016/j.knosys.2018.11.024.
60. Abdollahzadeh B, Gharehchopogh FS, Khodadadi N, Mirjalili S. Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Adv Eng Softw.* 2022;174(3):103282. doi:10.1016/j.advengsoft.2022.103282.
61. Aydemir SB. Enhanced marine predator algorithm for global optimization and engineering design problems. *Adv Eng Softw.* 2023;184(2):103517. doi:10.1016/j.advengsoft.2023.103517.
62. Mirjalili S. The ant lion optimizer. *Adv Eng Softw.* 2015;83:80–98. doi:10.1016/j.advengsoft.2015.01.010.
63. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. *Simul-t Soc Mod Sim.* 2001;76:60–8. doi:10.1177/003754970107600201.
64. Abualigah L, Diabat A, Mirjalili S, Elaziz MA, Gandomi AH. The arithmetic optimization algorithm. *Comput Method Appl Mech Eng.* 2021;376(2):113609. doi:10.1016/j.cma.2020.113609.
65. Houssein EH, Oliva D, Celik E, Emam MM, Ghoniem RM. Boosted sooty tern optimization algorithm for global optimization and feature selection. *Expert Syst Appl.* 2023;213(168):119015. doi:10.1016/j.eswa.2022.119015.
66. Price KV. Differential evolution. In: *Handbook of optimization: from classical to modern approach.* Berlin/Heidelberg: Springer; 2013. p. 187–214.
67. Zhao W, Wang L, Zhang Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl-Based Syst.* 2019;163(4598):283–304. doi:10.1016/j.knosys.2018.08.030.
68. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Inform Sci.* 2009;179(13):2232–48. doi:10.1016/j.ins.2009.03.004.