Complex & Intelligent Systems (2023) 9:6611–6643
https://doi.org/10.1007/s40747-023-01069-5

ORIGINAL ARTICLE

A particle swarm optimization algorithm based on diversity-driven fusion of opposing phase selection strategies

Jiucheng Xu[1,2] · Shihui Xu[1,2] · Lei Zhang[1,2] · Changshun Zhou[1,2] · Ziqin Han[1,2]

Received: 25 November 2022 / Accepted: 1 April 2023 / Published online: 26 May 2023

Abstract

Opposition-based learning (OBL) is often embedded in intelligent optimization algorithms to solve practical engineering and mathematical problems, but the combinatorial problems among different OBL variants are rarely studied. To this end, we propose a novel OBL variant based on the principle of optical imaging, which combines two novel types of quasi-opposite learning and extended opposite learning, called diversity-driven fused opposition learning (SQOBL). First, a density center based on a neighborhood model is proposed. Based on the rapid convergence of the centroid, combined the advantages of density and centroid to construct a double mean center (DMC) to replace the original center point in quasi-opposite learning based on the principle of refraction. Secondly, an extended opposite learning method based on optical refraction imaging is proposed. Diversity is then exploited to drive different opposing learning strategies at different stages of evolution, thus controlling the exploration and utilization of the algorithm. Finally, SQOBL was embedded in the PSO with eight others representative OBL variants to find the most optimal solution for a test suite. In addition, 8 novel intelligent optimization algorithms and the first three algorithms were selected to evaluate the performance of the latest CEC2022 benchmark test set and realistic constrained optimization problems. Experiments with 56 test functions and 3 real-world constraint optimization problems show that the proposed SQOBL has good integrative properties in CEC2015, CEC2017, CEC2020, and CEC2022 test suites.

Keywords Opposition-based learning · Particle swarm optimization · Exploration and exploitation · Optimization problems

Introduction

With the development of human cognition, complex optimization problems arise in many fields, such as material design [1], remote sensing [2], energy consumption [3] and epidemic control [4]. Traditional optimization algorithms are only suitable for solving small-scale problems because of their large computation volume, and it is often difficult to be effective in practical engineering applications. In recent years, many intelligent optimization algorithms have come into being due to the influence of biological evolution or group behavior in nature [5–11].

The meta-heuristic optimization algorithm is a commonly used method to solve global optimization problems. It is mainly achieved by simulating the intelligence of nature and people. Broadly speaking, metaheuristic algorithms can be divided into four branches: biological evolutionary algorithms, algorithms based on physical and chemical laws, algorithms based on human behavior, and algorithms based on group intelligence. Evolutionary algorithms are inspired by the evolution of living things in nature. Based on Darwin's theory's theory of evolution, simulates natural selection and survival of the fittest to achieve the overall evolution of the population. Common evolutionary algorithms include

✉ Shihui Xu
mutezoo@163.com

Jiucheng Xu
jiuchxu@gmail.com

Lei Zhang
2108283069@stu.htu.edu.cn

Changshun Zhou
2108283068@stu.htu.edu.cn

Ziqin Han
qin15737355567@163.com

[1] College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

[2] Engineering Lab of Intelligence Business and Internet of Things, Henan Normal University, Xinxiang 453007, Henan, China

genetic algorithm [12], differential evolution algorithm [13] and optimization algorithm based on biogeography [14]. Algorithms based on the laws of physics and chemistry are mainly derived from the laws of physics and chemical reactions in the universe. Famous algorithms based on physics and chemistry include simulated annealing [15], big-bang big-crunch [16], gravity search [17], and black hole algorithm [18]. Human-centered algorithm is mainly inspired by human behavior, such as teaching behavior, social behavior, emotional behavior, etc. Learner performance-based behavior optimization [19], children's drawing development optimization [20], socio evolution and learning optimization algorithm [21], volleyball premier league algorithm [22] is an intelligent optimization algorithm based on human behavior. Group intelligence is a major branch of intelligent computing. The group intelligence optimization algorithm achieves the global optimal solution by simulating the intelligence of the group. In this algorithm, each group is a biological group, and cooperative behavior among individuals in the group can accomplish tasks that individuals cannot. So far, various swarm intelligence algorithms have been widely studied to solve different optimization problems, such as particle swarm optimization [23], gray wolf optimizer [24], whale optimization algorithm [25], fox algorithm [26], and nest algorithm [27], and fitness-dependent optimizer [28].

The success swarm intelligence algorithm is largely due to the adjustment of parameters which reduces the overall complexity of the algorithm, the calculation cost, and the time consumption. The common goal of all optimization algorithms is to obtain the highest-quality solution at the lowest cost. Different types of optimization problems require appropriate algorithms. No optimization algorithm in the world is universally applicable, that is, it can solve all kinds of optimization problems. Therefore, the central idea of dealing with optimization problems is to choose the appropriate optimization method wisely to solve a given optimization problem with less effort and higher convergence performance [29]. Eberhardt and Kennedy [23] introduced particle swarm optimization as an optimization method, the basic idea of which is derived from the study of bird foraging behavior. The whole algorithm is simple in structure, involves few parameters, and is easy to implement. It has a strong global search ability and fast convergence speed. It is a well-known swarm intelligence algorithm widely used in hybrid technology. Among many swarm intelligence algorithms, Particle Swarm Optimization (PSO) has been widely favored by researchers in the field of optimization because of its simplicity and adaptability [30]. TAREQM. SHAMI et al [31] have thoroughly and rigorously examined the basic concepts of PSO optimization, neighborhood topology, binary particle swarm optimization, recent variations of particle swarm optimization, important engineering applications and their drawbacks, and many researchers are working to improve particle swarm algorithm.

The PSO algorithm is one of the most acclaimed swarm intelligence algorithms in the literature, and it showed good characteristics of fast convergence speed and high solution efficiency when it was proposed. However, the closer we get to the most optimal particle, the more "convergent" particle population swarms become, unable to take full advantage of the information gained in the search and prone to local optimality. Therefore, it is necessary to dynamically coordinate the relationship between overall exploration and local exploitation in order to obtain optimization results. How to grasp the balance of exploration and development is the basis of using swarm intelligence algorithms to solve problems [32]. Therefore, people try to improve the PSO algorithm from different perspectives and research backgrounds.

At present, the improvement direction of PSO is divided into three categories: (1) the study of important parameters [33–35] and the selection of topology among particles [36–38]. (2) Revision of velocity update strategy [39–41] and particle location formula [42–44]; (3) Keep the fundamental rules of particle interaction the same and combine other intelligent methods to mine useful information from intermediate data to improve individual learning as a complement to particle interaction [45–48]. Opposition-Based Learning (OBL) is one of them.

OBL is an intelligent technology developed by Tizhoosh [49]. The main idea is to evaluate both current and reverse solutions and use the best solutions to speed up the search process. If the fundamental rules of interaction between particles resemble the skeleton of an intelligent optimization algorithm, then OBL is like a soul. Without strong bones, the soul has no support. Instead, if you lose your soul, it's like walking dead. OBL can be combined with swarm intelligence algorithms to solve engineering and math problems [50], often with better results than individual intelligence algorithms. The combination of PSO and OBL has two main aspects. On the one hand, the idea of OBL is used to update particle parameters and selection strategies, such as the inertia weight based on the idea of OBL [51]. Another aspect is to embed the OBL framework into the particle initialization phase and local renewal strategies based on the jump rate. The particle swarm optimization algorithm (OPSO) based on opposite-based learning was originally proposed in 2007 [52]; Wang et al. [53] extended the opposite-based learning and proposed the generalized opposite-based learning particle swarm optimization algorithm (GOPSO); in [54], GOPSO is complemented with an adaptive mutation selection strategy for performing a local search process on globally optimal particles; Tang et al. [55] designed an enhanced opposite learning strategy controlled by an adversarial probability parameter, which is similar to particle group optimization algorithm combination (EOPSO); Shao Peng et

al. [56] improved the opposite-based learning strategy with the principle of refraction in physics, and combined it with PSO to apply it to the design of FIR digital filtering.

Through these studies, the application of OBL to the PSO algorithm has obtained good results [56]. However, the current research on OBL also has the following shortcomings: (1) in recent years, scholars have used different OBL variants to improve the degree of particle exploration and development [57], but there is no research on the combination of extended opposition variants and quasi-opposition variants; (2) the center of gravity opposite learning strategy using the group search experience has stronger learning ability than the opposite-based learning strategy using the search space boundary [58], but the selection of the center of mass will ignore the favorable information carried by the overall optimization, such as the population dynamic distribution.

It is well known that the current research on OBL variants mainly focuses on quasi-opposition [52, 53], and there is little research on expanding opposition mutations. Extending opposition learning has unique advantages, which can expand the global search ability of population to a certain extent and avoid premature convergence. Therefore, it is necessary to study the variant of extended objection, which supports the construction of a comprehensive OBL variant system. The quasi-opposition variant significantly improves the local search ability of the population. It is interesting to observe the changes in the performance of the algorithm by combining the advantages of the quasi-opposition variant and the extended opposition variant and embedding them into the swarm intelligence optimization algorithm.

To this end, we propose a diversity-driven opposite-fusion strategy (SQOBL) that utilizes refraction imaging principles in physics to construct extended opposites and quasi-opposition variants to balance the exploration and development capabilities of this population. At the same time, the mathematical properties of density are used to redefine the population center and improve the quasi-opposite variant, considering that the center of mass is not representative of the population as a whole. The proposed SQOBL strategy is compared with experimental results from CEC2015, CEC2017, CEC2020, and CEC2022 test suites to validate the algorithm.

Overall, the main innovations of this paper are summarized as follows:

1. A density definition method based on a neighborhood model is proposed. Combining the advantages of density and centroid, a double mean centroid is constructed to replace the original centroid and improve the learning ability of quasi-opposing variables;
2. Inspired by optical refraction imaging, a new extended opposition OBL variant is constructed;
3. In order to balance the ability of exploration and development, diversity-driven different opposition learning variants are used at different stages of optimization, called the diversity-driven fusion of opposition strategies (SQOBL);
4. For the first time, extended opposition and quasi-opposition are combined, and the new OBL strategy constructed combines the advantages of the two variants.

The rest of this article is structured as follows. In the "Preliminaries" section, PSO, OPSO algorithms, and OBL-related variants are introduced; the next section introduces the combination of SQOBL and PSO. In the "Experiments" section, various benchmark sets are used to comprehensively assess SQOBL performance. Finally, in the "Statistical analysis" section, the statistical results of the experiment are presented.

Preliminaries

PSO algorithm

The PSO algorithm is generated by simulating the foraging behavior of a flock of birds. Initially, the position of each bird is random, and each bird is flying in different directions. The flock of birds transmits information through interaction, gradually gathers into small groups, and flies in the same direction at the same speed, and finally, the whole flock gathers at the same location, which is the source of food. In the basic PSO algorithm, on the one hand, the independent individuals in the population will carry simple and limited own-specific information and continuously exchange local information. On the other hand, it will form a group, complete the information dissemination of the whole population in the way of group behavior, and show excellent cooperation ability and intelligence. Particles search continuously by learning their own historical cognition (pbest) and group cognition (gbest) to achieve the purpose of optimization. For a D-dimensional search space, let the number of swarm particles be n, and the position and velocity of the i-th particle are denoted as $X_i = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,D}\right)$ and $V_i = \left(v_{i,1}, v_{i,2}, \ldots, v_{i,D}\right)$, $i = 1, 2, \ldots, n$, respectively. The formula for calculating the d $(1 \le d \le D)$-dimensional velocity and position of the i-th particle are as follows:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_1 \left(pbest_{i,d} - x_{i,d}(t)\right) + c_2 r_2 \left(gbest_d - x_{i,d}(t)\right) \quad (1)$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad (2)$$

Among them, $v_{i,d}(t+1)$ and $x_{i,d}(t+1)$ are the velocity and position of particle i in the $t+1$ -th generation respectively, ω is the inertia factor, the acceleration constants c_1

and c_2 are the weight parameters for adjusting the current optimal value and the global optimal value, r_1 and r_2 are [0, 1] uniformly distributed random numbers.

Combination of PSO basic algorithm and OBL

The combination of the PSO algorithm and OBL is usually divided into two ways, one is to use the opposite idea to improve the algorithm locally, and the other is to embed the OBL variant in the whole algorithm. In the BPSO algorithm, the idea of OBL is integrated into the individual optimal update equation of particles [59]. In [51], the ideas of opposition and rank are introduced to construct a new inertia weight, which is used to balance the degree of exploration and development of particle swarms, thereby speeding up the convergence speed.

The most basic OBL variant was first combined with the PSO algorithm called OPSO [52]. In OPSO, OBL is used in the population initialization phase and the local update strategy based on the jump rate. In the population initialization stage, the population $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ is randomly generated within the constraint range, and then n opposite solution$\tilde{X}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, \ldots, \tilde{x}_{i,D})$ of the random population is obtained based on the OBL strategy. The specific calculation formula is as follows:

$$\tilde{x}_{i,d} = a_i + b_i - x_{i,d} \tag{3}$$

where $x_i \in [a_i, b_i]$, $d = 1, 2, \ldots, D$.

Finally, select N individuals with the best fitness from $\left\{X(N) \cup \tilde{X}(N)\right\}$ to form the initial population.

In the update process, after the population completes the speed and position update, if a random number rand(0,1) obeying a uniform distribution is less than the set jump rate Jr, the population shows opposition, that is, the local OBL strategy is executed, and the specific update steps Similar to the initialization phase. First, generate N opposing positions corresponding to the original N positions, select the best from the original population and the opposing population, and select the N solutions with the best fitness as the current new population:

$$\tilde{x}_{i,d} = a_d + b_d - x_{i,d} \tag{4}$$

The boundary of the relative individual is different from the initialization stage, which changes with the iterative process, and a_d and b_d represent the interval boundary ($x_{i,d} \in [a_d, b_d]$) of the i-th variable d-dimension.

OBL variants and improvements

Tizhoosh et al. [49] proposed the idea of opposition learning in 2005, and then in 2008 provided the relevant definitions of the type-I opposite point and the type-II opposite point [60].

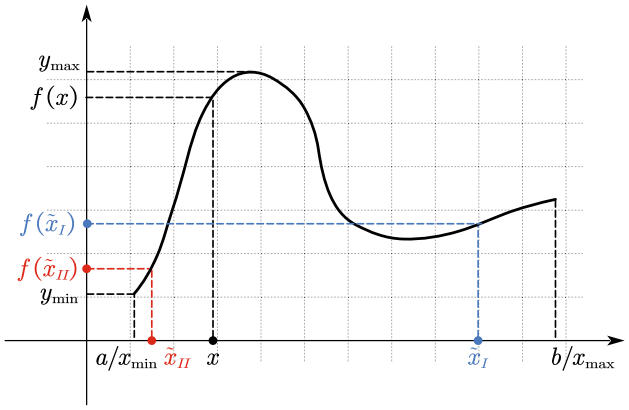


Fig. 1 Example samples of Type-I and Type-II opposition points

Type-I opposite points are aimed at the relationship between points in the search space, and Type-II opposite points consider the relationship between the target values of points in the space.

Definition 1 (*Type-I opposite points*) [60] Given $X = (x_1, x_2, \ldots, x_D)$ be a point in D-dimensional space, where $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, \ldots, D\}$. The type-I opposite of X is defined by $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_D)$ as follow:

$$\tilde{x}_I = a + b - x \tag{5}$$

Definition 2 (*Type-II opposite points*) [60] Given $X = (x_1, x_2, \ldots, x_D)$ be a point in D-dimensional space, $\forall i \in 1, 2, \ldots, D$.Suppose that for the function $y = f(x_1, x_2, \ldots, x_D)$, where $y \in [y_{\min}, y_{\max}]$. The type-II opposite of X is defined by $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_D)$ as follow:

$$\tilde{x}_{II} = \{x \,|\tilde{y} = y_{\max} + y_{\min} - y\} \tag{6}$$

Example samples of Type-I and Type-II opposite points are shown in Fig. 1.

Because Type-II opposition needs to clarify the evaluation value of the target space in advance, it is very difficult to calculate Type-II opposition when solving black-box problems [50]. At present, most research are carried out for Type-I opposition. Considering the distance between the opposite solution and the current solution, two types of opposition are extended in Type-I opposition: Type-I Super-Opposition and Type-I Quasi-Opposition [60].

Definition 3 (*Type-I super-opposition*) [60] Let C be an n-dimensional set of concepts. Then for the distance function $d(\cdot, \cdot)$, when $d\left(\tilde{C}_q, C\right) < d\left(\tilde{C}, C\right)$, all points $\tilde{C}_q$ are Type-I Super-Opposite of C.

Definition 4 (*Type-I quasi-opposition*) [60] Let C be an n-dimensional set of concepts. Then for the distance function

$d(\cdot,\cdot)$, when $d\left(\tilde{C}_{\mathrm{q}}, C\right) > d\left(\tilde{C}, C\right)$, all points $\tilde{C}_{\mathrm{q}}$ are Type-I Quasi-Opposite of C.

Seif et al. [61] propose the following extended opposition for Type-I Super-Opposition.

Definition 5 (*Reflected extended opposition*) [61] Given $X = (x_1, x_2, \ldots, x_{\mathrm{D}})$ be a point in D-dimensional space, where $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, \ldots, D\}$. The Reflected extended point of X is defined by $\tilde{X}^{\mathrm{reo}} = (\tilde{x}_1^{\mathrm{reo}}, \tilde{x}_2^{\mathrm{reo}}, \ldots, \tilde{x}_D^{\mathrm{reo}})$ as follow:

$$\tilde{x}_i^{\mathrm{reo}} = \begin{cases} \mathrm{rand}\left(x_i, b_i\right), & x_i > \left(a_i + b_i\right)/2 \\ \mathrm{rand}\left(a_i, x_i\right), & x_i < \left(a_i + b_i\right)/2 \end{cases} \tag{7}$$

Definition 6 (*Extended opposition*) [55] Given $X = (x_1, x_2, \ldots, x_D)$ be a point in D-dimensional space, where $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, \ldots, D\}$. The extended point of X is defined by $\tilde{X}^{\mathrm{eo}} = \left(\tilde{x}_1^{\mathrm{eo}}, \tilde{x}_2^{\mathrm{eo}}, \ldots, \tilde{x}_{\mathrm{D}}^{\mathrm{eo}}\right)$ as follow:

$$\tilde{x}_i^{\mathrm{eo}} = \begin{cases} \mathrm{rand}\left(\tilde{x}_i, b_i\right), & x_i < \left(a_i + b_i\right)/2 \\ \mathrm{rand}\left(a_i, \tilde{x}_i\right), & x_i > \left(a_i + b_i\right)/2 \end{cases} \tag{8}$$

Based on the above two definitions, it can be found that reflected extended opposition randomly generates each initial solution between the current solution and the closest boundary. Whereas extended opposition randomly generates each solution between the solution's opposing solution and the farther bound.

Similarly, Type-I Quasi-Opposition can also be divided into two categories: reflection and opposition.

Definition 7 (*Quasi-reflection opposition*) [62] Given $X = (x_1, x_2, \ldots, x_{\mathrm{D}})$ be a point in D-dimensional space, where $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, \ldots, D\}$. The Quasi-reflection point of X is defined by $\tilde{X}^{\mathrm{qr}} = \left(\tilde{x}_1^{\mathrm{qr}}, \tilde{x}_2^{\mathrm{qr}}, \ldots, \tilde{x}_D^{\mathrm{qr}}\right)$ where $\tilde{x}_i^{\mathrm{qr}} = \mathrm{rand}\left(x_i, c_i\right)$ and $c_i = \left(a_i + b_i\right)/2$.

Definition 8 (*Quasi opposition*) [63] Given $X = (x_1, x_2, \ldots, x_D)$ be a point in D-dimensional space, where $x_i \in [a_i, b_i]$, $\forall i \in \{1, 2, \ldots, D\}$. The Quasi point of X is defined by $\tilde{X}^{\mathrm{qo}} = \left(\tilde{x}_1^{\mathrm{qo}}, \tilde{x}_2^{\mathrm{qo}}, \ldots, \tilde{x}_D^{\mathrm{qo}}\right)$ where $\tilde{x}_i^{\mathrm{qo}} = \mathrm{rand}(c_i, \tilde{x}_i)$ and $c_i = (a_i + b_i)/2$.

Based on Definitions 5, 6, 7, and 8, it can be found that quasi-reflected opposition extends the range of opposite-based learning between the value center and the current solution, and quasi-opposition moves each solution between the solution's opposite and the value center. Figure 2 shows the range of opposition points for Quasi opposition, Extended opposition, and Reflected extended opposition.

Currently, the vast majority of literature is on quasi opposition variants. In [53], Wang et al. proposed the definition

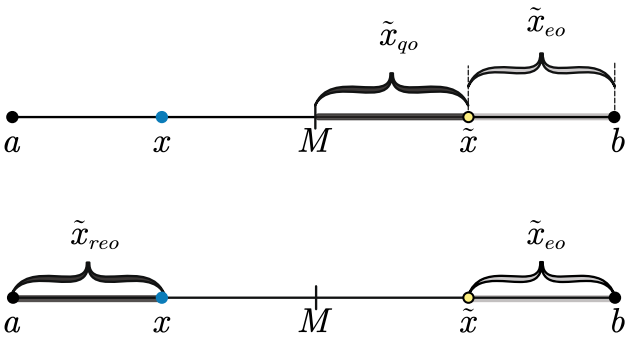


Fig. 2 Opposition range of quasi opposition, extended opposition and reflected extended opposition

of generalized opposite-based learning (GOBL):

$$\tilde{x}_i^{\mathrm{go}} = k \cdot \left(a_i + b_i\right) - x_i \tag{9}$$

where k is a random number in the range[0,1].

Although GOBL can alleviate the problem of premature convergence of the algorithm to a certain extent, it will also increase the probability of falling into the local optimum. To this end, [64] employs two adaptive strategies as supplements to improve the performance of GOPSO.

The opposition based on the current optimal solution information was proposed by Xu et al. [65] The definition of COOBL is as follows:

$$\tilde{x}_i^{\mathrm{coo}} = 2x_i^{\mathrm{best}} - x_i \tag{10}$$

where x_i^{best} is the optimal solution for the current population.

Rahnamayan et al. [66] proposed centroid opposition computing, which uses the center of gravity of the entire group as a reference point to calculate the reverse point, so that the reverse point includes the group search experience. Let the center of gravity of $X = (x_1, x_2, \ldots, x_D)$ be C, then the definition of the centroid opposition points $\tilde{X}^{\mathrm{co}} = \left(\tilde{x}_1^{\mathrm{co}}, \tilde{x}_2^{\mathrm{co}}, \ldots, \tilde{x}_D^{\mathrm{co}}\right)$ is as follows:

$$\tilde{x}_i^{\mathrm{co}} = 2C - x_i \tag{11}$$

Tables 1 and 2 show the combination of the PSO algorithm and OBL, and give the most influential OBL-related variants in recent years. For the algorithm, the common purpose of adding the OBL strategy is to speed up the convergence speed of the population. The OBL variant is mainly used in the initialization phase of the population and the iterative update phase based on the hopping rate.

Population diversity analysis

In the swarm intelligence algorithm, population diversity is used to describe the distribution of a single particle in the

Table 1 Summary of OBL variants

Variants	Type
OBL [49]	Type-I Quai-Opposition
QOBL [63]	Type-I Quai-Opposition
QROBL [62]	Type-I Reflected
EOBL [55]	Type-I Super-Opposition
REOBL [61]	Type-I Reflected
GOBL [53]	Type-I Quai-Opposition
COBL [66]	Type-I Quai-Opposition
COOBL [65]	Type-I Quai-Opposition
NOBL [67]	Type-I Quai-Opposition
DOBL [57]	Type-I Quai-Opposition

Table 2 Combination of PSO basic algorithm and OBL

	Use OBL ideas	Pbest update equation [55]	
		Inertia weight [51]	
Combination of PSO basic algorithm and OBL	OBL variants and improvements	OPSO [52]	Combination of OBL and PSO
		QOPSO [69]	Combination of QOBL and PSO
		GOPSO [53]	Combination of GOBL and PSO
		AMOPSO [64]	Introducing Adaptive Strategies to GOBL
		OpbestPSO [68]	Adversarial measures against the best individual
		EOPSO [55]	EOBL combined with PSO

search area, which is an important factor affecting the global performance of the algorithm. When the PSO algorithm is searching, it is usually accompanied by the lack of population diversity, which causes the algorithm to fall into local optimum and premature [70]. Exploration and exploitation are two important stages of population evolution. When in the exploration stage, the population will explore a wider search area, and the positions of the particles will be relatively scattered; during the exploitation, the population will perform a fine search for the optimal solution area, and the particle positions will be relatively concentrated. To show how different OBL variants differ in population diversity, we introduce a diversity measure [71]. In a D-dimensional search space, the population diversity defined based on the mean is as follows:

$$\mathrm{Div}_j = \frac{1}{N}\sum_{i=1}^{N}\left|x_{ij} - \bar{x}_j\right| \tag{12}$$

$$\mathrm{Div} = \frac{1}{D}\sum_{j=1}^{D}\mathrm{Div}_j \tag{13}$$

where N is the population size, x_{ij} is the value of the jth variable of the ith particle, and $\bar{x}_j$ represents the average value of the jth dimension of the population, defined as follows:

$$\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} \tag{14}$$

In order to show the variation trend of diversity of different OBL variants in the process of population iteration, the multimodal function Griewank is selected as the experimental function [72]:

$$f(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{15}$$

In the experiment, set the common parameters population size N = 100, dimension D = 30, and maximum function evaluation times to 100,000. After 25 independent runs of each algorithm, the curve of the mean population diversity with the number of iterations is shown in Fig. 3.

It can be clearly seen from the figure that as the number of iterations increases, EOBL can maintain a higher population diversity than other OBL variants, which means a wider range of search for optimal values, effectively avoiding premature trapping local optimum. However, the local search ability of particles is insufficient, and the convergence accuracy cannot be guaranteed. COBL relies on the advantage of

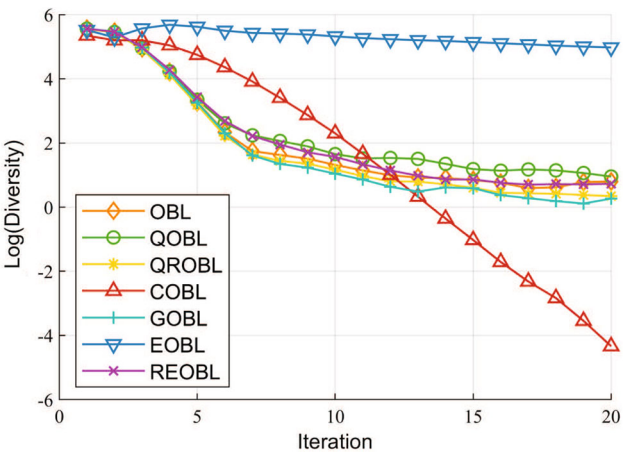


Fig. 3 Diversity trends of various OBL variants

centroid-based opposition to converge rapidly compared to other OBL variants. Premature convergence will lead to the inability to jump out of the local optimum, and the lack of searchability in the later stage is an obvious defect of COBL. To this end, how integrating the advantages of the above two basic variants and balancing the ability of population exploration and development is one of the research focuses of this paper.

Proposed algorithm

Double mean center

In the original version of OBL, the selection of opposite points depended on the center point of each dimension boundary. But compared to the centroid point of the overall population, the center point is obviously not an appropriate choice. Therefore, Rahnamayan proposed in 2014 to replace the center point with the centroid point to calculate the corresponding opposite point [66]. In the D-dimensional search space, suppose there are N particles $X = (x_1, x_2, \ldots, x_D)$, and each particle in this space has a mass attribute. Then the mass center point of the particle population, that is, the Mean Center (MC) of the population can be expressed by the following formula:

$$x_d^{\mathrm{MC}} = \frac{\Sigma_{i=1}^{N} x_{i,d}}{N} \tag{16}$$

Among them, $i = 1, 2, \ldots, N$, N is the number of particles, $d = 1, 2, \ldots, D$, in the formula represents the d-th dimension value of the mean center, that is, the particle's centroid value in the d-th dimension.

In physics, the center of mass is defined as the imaginary point of concentration of an object of uniform density and is used to characterize the average location of the mass of the object. However, the solution space where the particles are located is not always strictly uniform distribution, but there is a certain degree of offset [73], so it is not rigorous to describe the properties of the entire population with the centroid point. In the calculation of the center of mass, only the influence of the mass attribute is considered, but the density attribute of the particle is ignored. The opposite points obtained through the centroid points cannot reflect the distribution of data points in the neighborhood. Therefore, this paper takes the mean density of all particles as the benchmark and selects particles higher than the mean density to form a representative density subgroup. The particle density is defined by calculating the number of data points contained in the decentered neighborhood, the Euclidean distance between samples, and the corresponding fitness of samples. Assuming that $X = (x_1, x_2, \ldots, x_D)$ is N points in the D-dimensional search space, the average distance between particle points and particle points in the neighborhood is:

$$D(x_i) = \frac{1}{k} \sum_{x_j \in N_k} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{17}$$

Among them, $N_k = \left\{ x_j \,|0 < \left|x_i - x_j\right| < \varepsilon \right\}$ is the neighborhood of x_i, ε is the radius of the neighborhood, and k is the number of particles in the neighborhood of particle x_i.

Let $x^k = \{x_i | \text{The number of neighbors of } x_i \text{ is } k\}$, $D_{\max}^k = \max_{x_i \in x^k} D(x_i)$, $D_{\min}^k = \min_{x_i \in x^k} D(x_i)$, introduce a linear normalization function to reduce the computational complexity, then the density of particle x_i is:

$$\rho_i = \begin{cases} k + \frac{D_{\max}^k - D(x_i)}{D_{\max}^k - D_{\min}^k} & k \neq 0 \\ 0 & k = 0 \end{cases} \tag{18}$$

Select particles that are better than the overall density mean to form a density subgroup, and the mean of the density subgroup is the Density Center (DC) of the population, which can be expressed by the following formula:

$$x_d^{\mathrm{DC}} = \frac{\Sigma_{i=1}^{n} x_{i,d}}{n} \tag{19}$$

Among them, n is the number of particles that are better than the mean density, and the formula represents the d-th dimension value of the density center, that is, the mean value of all the particles that are better than the density means in the d-th dimension.

In order to comprehensively consider the two factors of the mean center and the density center, this paper combines the advantages of the two to define a new center called the double mean center(DMC), which can be expressed by the following formula:

$$x_d^{\mathrm{DMC}} = \left(x_d^{\mathrm{DC}} + x_d^{\mathrm{MC}}\right)/2 \tag{20}$$

where represents the d-th dimension value of DMC.

Judging the range of parameters according to the characteristics of data distribution can effectively determine the parameters in a relatively reasonable range. Kernel density estimation can characterize the data distribution [74] and is a nonparametric method. Assuming that the independent distribution F contains x_1,x_2,…,x_n sample points, the probability density function is f, and the kernel density is estimated as follows:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{21}$$

Among them, h is the bandwidth, $K(x)$ is the kernel function, and n is the number of samples.

In the kernel density estimation theory, the choice of the bandwidth value has a great influence on the estimator [75]. If h is too small, the density estimation is limited to the nearby area of the observed data, resulting in too few observed samples in the neighborhood. If h is too large, then the density estimation divides the probability densities too scattered, which filters out some important properties. According to the above analysis, if the size of h can be determined mathematically, the value of the parameter neighborhood radius ε can be determined. In statistics, the integral mean square error $MISE(h)$ is usually used as the criterion for judging the quality of the density estimator [75].

$$\text{MISE}\,(h) = \text{AMISE}\,(h) + O\left(\frac{1}{nh} + h^4\right) \tag{22}$$

In,

$$\text{AMISE}\,(h) = \frac{\int K^2(x)\,\mathrm{d}x}{nh} + \frac{h^4\sigma^4 \int \left[f''(x)\right]^2 \mathrm{d}x}{4} \tag{23}$$

Minimizing MISE (h) is equivalent to minimizing AMISE (h). We must set the bandwidth h at some intermediate value to avoid excessively biased kernel estimates. First, take the partial derivative of AMISE (h) and set the derivative equal to 0:

$$\begin{aligned}&\frac{\partial}{\partial h}\text{AMISE}\,(h)\\ &= -\frac{\int K(x)^2\,\mathrm{d}x}{nh^2} + \left(\int x^2 K(x)\,\mathrm{d}x\right)^2 h^3 \int \left[f''(x)\right]^2 \mathrm{d}x\end{aligned} \tag{24}$$

Then the optimal bandwidth value is:

$$h = \left(\frac{\int K^2(x)\,\mathrm{d}x}{n\sigma^4 \int [f''(x)]^2\,\mathrm{d}x}\right)^{\frac{1}{5}} \tag{25}$$

For the unknown quantity $\int \left[f''(x)\right]^2 \mathrm{d}x$ in the optimal bandwidth, Sliverman proposed a rule of thumb [76], replacing f with a normal density whose variance matches the estimated variance. This is equivalent to estimating $\int \left[f''(x)\right]^2 \mathrm{d}x$ using $\int \left[\phi''(x)\right]^2 \mathrm{d}x/\hat{\sigma}$, where ϕ is the standard normal density function. If the kernel function is the Gaussian density kernel function, ϕ uses the sample variance $\hat{\sigma}$, and uses the rule of thumb to obtain the adaptive neighborhood radius:

$$\varepsilon = h = \left(\frac{4}{3}n\right)^{\frac{1}{5}} \hat{\sigma} \tag{26}$$

Fig. 4 shows the particle distributions with reference to the convergence of different centers in the iterative process of the population. Four algorithms were selected and compared at 1, 10, and 30 iterations. Figure 4a–c are the particle distributions of the standard PSO algorithm, and Fig. 4d–f are the particle distributions of the standard PSO algorithm. The particle distribution of the OPSO algorithm, Fig. 4g–i are the particle distribution of the OPSO algorithm with the center of mass instead of the center, Fig. 4j–l are replaced by the double mean center. The particle distribution of the center's OPSO algorithm. Overall, the four algorithms are randomly dispersed in the solution space in the early stage of evolution. With the increase in the number of iterations, OBL uses its own characteristics to guide the particles to gather the global optimal solution, which is obviously different from the ordinary PSO algorithm. In addition, the OPSO algorithm that replaces the center with MC and the OPSO algorithm that replaces the center with DMC can converge to the global optimal solution faster than the PSO algorithm.

In order to reflect the advantages of DMC, the test is carried out on 12 test functions of CEC2022. The population size is set to $N = 10$, the dimension $D = 20$, the maximum number of function evaluations is 5000, each test function is run independently 30 times, and the final result is taken as 30. The average value of the times, the experimental results obtained are shown in Table 3. Wilcoxon rank sum test and Friedman test were performed on the data in the table, and the rank means of DMC was the smallest, which verified that DMC has more advantages than MC. The specific DMC strategy is shown in Algorithm 1.

Algorithm 1: DMC strategy

```
Calculate the adaptive neighborhood radius Nbr using equation (26);
Calculate the mean center(MC);
for i = 1 to popsize do
    for j = 1 to popsize do
        if |x_i − x_j| < Nbr then
            Add x_j to the set of decentered neighborhoods of x_i
            and calculate the distance according to equation (17);
Calculate the density of each particle according to equation (18)
and the density mean T;
for i = 1 to popsize do
    if p_i > T then
        The density of the current x_i is greater than the threshold,
        put the point into the density subgroup;
Calculate the density center(DC) according to equation (19);
Calculate the dual mean center(DMC) according to equation (20);
final ;
return ;
```

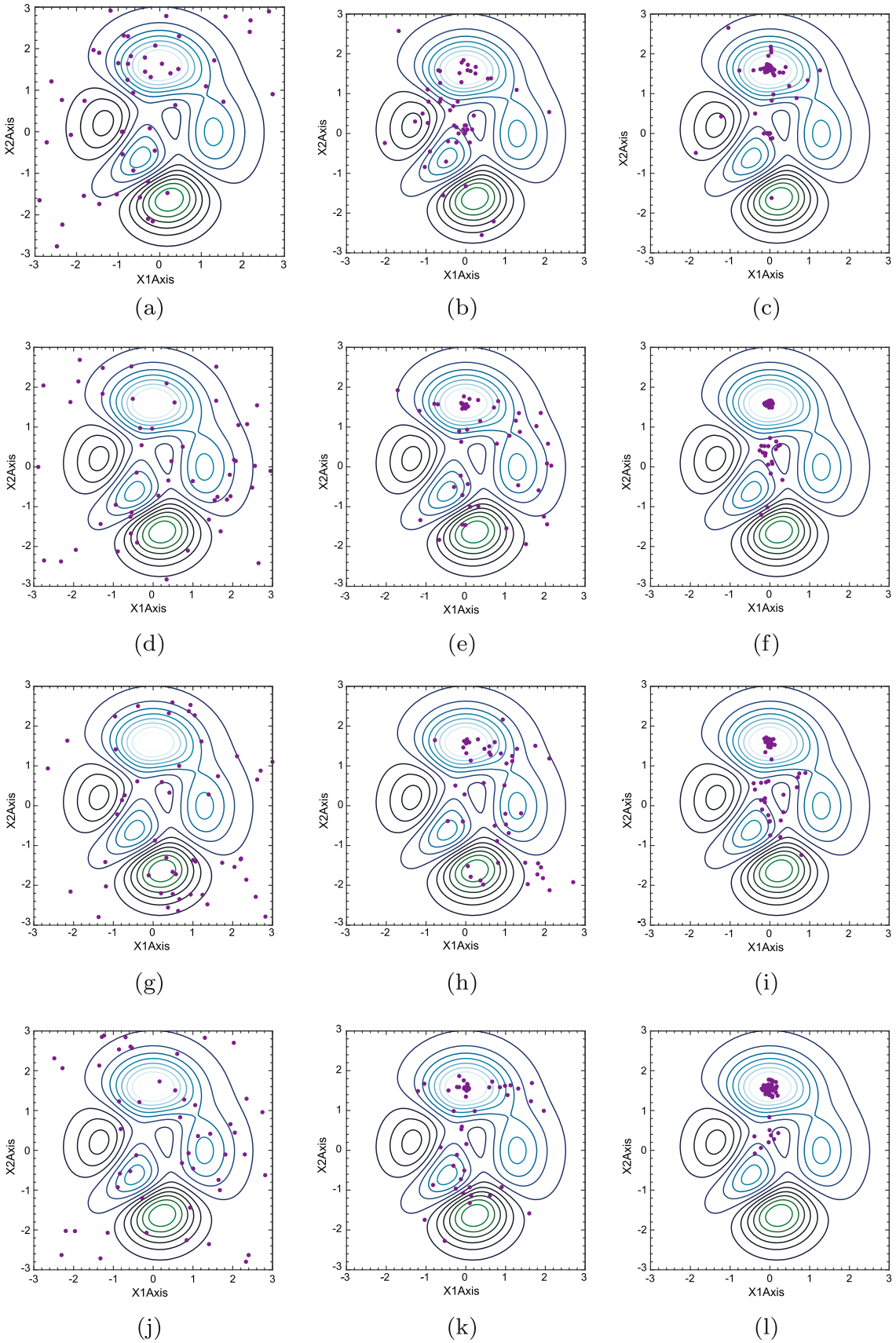


Fig. 4 Particle distribution of PSO, OPSO, OPSO using MC and OPSO using DMC after 1, 10 and 30 iterations

Table 3 Results of OPSO algorithm with different reference centers at CEC2022

	OPSO	MC	DMC
F1	1.78E+04 (−)	3.04E+04 (−)	**1.14E+04**
F2	6.89E+02 (−)	6.58E+02 (−)	**6.27E+02**
F3	6.35E+02 (−)	6.42E+02 (−)	**6.20E+02**
F4	8.58E+02 (−)	8.65E+02 (−)	**8.46E+02**
F5	1.60E+03 (−)	2.02E+03 (−)	**1.32E+03**
F6	1.03E+08 (−)	6.08E+06 (−)	**2.09E+07**
F7	**2.06E+03 (=)**	2.08E+03 (−)	2.06E+03
F8	**2.24E+03 (+)**	2.24E+03 (+)	2.25E+03
F9	2.62E+03 (−)	2.60E+03 (−)	**2.57E+03**
F10	2.66E+03 (−)	3.01E+03 (−)	**2.65E+03**
F11	**3.01E+03 (+)**	3.26E+03 (−)	3.03E+03
F12	2.89E+03 (=)	**2.88E+03 (+)**	2.89E+03
-/=/+	8/2/2	10/0/2	–
Ranks	2.17	2.33	**1.50**

Refraction imaging quasi-opposition

Refraction of light is a common physical phenomenon in nature. Snell gave the definition of light refraction, called Snell's Law [77]. When light is refracted into another medium from a vacuum, the ratio n of the incident angle α to the sine of the refraction angle β is called the absolute refractive index of the medium, which can be expressed by the following formula:

$$n = \frac{\sin\alpha}{\sin\beta} \tag{27}$$

The process of OBL strategy is improved by introducing the principle of refraction, as shown in Fig. 5, which is a schematic diagram of the Quasi opposition process based on the principle of refraction.

The discrete solution space is divided into two along the x-axis, the upper half of the space is in a vacuum state, and the lower half can be regarded as other medium space (such as water). Assuming that the value range of particle x on the x-axis is $[a, b]$, point O is the midpoint of the value range of x. Make the normal of the x-axis along the point O, which is the y-axis. In order to apply the principle of refraction to the opposition learning process, the following assumption is made: there is a beam of light source x' (called the incident point) directly above the particle x, and a beam of incident ray Ox' is emitted to the midpoint O, and the distance of Ox' is h. The incident light is refracted at the midpoint O, the refracted ray is $x^{*'}O$, $x^{*'}$ is the refraction point, and the distance of $x^{*'}O$ is h^*. It is not difficult to obtain the following calculation formula from Fig. 5:

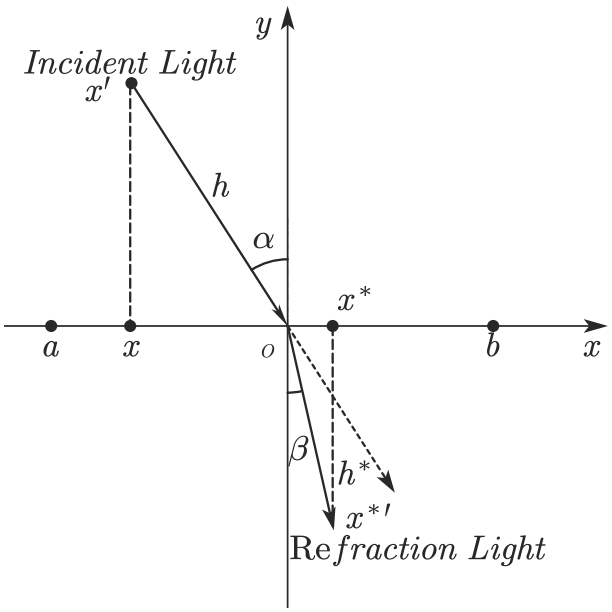


Fig. 5 Schematic diagram of quasi opposition process based on the principle of refraction

$$\frac{\sin\alpha}{\sin\beta} = \frac{((a+b)/2 - x)\cdot h^*}{(x^* - (a+b)/2)\cdot h} = n \tag{28}$$

Set $k = h/h^*$, then Eq. (28) can be rewritten as:

$$x^* = (a+b)/2 + (a+b)/(2kn) - x/kn \tag{29}$$

Extending Eq. (29) to multidimensional space, it can be rewritten as:

$$x^*_{i,j} = \left(a_j + b_j\right)/2 + \left(a_j + b_j\right)/(2kn) - x_{i,j}/kn \tag{30}$$

where a_j and b_j are the maximum and minimum values of the jth dimension in the current population, $x_{i,j}$ is the jth dimension value of the ith particle in the population, and $x^*_{i,j}$ is the refraction solution of $x_{i,j}$.

Replacing the midpoint O with the DMC mentioned in the previous section, Eq. (29) can be rewritten as:

$$x^* = \mathrm{DMC}/2 + \mathrm{DMC}/kn - x/kn \tag{31}$$

Extending Eq. (31) to multidimensional space, it can be rewritten as:

$$x^*_{i,j} = \mathrm{DMC}^t_j/2 + \mathrm{DMC}^t_j/kn - x_{i,j}/kn \tag{32}$$

where DMC^t_j represents the double mean center of the j-th dimension of the population at the t-th iteration, $x_{i,j}$ is the j-th dimension value of the i-th particle in the population, and $x*_{i,j}$ is the refraction solution of $x_{i,j}$.

The refractive index n is 4/3 in water, and the value of k is linearly decreasing:

$$k = k_{\max} - (k_{\max} - k_{\min})\, t/t_{\max} \tag{33}$$

Among them, $k_{\max}$ and $k_{\min}$ are the maximum and minimum values of k, respectively, t is the current number of iterations, and $t_{\max}$ is the maximum number of iterations.

Therefore, in this paper, the opposition learning strategy of refraction imaging is combined with the double mean center, and the Quasi opposition learning strategy of refraction imaging based on DMC is obtained.

Extended opposition based on the principle of refraction

Macroscopically, light reflected from an object contains innumerable photons, which are radiated in all directions due to diffuse reflection. The virtual image generated by the object through refraction is actually a collection of image points corresponding to each point on the object. What people see is like the intersection of the reverse extension lines of the light source emitted by the object and entering the human eye after being refracted, and the position of the virtual image can be determined by selecting any two of the many rays. As shown in Fig. 6, two rays are arbitrarily selected, the slopes of the straight lines l_0 and l_1 are set as k_0 and k_1 respectively, the coordinates of the intersection of the straight lines are (x, y), and the intercepts between the two straight lines and the coordinate axes are $x_0^{'}$, $x_1^{'}$, $y_0^{'}$, $y_1^{'}$, then the straight line equation of l_0 and l_1 can be expressed as:

$$y = k_0\left(x - x_0^{'}\right) \tag{34}$$

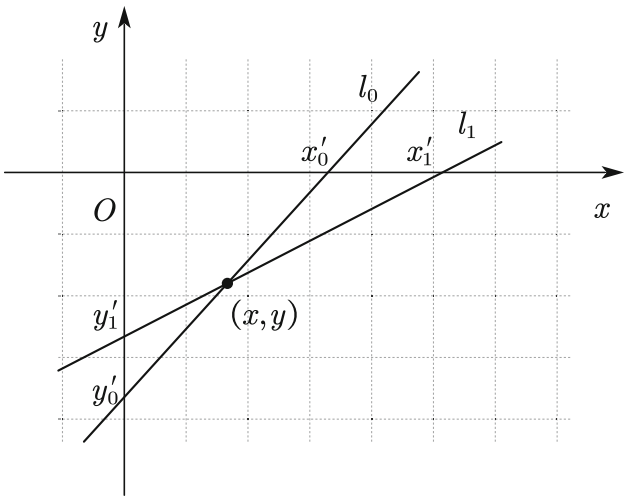


Fig. 6 Intersection and intercept of two refracted rays

$$y = k_1\left(x - x_1^{'}\right) \tag{35}$$

Substitute $y_0^{'} = -k_0 x_0^{'}$ and $y_1^{'} = -k_1 x_1^{'}$ into Eqs. (34) and (35) to obtain the coordinates of the intersection:

$$x = -\frac{y_1^{'} - y_0^{'}}{k_1 - k_0} \tag{36}$$

$$y = -\frac{x_1^{'} - x_0^{'}}{1/k_1 - 1/k_0} \tag{37}$$

If the intercept of a straight line changes continuously with the slope, then the numerator and denominator in Eqs. (36) and (37) are changed to differential equations, and the formula for the coordinates of the intersection of two approximately parallel straight lines can be obtained:

$$\begin{cases} x = -\frac{\mathrm{d}y'}{\mathrm{d}k} \\ y = -\frac{\mathrm{d}x'}{\mathrm{d}(1/k)} \end{cases} \tag{38}$$

It can be seen from Eq. (38) that the intersection coordinates of two approximately parallel straight lines can be obtained by derivation of the intercept of the straight line against the slope or the reciprocal of the slope.

When people look down at the chopsticks that are obliquely inserted in the bowl full of water, they will see that the chopsticks are deflected upwards. This is actually because the virtual image produced by the chopsticks is deflected upwards relative to the chopsticks themselves. As shown in Fig. 7, it is the learning process of Extended opposition based on the principle of refraction.

Insert the chopsticks AA' into the water diagonally in half, the angle θ between the chopsticks and the water surface, point O is both the midpoint of the chopsticks and the origin of the coordinate axis. Assume that the abscissa corresponding to endpoint A is particle x, and the search space of x is $[a, b]$. The underwater object point $A'$is at a depth h from the water surface (x-axis), and an incident light beam is emitted from the object point A', and refraction occurs at the intersection point M of the water surface. The virtual image point A^* is obtained through the reverse extension line of the refracted light through point M, the incident angle is i, and the refraction angle is r. The projection of the virtual image point A^* on the x-axis is x^*, and x^* is called the extended opposition solution of particle x based on the principle of refraction.

The midpoint O of Extended opposition learning based on refraction imaging, like OBL, is the midpoint $(a+b)/2$ of the particle x search space. It can be seen from Fig. 7 that $x' = h\tan i + a + b - x$, y' = -(x'+x-a-b) $\cot r = -h\tan i \cdot \cot r$, $h = \tan\theta \cdot |x|$ are the slope $k = \cot r$ of the refracted ray l. The position coordinates of the virtual image point in the direction of viewing angle r can be obtained by

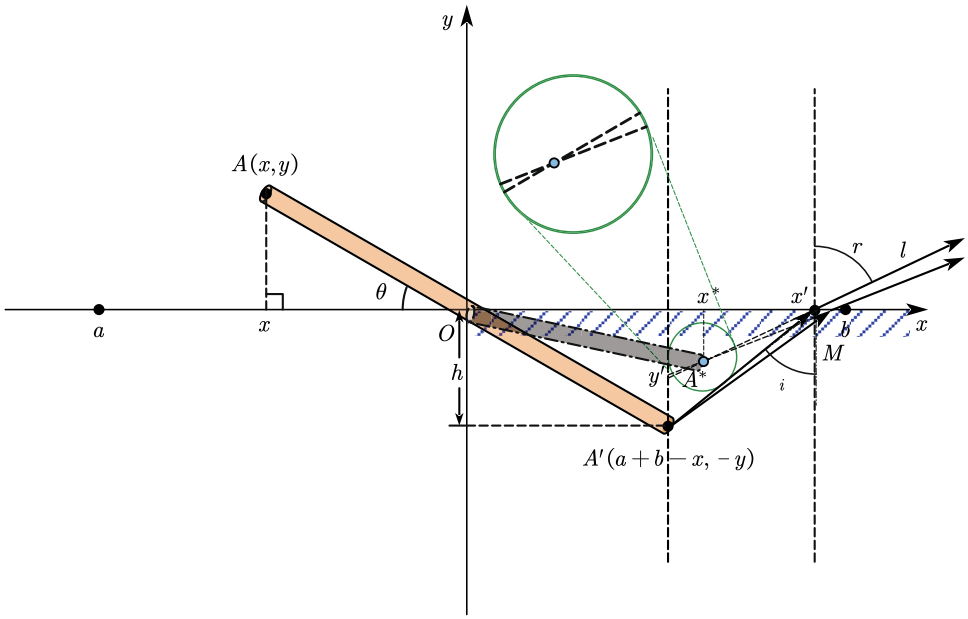


Fig. 7 Extended opposition learning process based on the principle of refraction

substituting the above parameters into Eq. (38). First, using $n = \sin r / \sin i$ of Snell's Law (when n is the refractive index of water, $n = 4/3$), $\tan i$, x', and y' are expressed as:

$$\tan i = \frac{1}{\sqrt{n^2k^2 + \left(n^2 - 1\right)}} \tag{39}$$

$$x' = \frac{\tan\theta \cdot |x|}{\sqrt{n^2k^2 + \left(n^2 - 1\right)}} + a + b - x \tag{40}$$

$$y' = -\tan\theta \cdot |x| \cdot \frac{k}{\sqrt{n^2k^2 + \left(n^2 - 1\right)}} \tag{41}$$

Substituting the above formula into the previously obtained intersection coordinate Eq. (38), we can get:

$$x^* = a + b - h \cdot \left(n^2 - 1\right) \cdot \tan^3 i - x \tag{42}$$

$$y^* = -\frac{h}{n}\left[1 - \left(n^2 - 1\right) \cdot \tan^2 i\right]^{3/2} \tag{43}$$

It can be seen from Eqs. (42) and (43) that as i (or viewing angle r) increases, the coordinates x and y increase synchronously, which means that the virtual image point a moves upward to the right as the viewing angle r increases, until it moves to the water surface ($y = 0$), at this time $\tan^2 i = 1/\left(n^2 - 1\right)$, namely $\sin i = 1/n$, is obviously the critical position where total reflection occurs. At this time, the coordinate of the virtual image point A^* is $x_{\max} = a + b - x + h \cdot \sqrt{1/\left(n^2 - 1\right)}$, which is also the maximum value of the opposition solution x^*. When the viewing angle is $r \to 0$, the coordinate of the virtual image point is $(a + b - x, -h/n)$, which is directly above the object point A', and the depth is about 75% of the depth of the object point A'. At this time, the opposition solution corresponding to the virtual image point A' is actually the OBL opposition solution. In order to see the change trajectory of the virtual image point more clearly, with $a + b - x$ as the origin, the trajectory equation can be obtained by eliminating $\tan i$ from Eqs. (42) and (43) simultaneously:

$$y = -\frac{\left[h^{\frac{2}{3}} - \left(n^2 - 1\right)^{\frac{1}{3}} x^{\frac{2}{3}}\right]^{\frac{3}{2}}}{n} \tag{44}$$

The specific trajectory is shown in Fig. 8.

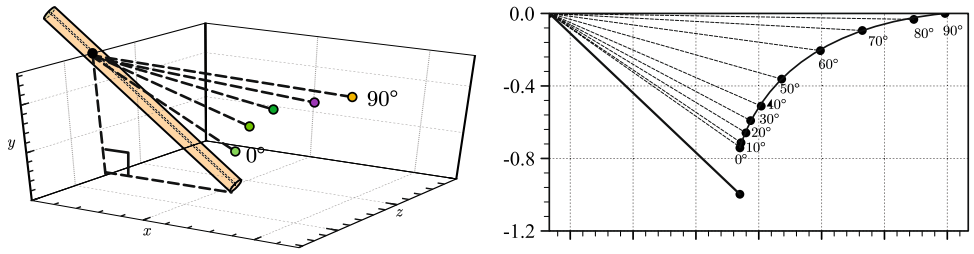
Extending Eq. (42) to multidimensional space, it can be rewritten as:

$$x^*_{i,j} = a_j + b_j - x_{i,j} \cdot (\tan\theta \cdot (n^2 - 1) \cdot \tan^3 i + 1) \tag{45}$$

Among them, the angle $\theta = 30$ between the chopsticks and the water surface, the incident angle is i, the refractive index in the water is $n = 4/3$, $x_{i,j}$ is the j-th dimension value of the i-th particle in the population, and $x^*_{i,j}$ is the refraction solution of $x_{i,j}$. The value of the incident angle i is linearly decreasing in the range $[0°, 90°)$:

$$i = i_{\max} - (i_{\max} - i_{\min})\, t / t_{\max} \tag{46}$$

Among them, $i_{\max}$ and $i_{\min}$ are the maximum and minimum values of i, respectively, t is the current number of iterations, and $i_{\max}$ is the maximum number of iterations.



(a) 1 (b) 2

Fig. 8 The change trajectory of the virtual image point position with the viewing angle r when h = 1

Integration of opposites

According to the global search characteristics of the PSO algorithm, with the iterative operation of the algorithm, the particles gradually approach each other, and the diversity of the population decreases. At this time, the global exploration ability of the algorithm is weakened, and it is easy to fall into the local optimum. However, it is not enough to only enhance the global exploration ability. Neglecting the local development ability will cause the particles to be in the search stage all the time, unable to converge to the optimal value. Therefore, the degree of exploration and development needs to be balanced. Different evolutionary stages require different diversity and accurate control of population diversity helps population evolution. Given the above analysis, in order to prevent local convergence or premature maturity, the percentage of exploration and exploitation in each iteration of the algorithm needs to be determined according to the diversity metric in the Preliminaries section [78]:

$$Xpl\% = \frac{\text{Div}}{\text{Div}_{\max}} \times 100 \tag{47}$$

$$Xpt\% = \frac{|\text{Div} - \text{Div}_{\max}|}{\text{Div}_{\max}} \times 100 \tag{48}$$

where Div is the diversity of the population in the current iteration and $\text{Div}_{\max}$ is the maximum diversity of the population in all iterations. $Xpl\%$ and $Xpt\%$ are iterative exploration and development percentages, respectively, and these two elements are conflicting and complementary.

In order to better balance the global exploration and local development capabilities of the population, a diversity-driven fusion-opposition strategy (SQOBL) is designed in this paper. Diversity is used to drive the mutual adaptive conversion of the two opposing strategies. When Xpl% is greater than the set threshold, the refraction principle based on the double mean center is used to quasi-opposition, which not only accelerates the algorithm convergence but also performs local fine search. However, when the population diversity gradually decreases, the particles are also prone to fall into the local optimum, that is, the case where Xpl% is lower than the threshold. At this time, the principle of refraction is used to expand the opposition, increase the diversity of the population, help the particles to jump out of the local optimum, and prevent premature maturity. Once Xpl% is above the threshold, the extended opposition learning strategy stops, switches to the quasi-opposition learning strategy again, and repeats this step until the maximum number of iterations is reached.

Algorithm 2: SQOPSO

```
Random initialization of population X and velocity V, population size N;
for i = 1 to N do
    Calculate opposite of initial population OX_i according to equation (30);
    Calculate the fitness of population f(X_i);
    Calculate the fitness of opposite population f(OX_i);
    if f(X_i) < f(OX_i) then
        The opposite solution OX_i replaces X_i to join the population in the iteration ;
while FE ≤ FE_Max do
    Update the incidence angle i and contraction factor k;
    Update Double Mean Center(DMC);
    for i = 1 to N do
        Update velocity V and position Xnew according to equations (1) and (2); if XPL < T then
            Calculate extended opposite population OX_i according to equation (45);
        else
            Calculate quasi opposite population OX_i according to equation (32);
        Calculate the fitness of population f(X_i);
        Calculate the fitness of opposite population f(OX_i);
        if f(X_i) < f(OX_i) then
            The opposite solution OX_i replaces X_i to join the population in the iteration;
    Update the personal best and the global best;
final ;
return ;
```

On the basis of the above research, a combination of the SQOBL strategy and PSO algorithm is proposed. The specific flow chart of the algorithm is shown in Fig. 9, and its pseudo-code is shown in Algorithm 2. The code is available at https://github.com/ZooooZoooo/SQOBL. The algorithm flow of the combination of SQOBL and PSO is shown in Algorithm 2. It can be seen from Algorithm 2 that the

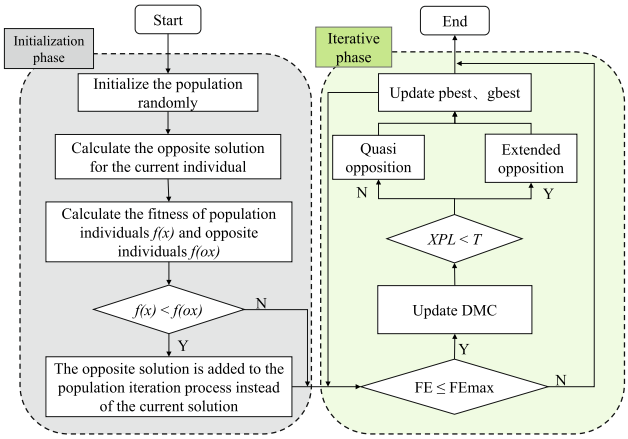


Fig. 9 Flow chart of SQOBL strategy

SQOPSO algorithm mainly includes four components: initialized population, initialized opposition population, DMC strategy, and fusion-opposition strategy based on diversity drive. Let the dimension of the objective function be D, the population size be N, and the maximum number of iterations be T. The initialization phase includes initializing the population and initializing the opposing population. The time complexity of the initializing population and the opposing population are both $O(ND)$, so the time complexity of the entire initialization phase is $O(2ND)$. In the iterative process, the time complexity of Algorithm 1 is $O(NN)$, and the time complexity of the fusion-opposition strategy based on diversity drive is $O(D)$, so the time complexity of the entire iterative process is $O(T(NN + ND))$, then the total time complexity of SQOPSO is $O(2ND + T(NN + ND))$, which is only related to T, N, and D. Because the problem dimension D is often larger than or similar to the number of groups N, the low-order terms are ignored, and the time complexity of the SQOPSO algorithm is $O(TND)$, which is consistent with the time complexity of the PSO algorithm.

SQOBL strategy analysis

In this subsection, the proposed SQOBL strategy is analyzed. First of all, the opposite solution with DMC as the reference point is closer to the optimal solution of the optimization problem than the current solution, and the proof derivation process refers to the literature [79].

Theorem 1 *When the optimal solution and the current solution are on different sides of the double mean center, the opposite solution x_i^* with DMC as the reference point is closer to the optimal solution x^{best} than x_i.*

Proof The population with the number of particles is N, and the center of the double mean is x^{DMC}. In the search space, there is a particle $x_i \in \{x_1, x_2, \ldots, x_N\}$, and $x_i \in [a, b]$ will opposite solve x_i^*:

$$x_i^* = 2 \cdot x^{\text{DMC}} - x_i \tag{49}$$

where $x^* \in \left[2x^{\text{DMC}} - b, 2x^{\text{DMC}} - a\right]$, it can be clearly seen from Eq. (48) that the center of the search space changes from $(a + b)/2$ to x^{DMC}, but the size of the space remains the same. When $x_i \neq x_i^*$, if x_i^* wants to be closer to the optimal solution x^{best} than the ordinary particle x_i, it needs to satisfy:

$$\left|x_i^* - x^{\text{best}}\right| < \left|x_i - x^{\text{best}}\right| \tag{50}$$

Then,

$$\begin{aligned} &\left(x_i^* - x^{\text{best}}\right)^2 - \left(x_i - x^{\text{best}}\right)^2 < 0 \\ &\Rightarrow \left(x_i^* + x_i - 2x^{\text{best}}\right)\left(x_i^* - x_i\right) < 0 \\ &\Rightarrow \left(x^{\text{DMC}} - x^{\text{best}}\right)\left(x^{\text{DMC}} - x_i\right) < 0 \end{aligned} \tag{51}$$

Obviously,

$$x_i < x^{\text{DMC}}, \quad if\ x^{\text{best}} > x^{\text{DMC}} \tag{52}$$

$$x_i > x^{DMC}. \quad if\ x^{\text{best}} < x^{\text{DMC}} \tag{53}$$

It can be seen that when the optimal solution x^{best} and the ordinary particle x_i are located on different sides of the double mean center x^{DMC}, the opposite solution x_i^* is closer to the optimal solution x^{best} than x_i. □

Then, under the given assumptions, briefly analyze the convergence of the extended oppositional learning strategy based on the refraction imaging principle:

Theorem 2 *If the PSO algorithm based on general opposite learning converges, then the SQOPSO algorithm is also convergent.*

Proof Let $x_{i,j}(t)$ and $x_{i,j}^*(t)$ be the values on the jth dimension of the current solution and opposite solution of the tth generation respectively, and the global optimal position of the function is x_{best}. It can be seen from the theorem that the PSO algorithm based on general opposite learning converges, and for the solution $x_{i,j}(t)$ in the t-th generation population, there is

$$\lim_{t\to\infty} x_{i,j}(t) = x_{\text{best},j} \tag{54}$$

Since $a_j(t) = \min\left(x_{i,j}(t)\right)$ and $b_j(t) = \max\left(x_{i,j}(t)\right)$, there is

$$\lim_{t\to\infty} a_j(t) = \lim_{t\to\infty} b_j(t) = x_{\text{best},j} \tag{55}$$

The reverse solution of the current solution generated by extended opposite learning based on the principle of refraction imaging is:

$$x_{\mathrm{i,j}}^{*} = a_j + b_j - x_{\mathrm{i,j}} \cdot \left(\tan\theta \cdot \left(n^2 - 1\right) \cdot \tan^3 i + 1\right) \quad (56)$$

When the algebra $t \to \infty$ in the above formula has

$$\begin{aligned} \lim_{t\to\infty} x_{i,j}^{*}(t) &= \lim_{t\to\infty} \left(a_j(t) + b_j(t)\right) \\ &\quad - \lim_{t\to\infty} \left(x_{i,j}(t) \cdot \left(\left(\frac{\sqrt{21}}{9}\right) \cdot \tan^3 i(t) + 1\right)\right) \\ &= 2x_{\mathrm{best},j} - x_{\mathrm{best},j} \\ &= x_{\mathrm{best},j} \end{aligned} \quad (57)$$

It can be seen from the results that when $x_{i,j}(t)$ converges to $x_{\mathrm{best},j}$, the extended opposite solution $x_{i,j}^{*}(t)$ established based on the principle of refraction imaging also converges to $x_{\mathrm{best},j}$. Therefore, the PSO algorithm based on general opposite learning converges, and the SQOPSO algorithm also converges. □

Experiments

Experimental setup

All experiments were performed in MATLAB R2022b and in a Windows 10 hardware environment with Intel Core i5-8500 with frequency 3.00 GHz and 8.0 GB of RAM. In the experiment, the population size $N = 100$, there are different problem dimension values on different benchmark function sets, the maximum number of function evaluations FE_max=100,000 and the iteration termination condition of each algorithm is that the current iteration number is equal to the maximum iteration number, and each algorithm is in 25 independent runs on each function.

To comprehensively and objectively evaluate the performance of SQOBL variants presented in the overall OBL variant framework, SQOBL was compared with 8 representative OBL variants. The OBL variants compared were: OBL [49], QOBL [63], QROBL [62], COOBL [65], COBL [66], GOBL [53], EOBL [55], and REOBL [61]. We embed all OBL variants into the PSO algorithm, such as QOPSO is to embed QOBL into PSO and QROPSO is to embed QROBL into PSO. In order to compare the differences between the algorithms fairly, the common parameters of each comparison algorithm are set uniformly, such as population size, maximum function evaluation times, etc.; other parameter settings are consistent with the original literature, and the specific parameter settings are shown in the appendix. The jump rate of OPSO, GOPSO, COOPSO, and COPSO is 0.3 [52, 53, 65, 66], and the jump rate of QOPSO, QRPSO, EOPSO, and REOPSO are 0.05 [55, 61–63].

In order to further observe the performance of the SQOPSO algorithm, we select 8 newly proposed intelligent optimization algorithms and test them on the latest benchmark function set CEC2022. The smart optimization algorithms compared are: Sparrow Search Algorithm (SSA) [5], Dwarf Mongoose Optimization (DMO) [6], Northern Goshawk Optimization (NGO) [7], Tuna swarm optimization (TSO) [8], Golden jackal optimization (GJO) [9], Sand Cat Swarm Optimization (SCSO) [10], Generalized normal distribution optimization (GNDO) [11] and PSO. The parameter settings of each comparison algorithm are consistent with the original literature, and the specific parameter settings are shown in the appendix. The early warning value of the SSA algorithm is 0.6, the proportion of discoverers is 0.7, and the proportion of aware of dangerous sparrows is 0.2. The number of babysitters in the DMO algorithm is 3, and the Alpha female vocalization is 2. In the TSO algorithm, it is determined that the degree of tuna following the optimal in the initial stage is 0.7. The constants c_1 and β of the GJO algorithm are both 1.5. The auditory characteristic of sand cats in the SCSO algorithm is 2.

In the SQOPSO algorithm, the refractive index n is 4/3, the angle of the chopsticks entering the water is 30°, the value of k is linearly decreasing, and the initial value is 0.75, which is near the middle value of the better interval [0, 1.4], the incident angle i is linearly decreasing in the interval [0°, 90°], and the threshold T of the fusion strategy is equal to 0.1 [58].

We choose different benchmark function sets for comparative experiments, they are divided into three kinds with different characteristics. The first is an unconstrained optimization problem, which selects 15 optimization functions in CEC2015 [80]; the second is a constrained optimization problem, which selects 29 optimization functions in CEC2017 [61]; the third is the latest single-objective optimization Question, 12 optimization functions in CEC2022 are selected. Record the mean (Mean) and standard deviation (SD) of the difference between the actual results of each calculation and the optimal solution.

Comparison with different OBL variants

CEC2015

CEC2015 [80] is an unconstrained single-objective optimization test set based on actual parameters. Nine algorithms are applied to the CEC2015 constraint function test set for testing. The 15 constrained problems of CEC2015 include unimodal functions (F1, F2), simple multimodal functions (F3–F5), hybrid functions (F6–F8), and composition functions (F9–F15). SQOPSO and the PSO algorithm embedded

Table 4 Means and standard deviations of PSO algorithm with different OBL variants for F1–F15 of 30D in CEC2015

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	1.58E+06	2.06E+06	1.37E+06	1.98E+06	1.32E+06	1.53E+06	1.60E+06	2.14E+06	**7.89E+05**
	Std	1.20E+09	2.04E+09	1.49E+09	1.35E+09	1.43E+09	1.74E+09	1.62E+09	1.79E+09	1.29E+06
F2	Mean	1.05E+07	1.18E+07	1.31E+07	9.38E+06	1.37E+07	1.60E+07	1.38E+07	1.28E+07	**3.80E+01**
	Std	5.16E+10	4.24E+10	6.06E+10	2.86E+06	6.37E+10	6.65E+10	2.18E+06	5.96E+10	3.41E+10
F3	Mean	2.09E+01	**2.07E+01**	2.08E+01	2.09E+01	**2.07E+01**	2.08E+01	2.09E+01	2.08E+01	**2.07E+01**
	Std	4.35E−02	7.46E−02	8.45E−02	4.31E−02	6.65E−02	5.53E−02	4.03E−02	8.66E−02	5.98E−02
F4	Mean	1.29E+02	1.17E+02	1.29E+02	1.13E+02	1.23E+02	1.09E+02	**1.04E+02**	1.20E+02	1.08E+02
	Std	2.46E+01	1.45E+02	1.41E+02	2.78E+01	1.65E+02	1.27E+02	3.12E+01	3.78E+01	3.55E+01
F5	Mean	3.34E+03	2.93E+03	3.03E+03	3.22E+03	2.98E+03	3.20E+03	2.82E+03	2.57E+03	**2.18E+03**
	Std	5.31E+02	1.02E+03	6.29E+02	5.63E+02	5.90E+02	4.82E+02	1.22E+03	7.38E+02	9.29E+02
F6	Mean	7.28E+04	3.04E+04	4.64E+04	7.45E+04	6.23E+04	5.01E+04	6.47E+04	3.99E+04	**1.94E+04**
	Std	1.34E+08	1.24E+08	1.68E+08	9.95E+07	1.35E+08	1.04E+08	1.48E+08	1.57E+08	9.74E+04
F7	Mean	9.52E+00	1.01E+01	1.05E+01	1.11E+01	9.21E+00	1.04E+01	**8.20E+00**	9.73E+00	8.94E+00
	Std	4.09E+02	4.44E+02	4.74E+02	4.98E+02	5.27E+02	4.44E+02	4.19E+02	4.93E+02	1.44E+02
F8	Mean	2.46E+04	4.01E+04	3.60E+04	2.19E+04	**1.33E+04**	2.99E+04	3.44E+04	2.24E+04	2.06E+04
	Std	1.79E+07	3.89E+07	3.98E+07	9.09E+06	3.90E+07	2.86E+07	1.04E+07	3.46E+07	3.39E+04
F9	Mean	1.03E+02	1.03E+02	1.03E+02	1.03E+02	1.03E+02	1.03E+02	1.03E+02	1.03E+02	**1.03E+02**
	Std	2.39E+02	1.55E+02	2.86E+02	2.63E+02	2.65E+02	2.38E+02	2.47E+02	2.65E+02	4.65E−01
F10	Mean	7.76E+04	4.49E+04	4.56E+04	6.39E+04	5.70E+04	4.96E+04	5.34E+04	6.03E+04	**2.49E+04**
	Std	5.42E+07	1.01E+08	9.80E+07	9.15E+07	9.33E+07	6.11E+07	4.60E+07	7.95E+07	1.54E+05
F11	Mean	3.05E+02	3.06E+02	3.04E+02	3.07E+02	3.04E+02	3.06E+02	3.04E+02	3.06E+02	**3.03E+02**
	Std	8.14E+02	8.57E+02	7.14E+02	5.95E+02	8.16E+02	7.15E+02	7.52E+02	8.33E+02	5.04E+02
F12	Mean	1.07E+02	1.07E+02	1.07E+02	1.10E+02	1.08E+02	**1.06E+02**	1.09E+02	1.07E+02	1.08E+02
	Std	6.12E+01	3.37E+01	6.11E+01	3.92E+01	7.43E+01	7.07E+01	7.18E+01	7.38E+01	1.79E+00
F13	Mean	1.51E+02	1.42E+02	1.44E+02	1.53E+02	1.47E+02	1.49E+02	1.48E+02	1.45E+02	**1.27E+02**
	Std	3.03E+02	4.98E+02	2.73E+02	4.37E+01	3.40E+02	4.28E+02	1.49E+02	6.64E+02	3.78E+01
F14	Mean	3.15E+04	3.14E+04	3.12E+04	8.07E+04	**1.04E+03**	3.12E+04	3.62E+04	1.07E+03	3.12E+04
	Std	3.91E+04	3.40E+04	5.75E+04	1.72E+04	6.29E+04	6.05E+04	1.87E+04	4.50E+04	1.65E+03
F15	Mean	1.02E+02	1.02E+02	1.02E+02	1.02E+02	1.02E+02	1.02E+02	1.02E+02	1.02E+02	**1.00E+02**
	Std	3.92E+00	7.09E+04	1.25E+05	2.20E+00	1.07E+05	1.23E+05	4.93E+00	1.11E+05	5.43E+04

with 8 OBL variants were tested on 30-D and 50-D on CEC2015. The specific experimental results of 30 dimensions are shown in Table 4, and the best results are highlighted in bold:

1. In the unimodal functions (F1, F2), it can be clearly seen that SQOPSO has achieved better solutions on all unimodal functions, and its performance exceeds the other 8 algorithms.
2. In the simple multimodal function (F3–F5), SQOPS-O finds the global optimal solution on the functions F3 and F5. Regarding function F4, EOPSO has better performance, and the optimal solution of SQOPSO is not much different from the result of EOPSO.
3. Among the hybrid functions (F6–F8), SQOPSO has a significant advantage on functions F6 and F7. Regarding the function F8, the results of the COPSO algorithm are better than the rest of the algorithms.
4. Among the composition functions (F9–F15), SQOPS-O finds the global optimal solution except for the F14 function. On the F14 function, REOPSO and COPSO have the best performance, followed by SQ-OPSO.

In most functions, the SQOPSO algorithm has significant advantages over other comparative algorithms. Therefore, SQOPSO has the best performance on the 30-dimensional CE-C2015 benchmark test set.

To intuitively display the convergence effect of each algorithm, the convergence curves of nine algorithms in 30 dimensions are shown in Fig. 10. Observing the trend change, the convergence curve of SQOPSO confirms our theory. In the early stage of iteration, affected by the double mean cen-

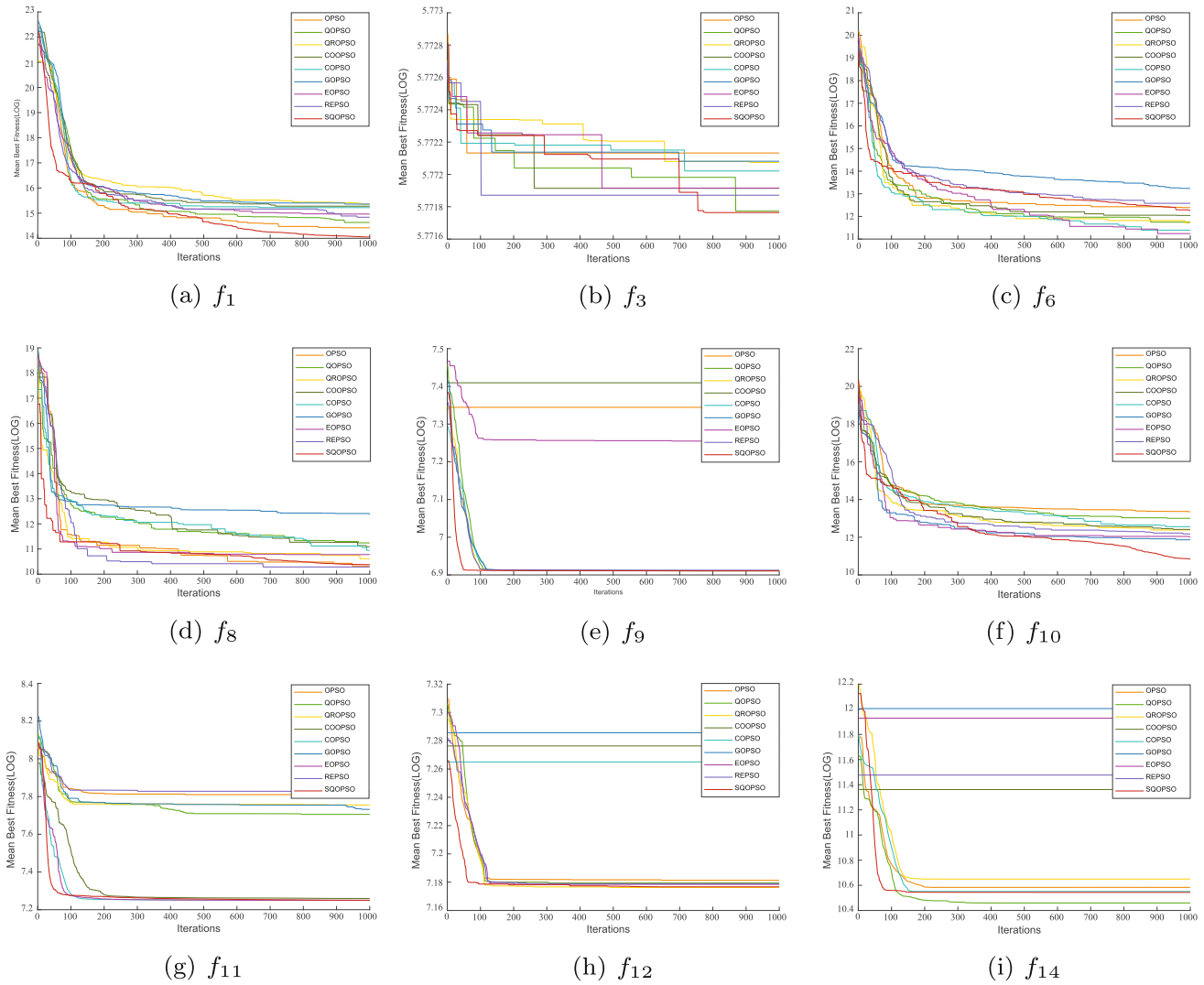


Fig. 10 Convergence curves of PSO algorithm with different OBL variants in 30-D on functions F1, F3, F6, F8, F9, F10, F11, F12 and F14

ter, it can quickly converge to the current optimum compared with other algorithms. But when the population diversity is lower than the set threshold, change the opposing strategy, increase the particle search range, help the particles jump out of the local optimum, and prevent premature maturity.

To study the impact of variables of different dimensions on the performance of the algorithm, we conducted tests on 50-D. Table 5 lists the mean, standard deviation, Wilcoxon rank sum test, and Friedman test results.

1. In the unimodal functions (F1, F2), it can be clearly seen that SQOPSO has achieved better solutions on all unimodal functions, and its performance exceeds the other 8 algorithms.
2. Among the simple multimodal functions (F3–F5), SQOPSO finds the global optimal solution on the functions F3, F4 and F5. On the function F4, the optimal solutions of each algorithm have little difference.
3. Among the hybrid functions (F6–F8), SQOPSO has a significant advantage on functions F6 and F7. Regarding the function F8, the results of the REOPSO algorithm are better than the rest of the algorithms.
4. In the composition functions (F9–F15), SQOPSO found the global optimal solution in all functions. On the F11 function, EOPSO, QROPSO and SQOP-SO all have the best performance. On functions F12 and F15, the performance of all algorithms is not much different.

It can be found that the SQOPSO algorithm performs best on 14 functions, and the REOPSO algorithm performs best on 1 function. In most functions, the SQOPSO algorithm has significant advantages over other comparative algorithms. Therefore, SQOPSO has the best performance on the 50-dimensional CEC2015 benchmark test set.

Table 5 Means and standard deviations of PSO algorithm with different OBL variants for F1–F15 of 50-D in CEC2015

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	1.05E+07	1.18E+07	1.04E+07	8.12E+06	1.07E+07	1.41E+07	1.10E+07	6.25E+06	**4.60E+06**
	Std	4.43E+09	4.70E+09	1.77E+09	4.88E+09	4.08E+09	5.06E+09	3.00E+09	4.48E+09	4.03E+09
F2	Mean	4.65E+07	5.72E+07	4.70E+07	4.96E+07	4.51E+07	4.81E+07	5.20E+07	4.14E+07	**1.36E+03**
	Std	6.25E+10	4.33E+10	8.59E+10	1.11E+11	1.13E+11	1.29E+11	1.19E+09	9.99E+10	6.68E+08
F3	Mean	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.10E+01	2.11E+01	2.11E+01	2.10E+01	**2.08E+01**
	Std	4.32E−02	3.03E−02	5.28E−02	8.66E−02	4.06E−02	3.83E−02	4.51E−02	4.44E−02	3.43E−02
F4	Mean	2.96E+02	3.23E+02	3.17E+02	3.03E+02	2.59E+02	2.52E+02	3.04E+02	3.09E+02	**2.20E+02**
	Std	2.42E+02	1.33E+02	2.07E+02	4.46E+01	2.82E+02	2.05E+02	2.25E+01	1.05E+02	3.21E+01
F5	Mean	5.86E+03	5.68E+03	5.54E+03	6.19E+03	5.54E+03	5.64E+03	4.74E+03	5.24E+03	**4.38E+03**
	Std	9.09E+02	9.46E+02	1.54E+03	8.74E+02	9.61E+02	9.85E+02	1.12E+03	1.28E+03	8.72E+02
F6	Mean	3.78E+05	3.35E+05	3.19E+05	4.69E+05	3.04E+05	3.93E+05	3.52E+05	3.23E+05	**2.18E+05**
	Std	2.63E+08	3.64E+08	3.90E+05	3.16E+08	3.67E+08	2.37E+08	2.55E+08	2.50E+08	2.87E+08
F7	Mean	1.85E+01	2.21E+01	1.95E+01	1.93E+01	1.94E+01	2.07E+01	1.77E+01	1.88E+01	**1.60E+01**
	Std	1.55E+03	1.04E+03	7.15E+02	1.81E+03	1.91E+03	1.68E+03	9.21E+02	1.44E+03	1.33E+03
F8	Mean	5.60E+05	3.92E+05	3.01E+05	3.36E+05	7.45E+05	4.14E+05	4.98E+05	**2.09E+05**	2.69E+05
	Std	5.04E+07	7.76E+07	3.58E+07	5.60E+07	5.38E+07	6.44E+07	4.62E+05	6.16E+07	6.09E+07
F9	Mean	1.05E+02	1.05E+02	1.06E+02	1.06E+02	1.05E+02	1.05E+02	1.05E+02	1.05E+02	**1.05E+02**
	Std	5.65E+02	5.61E+02	2.81E+02	6.63E+02	6.64E+02	6.65E+02	6.17E+02	5.60E+02	6.65E+02
F10	Mean	2.16E+05	2.28E+05	2.57E+05	3.47E+05	1.08E+05	1.40E+05	1.78E+05	2.60E+05	**4.30E+04**
	Std	1.78E+08	1.82E+08	1.83E+05	2.07E+08	1.26E+08	1.38E+08	8.59E+07	9.95E+07	1.94E+08
F11	Mean	3.20E+02	1.68E+03	3.13E+02	3.18E+02	1.64E+03	3.22E+02	3.15E+02	3.16E+02	**3.06E+02**
	Std	7.62E+02	8.03E+02	6.98E+02	1.07E+03	7.18E+02	8.90E+02	1.03E+03	1.01E+03	6.88E+02
F12	Mean	1.11E+02	1.12E+02	1.11E+02	2.64E+02	1.11E+02	1.12E+02	**1.10E+02**	1.12E+02	1.11E+02
	Std	1.15E+02	7.06E+00	6.21E+01	1.28E+02	1.43E+02	1.33E+02	7.66E+01	1.05E+02	2.95E+01
F13	Mean	2.56E+02	2.54E+02	2.56E+02	2.77E+02	2.54E+02	2.59E+02	2.59E+02	2.61E+02	**2.29E+02**
	Std	1.19E+03	8.48E+01	7.85E+01	1.67E+03	1.36E+03	1.31E+03	6.93E+02	9.15E+02	2.46E+02
F14	Mean	7.12E+03	4.98E+04	1.89E+03	1.78E+05	5.25E+04	5.12E+04	7.82E+04	1.96E+03	**1.10E+02**
	Std	1.99E+05	1.22E+05	1.88E+04	1.97E+05	2.00E+05	1.99E+05	4.97E+04	2.10E+05	3.48E+04
F15	Mean	1.03E+02	6.93E+04	1.09E+02	1.02E+02	1.03E+02	1.03E+02	1.02E+02	1.03E+02	**1.00E+02**
	Std	3.57E+00	4.44E+04	8.53E+04	5.17E+00	1.98E+05	1.68E+05	4.76E+00	1.89E+05	3.99E+00

The optimal value under the current index is in bold

The last row in Table 5 gives the average grade value of all algorithms based on the Friedman test on all test functions. The ranking of the nine algorithms after the average grade is SQOPSO, REOPSO, COPSO, EOPSO, QROPSO, OPSO, GOPSO, COOPSO, and QOPSO. This statistical result further verifies that the proposed SQOPSO algorithm has the best performance among all the algorithms participating in the comparison, and the SQOPSO algorithm performs better on high-dimensional data sets, ranking about 0.7 higher.

To intuitively display the convergence effect of each algorithm, the convergence curves of nine algorithms in 50 dimensions are given in Fig. 11. Compared with Fig. 10, the convergence effect of Fig. 11 is more obvious, indicating that the SQOPSO algorithm has significant advantages compared with other OBL variants when dealing with high-dimensional problems.

CEC2017

CEC2017 [81] is a test set with constraint functions proposed in the past five years. Nine algorithms are applied to the test set of CEC2017 constraint functions for testing. Among the 29 constrained problems of CEC2017 are unimodal functions (F1, F2), simple multimodal functions (F3–F9), hybrid functions (F10–F19) and composition functions (F20–F29). SQOPSO and 8 PSOs with OBL variants were tested in 30-D and 50-D at CEC2017. The specific experimental results are shown in Table 6, and the best results are highlighted in bold:

1. In the unimodal function (F1, F2), SQOPSO achieves a better solution on the function F1, and its performance exceeds the other 8 algorithms. But on the function F2, the effect of COPSO is better, and the performance of

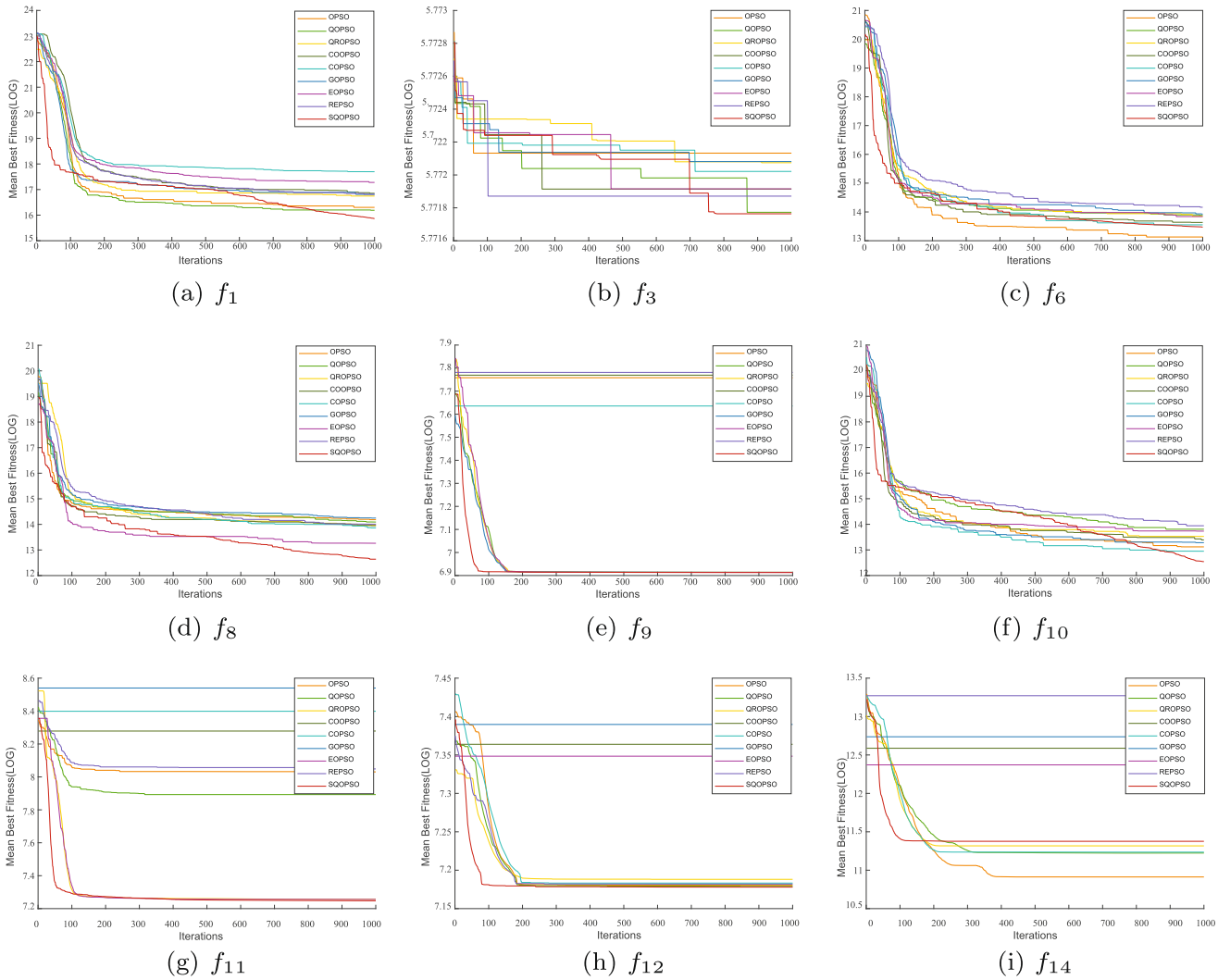


Fig. 11 Convergence curves of PSO algorithm with different OBL variants in 50-D on functions F1, F3, F6, F8, F9, F10, F11, F12 and F14

OPSO, COOPSO and REOPSO algorithms are all better than SQOPSO.

2. Among the simple multimodal functions (F3–F9), SQOPSO finds the global optimal solution on the functions F3, F4, F5, F6 and F8. Regarding functions F7 and F9, COOPSO and QOPSO have better performance, and the optimal solution of SQOSPO is not much different from the results of COOPSO and QOPSO.
3. Among the hybrid functions (F10–F19), SQOPSO has significant advantages on functions F10, F11, F12, F13, F14, F15, F16 and F18. In function F10 and function F16, there is little difference in the performance of GOPSO, REOPSO and SQOPSO. On function F17 and function F19, the result of OPSO algorithm is better than other algorithms.
4. Among the composition functions (F20–F29), SQOP SO has significant advantages in functions F21, F24, F52, F26, F27, F28 and F29. Among the functions F20, F22 and F23, the performance of COPSO, OPS O, REOPSO and GOPSO is better. While on function F24 and function F27, the performance of COO PSO, COPSO, GOPSO and SQOPSO is not much different.

It can be found that the SQOPSO algorithm performs best on 21 functions, the REOPSO algorithm performs best on 2 functions, and the OPSO and GOPSO perform best on 3 functions. QOPSO performs best on 1 function, and COOPSO and COPSO perform best on 4 functions. In most functions, the SQOPSO algorithm has significant advantages over other comparative algorithms. Therefore, SQOPSO has better performance on the 30-dimensional CEC2017 benchmark test set.

Table 6 Means and standard deviations of PSO algorithm with different OBL variants for F1–F29 of 30-D in CEC2017

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	1.39E+07	1.85E+07	1.25E+07	1.03E+07	1.14E+07	1.18E+07	1.12E+07	1.37E+07	**5.11E-01**
	Std	3.22E+10	1.66E+10	2.50E+10	2.22E+06	4.08E+10	4.20E+10	2.30E+06	4.06E+10	2.83E+10
F2	Mean	3.28E+02	4.73E+02	4.73E+02	3.51E+02	**2.25E+02**	3.85E+02	3.94E+02	3.31E+02	3.68E+02
	Std	2.45E+02	2.90E+04	3.01E+04	6.63E+02	3.27E+04	4.96E+02	3.15E+02	5.08E+02	5.05E+02
F3	Mean	1.59E+02	1.60E+02	1.63E+02	**1.05E+02**	1.07E+02	1.68E+02	1.50E+02	1.08E+02	**1.05E+02**
	Std	1.12E+04	9.03E+03	5.27E+03	2.82E+01	1.33E+04	1.43E+04	6.45E+03	1.46E+04	1.51E+04
F4	Mean	2.21E+02	2.82E+02	2.30E+02	2.28E+02	2.33E+02	2.27E+02	2.07E+02	2.32E+02	**1.67E+02**
	Std	8.18E+01	1.01E+02	4.99E+01	2.89E+01	1.19E+02	1.55E+02	2.76E+01	1.43E+02	1.43E+02
F5	Mean	1.36E+02	1.40E+02	1.40E+02	1.33E+02	1.42E+02	1.42E+02	1.41E+02	1.36E+02	**1.28E+02**
	Std	6.20E+00	2.90E+01	1.23E+01	6.67E+00	5.81E+00	1.32E+01	4.65E+00	2.19E+01	2.16E+01
F6	Mean	3.39E+02	6.21E+02	3.29E+02	3.17E+02	3.75E+02	3.71E+02	3.72E+02	3.75E+02	**2.11E+02**
	Std	4.95E+01	1.74E+02	3.92E+02	5.35E+01	6.19E+01	5.90E+01	5.09E+01	6.13E+01	3.80E+01
F7	Mean	2.01E+02	2.27E+02	2.03E+02	**1.69E+02**	2.03E+02	1.91E+02	1.97E+02	1.86E+02	1.77E+02
	Std	7.59E+01	1.17E+02	5.37E+01	3.57E+01	8.11E+01	7.78E+01	1.78E+01	3.82E+01	2.13E+01
F8	Mean	2.73E+03	3.75E+03	2.45E+03	2.89E+03	3.27E+03	3.17E+03	2.91E+03	3.77E+03	**1.60E+03**
	Std	1.27E+03	5.53E+03	2.85E+03	1.27E+03	3.78E+03	1.19E+03	1.19E+03	1.56E+03	5.94E+02
F9	Mean	3.13E+03	2.24E+03	3.80E+03	2.99E+03	3.20E+03	2.99E+03	3.55E+03	3.07E+03	**2.45E+03**
	Std	6.52E+02	1.24E+03	5.03E+02	5.97E+02	1.01E+03	6.38E+02	5.19E+02	5.74E+02	6.27E+02
F10	Mean	1.69E+02	1.80E+02	1.98E+02	1.69E+02	1.67E+02	1.63E+02	1.71E+02	1.85E+02	**1.62E+02**
	Std	3.33E+01	4.92E+03	4.17E+03	3.65E+01	4.98E+03	4.20E+01	3.43E+01	3.57E+03	4.17E+03
F11	Mean	2.61E+06	2.91E+06	2.95E+06	2.78E+06	3.02E+06	2.36E+06	3.73E+06	3.86E+06	**5.22E+04**
	Std	5.23E+09	8.20E+09	4.03E+06	5.08E+09	9.85E+09	9.94E+09	3.22E+09	8.64E+09	1.00E+10
F12	Mean	2.60E+05	4.27E+05	4.79E+05	3.71E+05	2.02E+05	2.80E+05	2.66E+05	3.00E+05	**3.39E+02**
	Std	6.72E+09	1.03E+10	1.61E+05	7.94E+09	8.43E+09	8.16E+09	7.44E+09	7.84E+09	8.67E+09
F13	Mean	8.29E+02	1.37E+03	9.14E+02	1.40E+03	1.16E+03	8.60E+02	9.79E+02	1.25E+03	**3.42E+02**
	Std	3.82E+06	4.68E+06	2.15E+06	5.22E+06	4.63E+06	9.16E+05	8.13E+05	4.37E+06	1.69E+06
F14	Mean	2.51E+04	4.04E+04	3.39E+04	3.77E+04	5.96E+04	5.18E+04	4.19E+04	3.81E+04	**2.76E+02**
	Std	3.28E+08	9.31E+08	3.80E+08	1.69E+09	7.03E+08	5.37E+08	3.75E+04	9.76E+08	4.38E+08
F15	Mean	1.12E+03	9.30E+02	9.62E+02	1.32E+03	1.28E+03	9.89E+02	9.67E+02	7.04E+02	**4.80E+02**
	Std	1.70E+03	1.94E+03	2.58E+02	1.80E+03	2.14E+03	2.21E+03	2.36E+03	2.34E+03	2.25E+03
F16	Mean	2.25E+02	2.51E+02	3.26E+02	5.66E+02	4.58E+02	3.00E+02	3.33E+02	**2.20E+02**	2.21E+02
	Std	5.42E+03	7.55E+03	2.38E+02	2.35E+03	2.92E+03	1.89E+03	2.11E+02	2.31E+03	5.70E+03
F17	Mean	**3.29E+04**	9.14E+04	3.49E+04	5.44E+04	1.18E+05	6.87E+04	6.43E+04	5.93E+04	5.99E+04
	Std	6.72E+07	9.91E+07	1.11E+05	5.07E+07	6.23E+07	1.54E+08	1.50E+05	2.34E+07	7.94E+07
F18	Mean	6.17E+04	1.67E+05	1.47E+05	1.01E+05	9.02E+04	1.12E+05	1.67E+05	9.46E+04	**2.52E+02**
	Std	4.19E+08	1.14E+09	3.77E+08	1.16E+09	8.95E+08	9.34E+08	3.44E+05	1.23E+09	1.23E+09
F19	Mean	**3.10E+02**	4.28E+02	4.64E+02	4.17E+02	3.57E+02	4.25E+02	3.84E+02	4.55E+02	4.11E+02
	Std	2.22E+02	2.20E+02	2.26E+02	1.95E+02	2.26E+02	1.60E+02	1.71E+02	1.97E+02	2.25E+02
F20	Mean	**2.07E+02**	4.67E+02	4.24E+02	4.06E+02	2.10E+02	4.34E+02	4.21E+02	4.52E+02	3.92E+02
	Std	1.53E+02	1.42E+02	4.38E+01	9.05E+01	1.86E+02	1.63E+02	3.16E+01	1.56E+02	1.65E+02
F21	Mean	2.22E+02	2.20E+02	2.21E+02	2.20E+02	2.22E+02	2.22E+02	2.21E+02	2.22E+02	**2.00E+02**
	Std	2.16E+03	2.72E+03	3.04E+03	2.40E+03	2.63E+03	2.55E+03	2.39E+03	2.82E+03	1.99E+03
F22	Mean	8.75E+02	8.02E+02	8.03E+02	1.06E+03	**7.66E+02**	7.82E+02	8.32E+02	7.70E+02	7.79E+02
	Std	2.82E+02	1.25E+02	1.25E+02	1.45E+02	3.36E+02	2.44E+02	2.49E+02	2.51E+02	3.05E+02
F23	Mean	7.64E+02	8.17E+02	7.91E+02	1.15E+03	9.02E+02	5.86E+02	9.57E+02	9.75E+02	**7.60E+02**
	Std	3.72E+02	1.27E+02	1.21E+02	1.25E+02	3.32E+02	4.80E+02	2.35E+02	4.02E+02	4.57E+02

Table 6 continued

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F24	Mean	4.87E+02	4.90E+02	4.88E+02	4.85E+02	4.85E+02	4.85E+02	4.85E+02	4.86E+02	**4.83E+02**
	Std	2.04E+01	1.40E+03	2.06E+03	2.01E+01	2.63E+03	3.12E+03	1.48E+01	2.57E+03	3.34E+00
F25	Mean	3.90E+02	4.26E+03	3.79E+02	4.02E+02	3.87E+02	3.94E+02	3.75E+02	3.86E+02	**3.00E+02**
	Std	3.60E+03	2.07E+03	2.83E+03	4.49E+03	3.89E+03	3.78E+03	4.02E+03	3.94E+03	4.09E+03
F26	Mean	7.98E+02	7.49E+02	7.08E+02	1.04E+03	7.17E+02	8.02E+02	8.11E+02	8.09E+02	**6.77E+02**
	Std	6.21E+02	4.80E+02	1.78E+02	3.66E+02	5.67E+02	5.92E+02	5.68E+02	6.01E+02	7.90E+02
F27	Mean	5.00E+02	5.07E+02	4.97E+02	4.81E+02	5.01E+02	4.98E+02	5.06E+02	5.03E+02	**4.76E+02**
	Std	2.66E+03	2.01E+03	2.35E+03	1.75E+03	3.47E+03	3.46E+03	1.42E+03	2.47E+03	2.34E+03
F28	Mean	1.25E+03	1.20E+03	1.07E+03	1.28E+03	1.03E+03	1.24E+03	9.08E+02	1.24E+03	**7.94E+02**
	Std	1.56E+03	3.34E+03	7.09E+02	3.43E+03	4.73E+03	2.67E+03	2.47E+02	3.18E+03	8.15E+03
F29	Mean	5.12E+05	1.13E+06	1.78E+06	4.40E+05	8.00E+05	9.35E+05	1.09E+06	7.82E+05	**4.52E+03**
	Std	3.58E+08	1.57E+09	8.16E+05	1.05E+09	1.63E+09	1.44E+09	1.39E+09	1.01E+09	1.36E+09

The optimal value under the current index is in bold

In order to study the impact of variables of different dimensions on the performance of the algorithm, we conducted tests on 50-D. Table 7 lists the mean, standard deviation, Wilcoxon rank sum test and Friedman test results.

1. In the unimodal functions (F1, F2), SQOPSO has obtained a better solution on the function F1, and its performance exceeds the other 8 algorithms. But on the function F2, the effect of COPSO is better, and the performance of OPSO, COOPSO and REOPSO algorithms are all better than SQOPSO.
2. Among the simple multimodal functions (F3–F9), SQOPSO finds the global optimal solution on the functions F3, F4, F5, F6 and F8. Regarding functions F7 and F9, COOPSO and QOPSO have better performance, and the optimal solution of SQOSPO is not much different from the results of COOPSO and QOPSO.
3. Among the hybrid functions (F10–F19), SQOPSO has significant advantages on functions F10, F11, F12, F13, F14, F15, F16 and F18. In function F10 and function F16, there is little difference in the performance of GOPSO, REOPSO and SQOPSO. On function F17 and function F19, the result of OPSO algorithm is better than other algorithms.
4. Among the composition functions (F20–F29), SQO-PSO has significant advantages in functions F21, F24, F52, F26, F27, F28 and F29. Among the functions F20, F22 and F23, the performance of COPSO, OPSO, REOPSO and GOPSO is better. While on function F24 and function F27, the performance of COOPSO, COPSO, GOPSO and SQOPSO is not much different.

It can be found that the SQOPSO algorithm performs best on 21 functions, the REOPSO algorithm performs best on 2 functions, and the OPSO and GOPSO perform best on 3 functions. QOPSO performs best on 1 function, and COOPSO and COPSO perform best on 4 functions. In most functions, the SQOPSO algorithm has significant advantages over other comparative algorithms. Therefore, SQOPSO has better performance on the 50-dimensional CEC2017 benchmark test set.

CEC2022

CEC2022 [82] is the latest single-objective bounded constraint optimization function test set, and 9 algorithms are applied to the CEC2022 function test set for testing. The 12 constrained problems of CEC2022 include unimodal functions (F1), basic functions (F2–F5), hybrid functions (F6–F8) and composition functions (F9–F12). SQOPSO and the PSO algorithm embedded with 8 OBL variants were tested on CEC2022 for 30-D and 50-D. The specific experimental results are as follows, and the best results are highlighted in bold:

1. In the unimodal function (F1), EOPSO, REOPSO and SQOPSO achieved better results than other algorithms.
2. Among the basic functions (F2–F5), SQOPSO has a significant advantage on function F3. There is little difference in the performance of all algorithms on functions F2 and F5. Regarding function F4, GOPSO has better performance, and the optimal solution of SQOSPO is not much different from the result of GOPSO.
3. Among the hybrid functions (F6–F8), SQOPSO has a significant advantage on functions F6 and F7. Regarding the function F8, the results of the QOPSO algorithm are better than the rest of the algorithms.
4. In the composition function (F9–F12), the performance difference of all algorithms is not big. But the perfor-

Table 7 Means and standard deviations of PSO algorithm with different OBL variants for F1–F29 of 50-D in CEC2017

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	4.66E+07	1.06E+11	4.89E+07	4.56E+07	4.19E+07	4.59E+07	4.01E+07	4.13E+07	**2.89E+03**
	Std	5.27E+10	1.35E+10	6.75E+10	5.92E+10	3.95E+10	7.79E+10	1.71E+08	4.98E+10	3.92E+08
F2	Mean	**2.03E+04**	2.93E+04	2.48E+04	2.66E+04	2.72E+04	3.08E+04	2.70E+04	3.23E+04	2.66E+04
	Std	9.73E+03	8.34E+04	7.38E+04	9.45E+04	1.22E+04	1.30E+04	6.98E+03	7.34E+03	9.28E+03
F3	Mean	1.46E+02	1.72E+02	1.37E+02	1.42E+02	1.88E+02	1.58E+02	**1.35E+02**	1.91E+02	1.36E+02
	Std	5.92E+01	1.49E+04	2.34E+04	2.84E+04	3.71E+04	2.19E+04	5.55E+01	2.66E+04	5.46E+01
F4	Mean	3.21E+02	3.89E+02	3.40E+02	3.24E+02	3.44E+02	3.51E+02	3.37E+02	3.43E+02	**2.89E+02**
	Std	5.76E+01	1.17E+02	1.35E+02	3.54E+01	1.21E+02	1.49E+02	3.78E+01	1.64E+02	3.50E+01
F5	Mean	1.56E+02	1.58E+02	1.54E+02	1.57E+02	1.55E+02	1.54E+02	1.53E+02	1.51E+02	**1.41E+02**
	Std	4.43E+00	2.85E+01	9.67E+00	3.83E+00	2.61E+01	1.81E+01	3.94E+00	1.35E+01	3.69E+00
F6	Mean	7.39E+02	**1.12E+03**	9.61E+02	7.00E+02	8.50E+02	8.01E+02	7.59E+02	7.85E+02	3.01E+02
	Std	7.75E+01	4.09E+02	4.56E+02	5.40E+01	8.81E+01	7.14E+01	7.63E+01	8.11E+02	6.62E+01
F7	Mean	3.49E+02	4.44E+02	3.63E+02	3.69E+02	3.54E+02	3.63E+02	3.19E+02	3.54E+02	**3.06E+02**
	Std	1.33E+02	1.61E+02	1.71E+02	1.92E+02	2.36E+02	1.46E+02	3.77E+01	2.14E+02	2.79E+01
F8	Mean	1.75E+04	2.31E+04	1.78E+04	1.92E+04	1.70E+04	1.79E+04	1.69E+04	1.80E+04	**6.90E+03**
	Std	3.19E+03	1.44E+04	8.13E+03	1.62E+03	1.00E+04	3.54E+03	2.46E+03	1.14E+04	2.50E+03
F9	Mean	6.16E+03	6.05E+03	6.25E+03	5.87E+03	6.17E+03	6.63E+03	6.02E+03	6.08E+03	**5.42E+03**
	Std	8.89E+02	2.75E+03	1.57E+03	5.13E+02	7.72E+02	7.68E+02	8.75E+02	7.95E+02	9.53E+02
F10	Mean	2.97E+02	3.12E+02	3.03E+02	3.03E+02	3.03E+02	2.74E+02	3.15E+02	3.03E+02	**1.97E+02**
	Std	4.82E+01	1.70E+04	1.59E+04	9.57E+03	5.63E+01	1.17E+04	3.39E+01	5.48E+01	5.29E+01
F11	Mean	2.32E+07	3.50E+07	2.01E+07	3.10E+07	2.38E+07	2.39E+07	2.08E+07	2.89E+07	**6.80E+05**
	Std	4.70E+10	3.64E+10	1.88E+10	5.33E+10	4.83E+10	6.17E+10	2.29E+10	5.44E+10	5.18E+10
F12	Mean	2.42E+06	2.57E+06	3.04E+06	2.31E+06	3.30E+06	2.89E+06	2.48E+06	1.54E+06	**1.30E+03**
	Std	3.45E+10	2.74E+10	7.12E+05	3.31E+10	2.91E+10	3.42E+10	2.57E+10	2.69E+10	3.12E+10
F13	Mean	2.77E+04	8.31E+03	**1.07E+04**	4.99E+04	2.69E+04	3.71E+04	2.93E+04	3.91E+04	1.37E+04
	Std	5.75E+07	1.07E+08	1.18E+05	8.29E+07	8.09E+07	8.51E+07	8.24E+07	8.63E+07	8.33E+07
F14	Mean	6.87E+05	8.01E+05	7.54E+05	4.76E+05	5.01E+05	5.55E+05	3.80E+05	4.74E+05	**5.95E+02**
	Std	5.45E+09	6.82E+09	3.48E+09	5.86E+09	6.10E+09	6.85E+09	3.32E+05	9.25E+09	5.64E+09
F15	Mean	1.65E+03	1.65E+03	1.51E+03	9.81E+02	1.59E+03	1.73E+03	1.69E+03	1.28E+03	**9.08E+02**
	Std	2.44E+03	3.51E+03	3.76E+02	4.43E+03	3.88E+03	4.20E+03	3.25E+03	3.12E+03	4.14E+03
F16	Mean	1.25E+03	1.15E+03	1.27E+03	1.22E+03	1.38E+03	1.15E+03	1.19E+03	1.38E+03	**1.06E+03**
	Std	5.16E+04	1.26E+04	6.42E+03	2.88E+02	6.14E+04	1.06E+04	3.29E+02	2.62E+04	3.07E+02
F17	Mean	2.07E+05	3.44E+05	3.20E+05	1.45E+05	3.90E+05	2.86E+05	2.49E+05	2.82E+05	**1.36E+05**
	Std	9.34E+07	2.25E+08	5.89E+07	1.77E+08	2.59E+08	1.05E+08	1.21E+08	9.94E+07	1.13E+08
F18	Mean	4.51E+05	6.31E+05	6.03E+05	4.32E+05	5.24E+05	6.03E+05	5.47E+05	5.13E+05	**3.22E+02**
	Std	2.85E+09	3.47E+09	1.02E+09	3.27E+09	4.42E+09	3.17E+09	2.30E+09	3.68E+09	2.81E+09
F19	Mean	1.03E+03	1.03E+03	7.95E+02	8.51E+02	6.37E+02	9.85E+02	**6.54E+02**	8.28E+02	8.02E+02
	Std	3.26E+02	2.63E+02	3.31E+02	2.47E+02	3.78E+02	3.05E+02	3.67E+02	3.52E+02	3.13E+02
F20	Mean	6.03E+02	6.47E+02	6.05E+02	6.05E+02	5.90E+02	6.11E+02	6.34E+02	5.94E+02	**5.67E+02**
	Std	2.11E+02	3.31E+02	7.41E+01	3.33E+02	3.66E+02	3.43E+02	4.24E+01	2.87E+02	7.48E+01
F21	Mean	7.51E+03	7.22E+03	6.11E+03	6.82E+03	7.60E+03	7.17E+03	5.98E+03	7.05E+03	**5.45E+03**
	Std	6.79E+02	2.11E+03	9.47E+02	1.75E+03	7.44E+02	1.70E+03	1.13E+03	9.59E+02	1.73E+03
F22	Mean	1.29E+03	1.41E+03	1.34E+03	1.91E+03	1.40E+03	1.39E+03	1.22E+03	1.51E+03	**1.17E+03**
	Std	4.98E+02	1.68E+02	1.51E+02	6.59E+02	5.43E+02	5.54E+02	5.23E+02	5.88E+02	1.86E+02
F23	Mean	1.30E+03	1.14E+03	1.24E+03	1.98E+03	1.20E+03	1.31E+03	1.44E+03	1.17E+03	**1.04E+03**
	Std	7.30E+02	4.13E+02	1.67E+02	9.36E+02	7.80E+02	8.26E+02	4.68E+02	8.97E+02	1.33E+02

Table 7 continued

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F24	Mean	6.11E+02	6.30E+02	5.90E+02	6.00E+02	**5.72E+02**	6.21E+02	6.17E+02	6.19E+02	6.25E+02
	Std	2.56E+01	4.37E+03	8.41E+03	2.32E+01	1.47E+04	2.97E+01	2.92E+01	1.07E+04	2.95E+01
F25	Mean	5.64E+02	1.59E+04	6.68E+02	5.63E+02	5.66E+02	6.17E+02	5.65E+02	6.12E+02	**5.51E+02**
	Std	7.12E+03	1.29E+03	5.79E+03	9.01E+03	9.57E+03	7.93E+03	8.02E+03	7.25E+03	8.05E+03
F26	Mean	1.44E+03	1.40E+03	1.73E+03	1.34E+03	1.55E+03	1.95E+03	1.56E+03	1.83E+03	**7.53E+02**
	Std	1.65E+03	5.44E+02	4.63E+02	1.74E+03	1.64E+03	1.63E+03	1.46E+03	1.41E+03	7.78E+02
F27	Mean	5.74E+02	**5.64E+02**	5.87E+02	5.78E+02	5.72E+02	5.81E+02	5.71E+02	5.82E+02	6.08E+02
	Std	4.61E+03	3.67E+03	4.71E+03	7.49E+03	8.48E+03	7.96E+03	3.99E+03	5.30E+03	3.72E+03
F28	Mean	1.95E+03	1.77E+03	1.95E+03	1.80E+03	1.98E+03	2.34E+03	1.88E+03	2.09E+03	**1.54E+03**
	Std	6.52E+04	1.71E+05	5.21E+02	2.02E+05	3.55E+05	1.47E+05	3.03E+05	1.26E+05	9.68E+04
F29	Mean	4.21E+07	4.11E+07	3.74E+07	4.22E+07	4.31E+07	4.01E+07	4.17E+07	4.43E+07	**6.79E+05**
	Std	3.72E+06	4.86E+09	3.45E+09	5.50E+09	4.30E+09	6.88E+09	4.95E+06	6.69E+09	5.42E+09

The optimal value under the current index is in bold

mance of SQOPSO on function F11 is better than other algorithms.

In order to study the impact of variables of different dimensions on the performance of the algorithm, we conducted tests on 20-D. Table 9 lists the mean, standard deviation, Wilcoxon rank sum test and Friedman test results.

1. In the unimodal function (F1), SQOPSO achieves a better solution on the function F1, and its performance exceeds the other 8 algorithms.
2. Among the basic functions (F2–F5), SQOPSO has significant advantages in all basic functions. Regarding function F3, the performance of SQOSPO and QROPSO is not much different.
3. Among the hybrid functions (F6–F8), SQOPSO has significant advantages on all blending functions.
4. Among the composition functions (F9–F12), SQOP-SO has advantages in all compound functions (except F11). COPSO performs better than other algorithms on function F11.

It can be found that SQOPSO algorithm performs best on 11 functions, and COPSO performs best on 1 function. In most functions, the SQOPSO algorithm has significant advantages over other comparative algorithms. Therefore, SQOPSO has the best performance on the 20-dimensional CEC2022 benchmark test set. Comparing Tables 8 and 9, it can be clearly seen that the higher the dimension, the better the performance of SQOPSO.

Figure 12 shows the comprehensive ranking results of nine algorithms in different dimensions on three benchmark test sets.

Comparison with the latest intelligent optimization algorithms

In recent years, in the field of intelligent computing, many more advantageous algorithms have been proposed and successfully applied to engineering practice. In order to further reflect the performance of SQOPSO algorithm, SQOPSO is compared with SSA, DMO, NGO, TSO, GJO, SCSO, GNDO and PSO. SQOPSO and 8 new intelligent optimization algorithms were tested in 10-D and 20-D on CEC2022. The specific experimental results are shown in Table 10, and the best results are highlighted in bold:

1. In the unimodal function (F1), SQOPSO, GNDO, TSO and NGO outperform the other 5 algorithms.
2. Among the basic functions (F2–F5), all algorithms perform roughly the same on function F2. SQOPSO, GNDO, SCSO, TSO, NGO, and DMO outperform other algorithms on function F3. NGO algorithm presents the best performance on function F4. SQO-PSO, PSO, GNDO, GJO, NGO, and DMO outperform other algorithms on function F5.
3. Among the hybrid functions (F6–F8), the performance of the GNDO algorithm on the function F6 is the best. SQOPSO, GNDO, TSO, NGO and DMO algorithms outperform other algorithms on function F7. SQOPSO, GNDO, SCSO and TSO perform better than other algorithms on function F8.
4. In the composition function (F9–F12), the performance differences of all algorithms on F9, F10 and F11 are not significant. In function F12, PSO, GNDO, SCSO, GJO, TSO, NGO and DMO performed better than other algorithms.

Table 8 Means and standard deviations of PSO algorithm with different OBL variants for F1–F12 of 10-D in CEC2022

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	5.87E−01	8.64E−01	7.67E−01	8.23E−01	9.43E−01	6.87E−01	4.88E−01	4.64E−01	**0.00E+00**
	Std	2.45E+03	4.75E+03	3.49E−01	2.62E−01	2.22E−01	3.11E+03	3.20E−01	3.08E−01	4.82E−01
F2	Mean	6.16E−02	9.93E−02	6.23E−02	3.07E−01	6.86E−02	1.49E−01	8.92E−02	6.70E−02	**3.00E−03**
	Std	4.36E+02	8.48E+02	1.82E+01	8.55E+02	8.86E+02	8.66E+02	7.30E+02	6.43E+02	9.48E+02
F3	Mean	6.30E+00	5.56E+00	5.25E+00	5.07E+00	5.59E+00	2.68E+00	1.85E+00	5.69E+00	**4.06E−01**
	Std	1.09E+01	1.86E+01	7.56E+00	1.39E+01	1.58E+01	7.46E+00	5.50E+00	1.07E+01	1.40E+01
F4	Mean	9.34E+00	1.41E+01	9.28E+00	7.29E+00	1.25E+01	**4.32E+00**	1.24E+01	5.35E+00	4.97E+00
	Std	9.70E+00	2.40E+01	9.34E+00	9.66E+00	9.33E+00	1.08E+01	1.02E+01	1.57E+01	8.36E+00
F5	Mean	1.27E−01	2.04E−01	2.27E−01	2.12E−01	1.92E−01	1.64E−01	1.92E−01	2.04E−01	**0.00E+00**
	Std	1.97E+02	5.17E+02	6.05E+01	4.31E+01	3.15E+02	3.95E+02	3.09E+01	1.72E+02	2.46E+02
F6	Mean	2.30E+02	4.80E+02	1.20E+02	6.53E+02	5.46E+02	3.62E+02	3.95E+02	5.24E+02	**8.60E+00**
	Std	1.04E+08	1.13E+08	1.67E+03	1.65E+08	3.75E+08	1.75E+08	2.38E+08	1.43E+08	3.13E+08
F7	Mean	4.92E+00	4.08E+00	1.33E+01	9.53E+00	1.72E+01	2.06E+01	2.25E+01	5.11E+00	**1.99E+00**
	Std	1.32E+01	2.61E+01	1.19E+01	1.53E+01	2.67E+01	2.21E+01	1.23E+01	1.57E+01	2.24E+01
F8	Mean	5.27E+00	**4.91E+00**	9.37E+00	2.32E+01	8.86E+00	2.19E+01	2.25E+01	2.30E+01	2.00E+01
	Std	3.02E+01	5.37E+01	3.92E+00	5.32E+01	8.48E+01	3.69E+01	4.26E+01	6.01E+01	6.65E+01
F9	Mean	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	**2.29E+02**
	Std	7.49E+01	1.40E+02	1.29E−03	1.10E+02	1.24E+02	6.67E+01	3.72E+01	1.10E+02	1.29E+02
F10	Mean	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	**1.00E+02**	1.00E+02	1.00E+02	1.00E+02
	Std	8.06E+01	1.87E+02	6.07E+01	7.81E+01	1.23E+02	1.91E+02	6.05E+01	1.15E+02	1.00E+02
F11	Mean	5.52E+00	6.22E+00	6.84E+00	7.12E+00	6.72E+00	5.45E+00	6.11E+00	6.70E+00	**0.00E+00**
	Std	1.69E+02	6.45E+02	9.66E+01	4.64E+02	4.65E+02	4.74E+02	2.37E+02	5.83E+02	4.29E+02
F12	Mean	1.67E+02	1.70E+02	1.69E+02	1.77E+02	**1.66E+02**	1.73E+02	1.72E+02	1.69E+02	1.68E+02
	Std	8.62E+01	4.08E+01	4.09E+01	3.91E+01	8.19E+01	1.38E+02	5.56E+01	8.19E+01	9.20E+01

The optimal value under the current index is in bold

It can be found that compared with the latest intelligent optimization algorithm, the performance of SQO-PSO algorithm is in the upper middle, but there is little difference with the optimal solution found by other algorithms. As a framework, SQOBL is applied to the PSO algorithm. After a number of tests, it can be clearly seen that it has been greatly improved compared with the PSO algorithm. If SQOBL is applied to the improvement of other intelligent optimization algorithms, it will have strong competitiveness.

This result further verifies that the proposed SQOBL strategy can greatly improve the optimization ability of the algorithm, and has a good competitiveness in the latest intelligent optimization algorithm.

In order to study the impact of variables of different dimensions on the performance of the algorithm, we conducted tests on 20-D. Table 11 lists the mean, standard deviation, Wilcoxon rank sum test and Friedman test results.

1. In the unimodal function (F1), the performance of SQOPSO, GNDO and TSO exceeds the other 6 algorithms.
2. Among the basic functions (F2–F5), the performance of SQOPSO and TSO on function F2 is more significant. The performance of SQOPSO and DMO on function F5 is better. For functions F3 and F4, NGO and DMO outperform the other 7 algorithms.
3. Among the hybrid functions (F6–F8), the GNDO algorithm has the best performance on functions F6 and F7. SQOPSO, GNDO, NGO and TSO perform better than other algorithms on function F8.
4. In the composition function (F9–F12), there is little difference in the performance of all algorithms on F9 and F10. In function F11, TSO and SQOPSO perform better than other algorithms. NGO algorithm is better than other algorithms in processing function F12.

It can be found that the SQOPSO algorithm performs well on 7 functions, and the TSO, GNDO and NGO algorithms all perform well on 6 functions. Consistent with the analysis of the 10-dimensional experimental results, observing the 20-dimensional test results, it can be found that compared with the latest intelligent optimization algorithm, the SQOPSO algorithm has better performance and has obvious advantages in dealing with high-dimensional problems.

Table 9 Means and standard deviations of PSO algorithm with different OBL variants for F1–F12 of 20-D in CEC2022

		OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
F1	Mean	6.70E+00	5.18E+00	5.50E+00	7.55E+00	5.13E+00	5.41E+00	7.30E+00	4.24E+00	**1.63E−02**
	Std	1.49E+00	1.97E+04	7.79E+03	1.86E+00	2.61E+04	2.16E+00	2.16E+04	1.42E+04	1.62E+00
F2	Mean	7.10E+00	4.92E+01	4.50E+00	5.25E+00	3.18E+00	4.51E+01	4.53E+00	6.99E+00	**2.52E−04**
	Std	1.54E+03	1.16E+03	1.13E+03	1.10E+03	2.25E+03	2.10E+03	1.81E+03	2.07E+03	1.59E+01
F3	Mean	2.48E+01	3.15E+01	**1.67E+01**	2.32E+01	2.71E+01	3.06E+01	2.25E+01	2.36E+01	1.72E+01
	Std	1.47E+01	2.41E+01	1.01E+01	8.67E+00	1.44E+01	1.43E+01	7.87E+00	1.88E+01	9.04E+00
F4	Mean	3.67E+01	3.22E+01	3.44E+01	4.10E+01	4.50E+01	3.89E+01	3.08E+01	4.52E+01	**2.79E+01**
	Std	2.00E+01	7.04E+01	4.44E+01	1.78E+01	4.91E+01	3.89E+01	5.57E+01	4.00E+01	2.01E+01
F5	Mean	4.61E+02	9.17E+02	5.97E+02	3.57E+02	3.21E+02	5.94E+02	4.67E+02	4.17E+02	**5.44E−01**
	Std	3.27E+02	1.91E+03	3.69E+02	3.36E+02	1.37E+03	1.35E+03	4.47E+02	5.71E+02	3.07E+02
F6	Mean	6.67E+04	7.24E+04	8.64E+04	8.82E+04	1.07E+05	9.52E+04	1.02E+05	8.66E+04	**1.19E+02**
	Std	4.77E+04	1.72E+09	8.47E+08	1.35E+09	1.69E+09	1.86E+09	1.34E+09	2.19E+09	6.56E+04
F7	Mean	6.38E+01	4.91E+01	5.04E+01	5.51E+01	5.91E+01	5.18E+01	5.09E+01	6.91E+01	**4.67E+01**
	Std	2.65E+01	5.96E+01	2.62E+01	2.44E+01	5.77E+01	6.81E+01	4.82E+01	4.25E+01	3.39E+01
F8	Mean	2.67E+01	2.70E+01	2.89E+01	2.94E+01	2.74E+01	2.64E+01	2.82E+01	2.88E+01	**2.09E+01**
	Std	2.59E+02	5.66E+02	5.91E+01	4.39E+02	2.23E+03	6.39E+02	1.54E+03	1.51E+02	5.95E+01
F9	Mean	1.81E+02	1.81E+02	1.81E+02	1.81E+02	1.81E+02	1.81E+02	1.81E+02	1.81E+02	**1.81E+02**
	Std	3.98E+02	4.85E+02	5.79E+00	4.26E+02	5.54E+02	5.45E+02	4.15E+02	4.46E+02	5.78E+00
F10	Mean	1.01E+02	1.01E+02	1.00E+02	1.01E+02	1.00E+02	1.01E+02	1.00E+02	1.01E+02	**1.00E+02**
	Std	1.20E+03	1.14E+03	1.01E+03	7.06E+02	1.05E+03	9.97E+02	9.59E+02	1.10E+03	9.99E+02
F11	Mean	3.53E+02	6.68E+01	6.95E+01	6.56E+01	**6.49E+01**	6.65E+01	6.64E+01	6.69E+01	3.00E+02
	Std	1.67E+03	2.85E+03	2.19E+03	2.63E+03	3.16E+03	2.57E+03	3.09E+03	3.42E+03	8.82E+01
F12	Mean	3.57E+02	4.15E+02	3.45E+02	3.71E+02	3.96E+02	4.38E+02	3.97E+02	3.34E+02	**3.05E+02**
	Std	3.87E+02	2.49E+02	1.81E+02	1.65E+02	3.16E+02	2.55E+02	4.74E+02	3.18E+02	2.71E+02

The optimal value under the current index is in bold

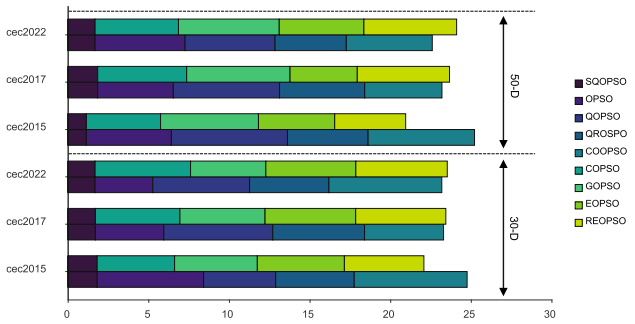


Fig. 12 Multi-group stack diagram showing the average rank of three testsets on nine algorithms

Realistic Constrained Optimization Problems

In this section, a practical engineering optimization problem is used to compare SQOBL with popular algorithms for better overall evaluation of SQOBL performance. The CEC2020 benchmark set is derived from real-world non-convex constrained optimization problems that reflect the difficulties and challenges that arise in real-world optimization scenarios [83].

In the comparative experiment, the top three advanced constrained optimization methods ranked in the Genetic and Evolutionary Computing Conference (GEC CO'20 Companion) were selected for exhaustive testing. The details of the methods are as follows:

1. SASS: Adaptive Sphere Search Algorithm. [84]
2. COLSHADE: Evolved from the basic L-SHADE algorithm, important features such as adaptive Levy flight and dynamic tolerance are introduced into the constraint processing technology. [85]
3. sCMAgES: A variant of Covariance Matrix Adaptive Evolutionary Strategy (CMA-ES) with linear temporal complexity. [86]

Table 10 Means and standard deviations of multiple intelligent optimization algorithms for 10-D F1–F12 in CEC2022

		PSO	GNDO	SCSO	GJO	TSO	NGO	DMO	SSA	SQOPSO
F1	Mean	1.10E+01	5.68E−14	3.30E+01	7.38E+01	**0.00E+00**	1.31E−12	6.83E+00	2.26E+00	**0.00E+00**
	Std	3.16E+03	6.23E−14	7.72E+02	1.44E+03	9.17E−14	1.92E−10	1.69E+01	3.71E+02	4.69E−14
F2	Mean	1.84E−01	**5.68E−14**	1.22E−01	1.07E−01	1.32E−04	2.66E−05	2.53E−01	1.09E−03	5.17E−02
	Std	7.06E+01	2.29E+01	2.82E+01	2.10E+01	3.45E+00	6.61E−02	2.56E+00	2.66E+01	7.91E+02
F3	Mean	1.06E+00	4.91E−02	2.91E−01	7.50E−01	3.15E−06	1.14E−13	**0.00E+00**	8.75E+00	1.14E−13
	Std	4.18E+00	2.14E+00	9.50E+00	4.74E+00	2.72E+00	9.66E−06	0.00E+00	1.16E+01	1.57E+01
F4	Mean	9.06E+00	6.96E+00	4.35E+00	1.07E+01	6.96E+00	**3.33E+00**	1.93E+01	8.95E+00	4.97E+00
	Std	8.53E+00	3.87E+00	8.11E+00	8.98E+00	6.91E+00	1.71E+00	3.45E+00	8.26E+00	7.79E+00
F5	Mean	2.86E−01	5.31E−09	2.47E+00	1.88E−01	6.33E−01	1.14E−13	**0.00E+00**	1.62E+02	**0.00E+00**
	Std	6.03E+00	8.87E+00	1.07E+02	4.25E+01	3.43E+01	6.48E−12	2.70E−12	1.33E+02	2.99E+02
F6	Mean	2.23E+02	**1.10E−01**	1.45E+02	2.15E+03	3.53E+01	5.48E+01	2.93E+02	1.18E+02	8.04E+00
	Std	1.72E+03	2.56E+00	2.15E+03	1.41E+03	5.80E+02	1.71E+01	1.55E+03	1.92E+03	2.55E+08
F7	Mean	2.37E+01	9.95E−01	2.24E+01	2.29E+01	9.95E−01	**1.05E−02**	1.22E+00	2.07E+01	4.97E+00
	Std	8.18E+00	1.21E+01	1.08E+01	1.14E+01	1.06E+01	3.55E+00	4.42E+00	2.73E+01	4.40E+01
F8	Mean	2.29E+01	**4.48E−01**	3.57E+00	5.85E+00	8.25E−01	7.92E+00	9.10E+00	2.37E+01	5.41E−01
	Std	4.03E+00	8.81E+00	5.57E+00	4.70E+00	6.87E+00	4.81E+00	4.14E+00	3.83E+01	1.90E+01
F9	Mean	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	2.29E+02	**2.29E+02**
	Std	4.10E+01	9.09E−14	3.39E+01	3.27E+01	1.58E-13	9.09E−14	0.00E+00	4.00E+01	1.51E+02
F10	Mean	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	**1.00E+02**
	Std	5.30E+01	5.46E+01	5.74E+01	5.57E+01	1.47E−01	2.07E+01	2.31E+01	7.68E+01	1.01E+02
F11	Mean	7.35E+00	4.55E−13	6.73E−01	5.35E+01	9.09E−13	4.55E−13	**0.00E+00**	4.48E−06	4.55E−13
	Std	1.51E+02	9.32E+01	9.21E+01	1.10E+02	8.93E+01	6.17E−13	7.18E−06	2.08E+02	3.89E+02
F12	Mean	1.63E+02	1.63E+02	1.60E+02	1.63E+02	**1.59E+02**	**1.59E+02**	1.61E+02	1.76E+02	1.70E+02
	Std	4.59E+01	2.51E+00	3.32E+00	1.02E+01	1.70E+01	1.45E+00	8.13E−01	5.18E+01	6.38E+01

The optimal value under the current index is in bold

In the previous GECCO'20 Companion, SASS ranked first. It is undeniable that SASS is an excellent constrained optimization algorithm, but it will be unstable when dealing with some real-world problems, resulting in a low score on the optimal value. Attach SQOBL to the SASS algorithm to solve engineering and mathematical problems, and observe the impact of the combination of the two on the overall performance of SASS. SQOBL acts on the population initialization phase and local update strategy of the SASS algorithm, and evaluates the current solution and the reverse solution at the same time to accelerate the search process. The SASS, COLSHADE, sCMAgES, and SASS-SQOBL algorithms were all run 25 times, and the results were expressed in mean, standard deviation, median, worst value, and optimal value.

Hydro-static Thrust Bearing Design Problem

Hydrostatic bearing is a kind of sliding bearing, which has the advantages of high load capacity and long working life and is widely used in precision manufacturing and aerospace, and other fields. The main optimization objective of this design problem is to optimize the bearing power loss using the four design variables of oil viscosity μ, bearing radius R, flow Q, and recess radius R_0, which contain seven nonlinear constraints in total. The problem is defined as follows:

Minimize:

$$f(\bar{x}) = \frac{QP_0}{0.7} + E_f \tag{58}$$

Subject to:

$$\begin{aligned}
&g_1(\bar{x}) = 1000 - P_0 \le 0\\
&g_2(\bar{x}) = W - 101000 \le 0\\
&g_3(\bar{x}) = 5000 - \frac{W}{\pi\left(R^2 - R_0^2\right)} \le 0\\
&g_4(\bar{x}) = 50 - P_0 \le 0\\
&g_5(\bar{x}) = 0.001 - \frac{0.0307}{386.4P_0}\left(\frac{Q}{2\pi Rh}\right) \le 0\\
&g_6(\bar{x}) = R - R_0 \le 0\\
&g_7(\bar{x}) = h - 0.001 \le 0
\end{aligned} \tag{59}$$

Table 11 Mean and standard deviation of PSO of various intelligent optimization algorithms of 20-D F1–F12 in CEC2022

		PSO	GNDO	SCSO	GJO	TSO	NGO	DMO	SSA	SQOPSO
F1	Mean	2.52E+03	**6.82E−13**	6.43E+02	4.65E+03	4.72E−02	1.87E+03	1.01E+04	6.97E+02	1.20E−02
	Std	4.56E+04	3.27E−08	3.34E+03	3.55E+03	1.68E+02	2.12E+04	2.95E+03	2.10E+03	7.78E+02
F2	Mean	4.92E+01	4.91E+01	5.95E+01	8.77E+01	1.94E−01	2.89E+01	4.65E+01	5.39E+01	**4.29E−03**
	Std	2.21E+03	1.31E+01	5.93E+01	6.91E+01	2.17E+01	2.86E+01	6.46E−01	4.48E+01	1.25E+01
F3	Mean	8.61E+00	3.89E+00	1.99E+01	7.56E+00	4.35E+00	**2.14E−05**	5.58E−04	4.15E+01	1.61E+01
	Std	2.61E+01	7.90E+00	1.35E+01	7.62E+00	1.03E+01	1.01E+01	2.93E−04	7.66E+00	3.03E−01
F4	Mean	4.77E+01	3.28E+01	5.91E+01	5.59E+01	3.18E+01	**3.13E+01**	7.41E+01	5.97E+01	3.78E+01
	Std	1.20E+01	1.17E+01	1.88E+01	2.63E+01	1.57E+01	2.17E+01	1.02E+01	1.15E+01	7.59E+00
F5	Mean	7.82E+01	1.46E+01	5.67E+02	2.70E+02	2.25E+02	1.12E+00	3.19E−02	7.99E+02	**2.60E−09**
	Std	4.07E+02	1.64E+02	3.64E+02	3.59E+02	2.62E+02	3.79E+02	1.14E−01	1.58E+02	1.86E+02
F6	Mean	1.09E+04	**5.29E+01**	3.98E+02	6.49E+03	9.84E+01	1.51E+02	2.58E+05	1.66E+02	7.40E+01
	Std	1.85E+09	1.04E+03	3.87E+06	1.06E+07	3.57E+03	1.90E+06	5.40E+05	4.23E+03	7.02E+02
F7	Mean	5.31E+01	**2.93E+01**	7.92E+01	4.16E+01	3.69E+01	5.32E+01	5.38E+01	1.20E+02	3.64E+01
	Std	8.50E+01	2.73E+01	3.04E+01	3.85E+01	3.01E+01	4.82E+01	1.01E+01	7.38E+01	8.93E+00
F8	Mean	3.07E+01	2.20E+01	2.53E+01	2.52E+01	2.16E+01	2.46E+01	3.68E+01	3.93E+01	**2.04E+01**
	Std	7.98E+02	2.25E+01	3.58E+01	3.83E+01	6.53E+00	6.08E+01	2.16E+00	1.68E+02	9.83E−01
F9	Mean	1.81E+02	**1.81E+02**	1.82E+02	1.98E+02	**1.81E+02**	**1.81E+02**	1.81E+02	1.82E+02	**1.81E+02**
	Std	3.14E+02	2.34E−06	2.60E+01	2.35E+01	4.01E−05	6.40E+01	1.46E−02	3.64E+01	2.37E−06
F10	Mean	1.01E+02	1.01E+02	1.01E+02	1.01E+02	1.01E+02	1.00E+02	1.00E+02	1.01E+02	**1.00E+02**
	Std	1.49E+03	5.73E+02	6.73E+02	1.50E+03	9.54E+01	9.87E+02	1.88E−01	1.21E+03	2.52E+01
F11	Mean	3.76E+02	3.00E+02	3.25E+02	6.90E+02	**6.14E−10**	3.00E+02	3.00E+02	3.00E+02	3.73E−06
	Std	1.31E+02	4.51E+01	3.19E+02	4.41E+02	9.58E+01	5.28E+02	1.46E−02	1.10E+03	3.67E+01
F12	Mean	2.57E+02	2.48E+02	2.54E+02	2.47E+02	2.45E+02	**2.33E+02**	2.45E+02	3.84E+02	2.89E+02
	Std	4.89E+02	2.07E+01	2.98E+01	2.52E+01	3.02E+01	1.14E+02	5.69E+00	2.18E+02	4.04E+00

The optimal value under the current index is in bold

Table 12 Results of SASS, SASS-SQOBL, COLSHADE and sCMAgES in hydrostatic thrust bearing design problems

RC25	SASS-SQOBL	SASS	COLSHADE	sCMAgES
Best	**1616.119765**	1616.12	**1616.11977**	2284.5
Median	**1616.119765**	1616.12	1616.12	3174.5
Mean	**1616.12245**	1624.59	1626.09957	3022.14
Worst	**1616.163148**	1821.19	1778.05667	3530.1
Std	**0.008971372**	40.9731	33.8	387.5561

The optimal value under the current index is in bold

With bounds:

$$\begin{aligned} 1 \le R \le 16, 1 \le R_0 \le 16, \\ 1 \times 10^{-6} \le \mu \le 16 \times 10^{-6}, \\ 1 \le Q \le 16 \end{aligned} \tag{60}$$

The results of all algorithms on the design of hydrostatic thrust bearings are given in Table 12. The best fitness values obtained by SASS-SQOBL and COLSHADE are both 1616.119765. The standard deviation of SASS-SPOBL calculation results is smaller than that of other constrained optimization methods. The performance of the SASS algorithm without embedding SQOBL is lower than that of COLSHADE in terms of worst value and standard deviation, and COLSHADE has the best performance in dealing with this problem. However, the result obtained by combining SQOBL and SASS not only achieves the optimal value but also is very stable, as can be seen from the standard deviation of only 0.00897.

Optimal Sizing of Distributed Generation for Active Power Loss Minimization

Optimizing the capacity of distributed generation can effectively improve system performance and meet the needs of users, and the economic operation of distribution network. The purpose of this optimization constrained problem is to determine the size of the distributed generation in the distribution network so that its power loss is minimized.

Table 13 reports the results of all algorithms on the optimal size problem for the minimization of active losses in distributed power generation problems. It can be seen that the SASS-SPOBL algorithm is superior to other algorithms in the selection of the optimal value. Although the results

Table 13 The results of SASS, SASS-SQOBL, COLSHADE and sCMAgES in the optimal size problem of distributed power generation active loss minimization

RC42	SASS-SQOBL	SASS	COLSHADE	sCMAgES
Best	**0.0832**	0.08655	− 7.7259	0.1026
Median	**0.0873**	0.08753	− 1.6089	0.1553
Mean	**0.0874**	0.08745	− 2.613796	8.280324
Worst	0.0946	**0.08893**	− 0.9228	46.3443
Std	0.002	**0.00047**	2.21724602	15.01552

The optimal value under the current index is in bold

of COLSHADE are obviously the lowest, under the premise of practical problems, negative numbers are obviously unacceptable. Therefore, the COLSHADE column is ignored for the analysis of the results. The top three the average fitness value are 0.0873, 0.0875, and 0.1553, which are SASS-SQOBL, SASS, and sCMAgES, respectively. Considering comprehensively, SASS-SQOBL has achieved good results on this issue.

SOPWM for 13-level inverters

SOPWM for 13-level inverters Synchronous Optimal Pulse Width Modulation (SOPWM) is an analog control method. On the premise of ensuring that the distortion is not affected, the switching frequency can be significantly reduced, thereby reducing the loss of the switch. For different levels of inverters, the SOPWM problem can be expressed in the following way:

Minimize:

$$f = \frac{\sqrt{\Sigma_k \left(k^{-4}\right)\left(\Sigma_{i=1}^{N} s\left(i\right)\cos\left(k\alpha_i\right)\right)^2}}{6\sqrt{\Sigma_K k^{-4}}} \tag{61}$$

Where $k = 5, 7, 11, 13, \ldots, 97$, $N = \lfloor \frac{6, f_{s,\max}}{f,m} \rfloor$, and $s = [1, 1, 1, -1, 1, -1, 1, -1, 1, 1, 1, 1]$.

Subject to:

$$\begin{aligned} &g_i = \alpha_{i+1} - \alpha_i - 10^{-5} > 0, \\ &\quad i = 1, 2, \ldots, N-1, \\ &h_1 = 6m - \sum_{i=1}^{N} s\left(i\right)\cos\left(\alpha_i\right) = 0 \end{aligned} \tag{62}$$

With bounds:

$$0 < \alpha_i < \frac{\pi}{2},\ i = 1, 2, \ldots, N \tag{63}$$

The results of all algorithms on the SOPWM problem on a class 13 frequency converter are reported in Table 14. The four statistical values obtained by SASS-SQOBL are better than other algorithms except the worst value. Although slightly lower than SASS in standard deviation, it is better than both COLSHADE and sCMAgES and finds the optimal fitness value. Therefore, comprehensive consideration, the performance of SASS-SQOBL is better than other algorithms. This result proves that SQOBL can be attached to the optimization algorithm to speed up the search process and improve the overall performance of the algorithm.

Table 14 Results of the SOPWM problem for SASS, SASS-SQOBL, COLSHADE and sCMAgES on a class 13 frequency converter

RC50	SASS-SQOBL	SASS	COLSHADE	sCMAgES
Best	**0.014682222**	0.01519	0.0205	0.018
Median	**0.017035917**	0.02932	0.0395	0.0807
Mean	**0.023679744**	0.02789	0.065092	0.08385
Worst	0.064545237	**0.06038**	0.2387	0.1503
Std	0.012480069	**0.00981**	0.0481876	0.02638

The optimal value under the current index is in bold

Statistical analysis

In this subsection, statistical methods are applied to systematically analyze the statistical significance of all experimental results. Friedman's test was used to judge whether there was a significant difference between the performance of all algorithms. The Friedman test [86] is represented by

$$\chi_F^2 = \frac{12T}{s\left(s+1\right)}\left(\sum_{a=1}^{s} R_a^2 - \frac{s\left(s+1\right)^2}{4}\right) \tag{64}$$

$$F_F = \frac{\left(T-1\right)\chi_F^2}{T\left(s-1\right) - \chi_F^2} \tag{65}$$

In the formula, T represents the total number of test functions under different test sets, s represents the number of algorithms, and R_a represents the average ranking of the a-th algorithm on the test set. If the F distribution subject to $s-1$ and $(T-1)(s-1)$ degrees of freedom is less than χ_F^2, the null hypothesis is rejected, indicating that there are significant differences between the algorithms, then we can continue to use the Nemenyi test [87] to draw. There is a statistical difference between which algorithms perform. The critical distance of the Nemenyi test is expressed as:

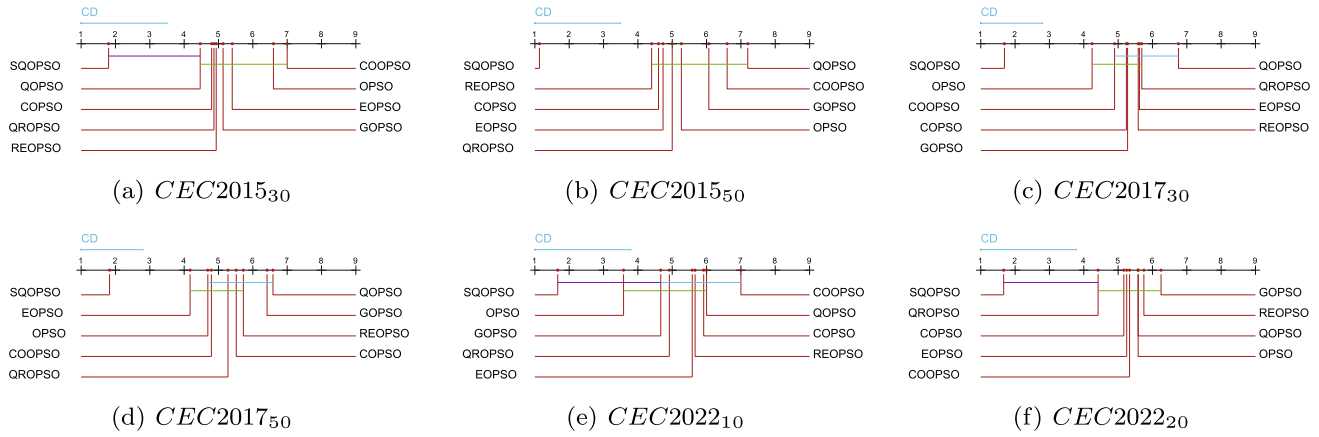
$$CD_a = q_\alpha\sqrt{\frac{s\left(s+1\right)}{6T}} \tag{66}$$

where q_α is the critical list value of the test and α is the importance of the Nemenyi test. Comparing the difference between the average rankings of each algorithm with the critical distance CD_a, if it is greater than the threshold, it means that the algorithm with a higher average ranking is statistically better than the algorithm with a lower average ranking.

Table 15 Friedman ranking of 9 algorithms in CEC2015, CEC2017 and CEC2022

Datasets	OPSO	QOPSO	QROPSO	COOPSO	COPSO	GOPSO	EOPSO	REOPSO	SQOPSO
CEC2015-30D	6.6	4.47	4.87	7	4.8	5.13	5.4	4.93	**1.8**
CEC2015-50D	5.27	7.2	5	6.6	4.6	6.07	4.73	4.4	**1.13**
CEC2017-30D	4.24	6.76	5.69	4.9	5.24	5.28	5.62	5.59	**1.69**
CEC2017-50D	4.69	6.59	5.28	4.79	5.52	6.41	4.17	5.72	**1.83**
CEC2022-10D	3.58	6	4.92	7	5.92	4.67	5.58	5.67	**1.67**
CEC2022-20D	5.58	5.58	4.42	5.33	5.17	6.25	5.25	5.75	**1.67**
Mean	4.9	6.03	5.02	5.86	5.19	5.6	5.1	5.32	**1.61**

The optimal value under the current index is in bold



(a) $CEC2015_{30}$ (b) $CEC2015_{50}$ (c) $CEC2017_{30}$

(d) $CEC2017_{50}$ (e) $CEC2022_{10}$ (f) $CEC2022_{20}$

Fig. 13 Nemenyi test results of nine OBL algorithms

By comparing the fitness values of the nine algorithms in Tables 4, 5, 6, 7, 8 and 9 in different dimensions on the CEC2015, CEC2017, and CEC2022 test sets, the Friedman rankings are shown in Table 15. When the importance is $\alpha = 0.1$, $F(8, 112) = 1.726$ is less than 5.6834 and 9.4317; $F(8, 224) = 1.698$ is less than 10.1612 and 10.4185; $F(8, 88) = 1.741$ is less than 5.4429 and 3.4922. It can be seen that there are significant differences in the performance of the nine algorithms on all test sets. If the performance of the nine algorithms is equivalent, the statistical values in Table 12 will not exceed the critical value under the importance $\alpha = 0.1$. To further obtain specific difference information, the Nemenyi follow-up test is used to distinguish each algorithm, and the corresponding CD diagram is shown in Fig. 13. The position of each algorithm on the horizontal axis in the CD graph represents the average rank. If the critical distance CD is less than the difference in mean rank, that means there is a significant difference in performance between algorithms.

As shown in Fig. 13, on the 30-dimensional CEC2015 test set, there is no significant difference in performance between SQOPSO and QOPSO, but they are significantly better than eight algorithms including COPSO, QROPSO, REOPSO, and GOPSO. On the 10-dimensional CEC2022 test set, there is no significant difference between SQOPSO and QROPSO. On the 20-dimensional CEC2022 test set, the performance of SQOPSO, OPSO, and GOPSO is similar. On the three test sets of 50-dimensional CEC2015, 30-dimensional and 50-dimensional CEC2017, there are significant differences between the SQOPSO algorithm and the other nine algorithms.

Next, the Friedman test was carried out on the fitness values of the nine new swarm optimization algorithms in Tables 10 and 11 under the CEC2022 test set. Tables 16 and 17 respectively list the average rankings of these nine methods in different dimensions, and at the same time give the number of times that different algorithms obtain the top four in multiple test functions. F_F is 9.8604 and 9.3512. After calculation, when the importance is $\alpha = 0.1$, $F(8, 88) = 1.741$ is smaller than 9.8604 and 9.3512. This shows that there are significant differences in the performance of the nine algorithms on the test set. The results of Nemenyi test are shown in Fig. 14a, b, respectively. It can be seen from Fig. 14 that on the 10-dimensional CEC2022 test set, NGO, GNDO, TSO, SQOPSO, DMO, and SCSO have little difference in performance, but are significantly better than SSA, GJO, and PSO. On the 20-dimensional CEC2022 test set, SQOPSO, TSO, NGO, GNDO and DMO perform similarly, but far outperform PSO, SCSO, GJO and SSA.

The above experimental analysis results show that the SQOPSO algorithm is significantly better than other algo-

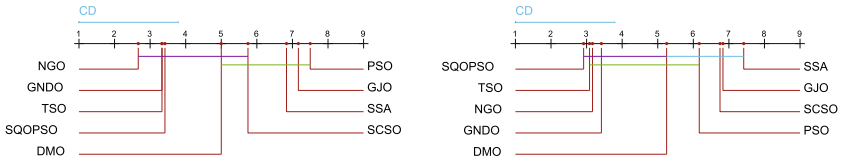


Fig. 14 Nemenyi test results of 9 algorithms

(a) $CEC2022_{10}$ (b) $CEC2022_{20}$

Table 16 Ranking of SQOPSO algorithm and other 8 algorithms on 10-dimensional CEC2022

Algorithms	1st	2nd	3rd	4th	Rank
SQOPSO	5	4	0	1	3.42
GNDO	3	0	4	2	3.33
SCSO	0	0	0	1	5.75
GJO	0	0	0	1	7.17
TSO	1	3	5	1	3.33
NGO	2	3	2	3	2.67
DMO	1	2	1	2	5
SSA	0	0	0	0	6.83
PSO	0	0	0	1	7.5

Table 17 Ranking of SQOPSO algorithm and other 8 algorithms on 20-dimensional CEC2022

Algorithms	1st	2nd	3rd	4th	Rank
SQOPSO	3	5	0	2	2.92
GNDO	4	0	3	1	3.42
SCSO	0	0	0	0	6.75
GJO	1	0	0	2	6.83
TSO	1	3	4	1	3.08
NGO	3	1	5	1	3.17
DMO	0	3	0	3	5.25
SSA	0	0	0	2	7.42
PSO	0	0	0	0	6.17

rithms in dealing with test sets of different dimensions. In summary, through the Friedman test and subsequent tests, our algorithm can achieve better results than other algorithms.

Conclusion and future work

Inspired by the principle of refraction imaging, this paper constructs a new adversarial learning strategy and embeds it into an intelligent optimization algorithm to further improve exploration and development capability. The mathematical properties of density are used to redefine the value center point in the OBL strategy and accelerate the convergence speed at the beginning of particle convergence. When diversity falls below a set threshold, this means that the current level of particle exploration is inadequate, and reverse strategies have been changed to improve population diversity and avoid local extremes and premature maturation. Theoretical analysis and experimental results show that the SQOBL learning strategy makes full use of search information in the population iteration process and has strong convergence. Population diversity is also increasing. Compared with other opposite learning particle swarm optimization algorithms and various advanced intelligent optimization algorithms, this algorithm has high convergence accuracy. As an intelligent technology, the SQOBL learning strategy can be embedded in an intelligent optimization algorithm, which is helpful to improve the performance of the algorithm. Statistical test results also prove that the SQOBL learning strategy has advantages in dealing with high dimensional optimization problems.

In the future, we will further optimize the strategy, focus of the adaptive operation of parameters, automatically select the optimal value and improve the performance of the algorithm. Multi-objective optimization is a very challenging research direction, and combining SQOBL as an intelligent technology with other algorithms in the multi-objective domain has some research value. Common OBL variants are usually calculated based on the uniform distribution of corresponding individuals. If you add other probability distributions to your SQOBL strategy, this is also an improvement worth considering.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant (61976082, 62076089, 62002103).

Data Availability All data is available from https://github.com/ZooooZoooo/SQOBL

Appendix

The adjustable parameters in the experiment, as well as the specific values and descriptions of the parameters are presented in Table 18 of the Appendix.

Table 18 Algorithm parameter settings

Parameter category	Parameters	Parameter description	Parameter value
Public parameters	N	Population size	100
	FE_max	Maximum number of function evaluations	100000
	c_1, c_2	Cognitive factor	2
	D	Problem dimension	30, 50 (CEC2015, 2017)
			10,20(CEC2022)
OBL variant parameters	JR	Jump rate	0.05, 0.3
Optimization algorithm parameters	ST	Early warning value	0.6 [5]
	PD	Proportion of discoverers	0.7 [5]
	SD	Awareness of the weight of dangerous sparrows	0.2 [5]
	bs	Number of babysitters	3 [6]
	$peep$	Alpha female vocalization	2 [6]
	a	The degree of following the optimal value	0.7 [8]
	c_1, β	Constants	1.5 [9]
	S_M	Auditory characteristics of sand cats	2 [10]
SQOBL parameters	T	Diversity threshold	0.1
	θ	Angle between chopsticks and water surface	30°
	k	Initial stage scaling factor	0.75
	n	Refractive index	4/3

References

1. Phan T, Sell D, Wang E-W, Doshay S, Edee K, Yang J, Fan J-A (2019) High-efficiency, large-area, topology-optimized metasurfaces. light-sci appl 8(1):1-9. https://doi.org/10.1038/s41377-019-0159-5
2. Berger K, Rivera Caicedo J-P, Martino L, Wocher M, Hank T, Verrelst J (2021) A survey of active learning for quantifying vegetation traits from terrestrial earth observation data. Remote Sens-basel 13(2):287. https://doi.org/10.3390/rs13020287
3. Lee C-C, Hussain J, Chen Y (2022) The optimal behavior of renewable energy resources and government's energy consumption subsidy design from the perspective of green technology implementation. Renew Energ 195:670–680. https://doi.org/10.1016/j.renene.2022.06.070
4. Zamir M, Abdeljawad T, Nadeem F, Wahid A, Yousef A (2021) An optimal control analysis of a COVID-19 model. Alex End J 60(3):2875–2884. https://doi.org/10.1016/j.aej.2021.01.022
5. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8(1):22-34. https://doi.org/10.1080/21642583.2019.1708830
6. Agushaka J-O, Ezugwu A-E, Abualigah L (2022) Dwarf mongoose optimization algorithm. Comput Method Appl M 391:114570. https://doi.org/10.1016/j.cma.2022.114570
7. Dehghani M, Hubálovský Š, Trojovský P (2021) Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems. IEEE Access 9:162059–162080. https://doi.org/10.1109/access.2021.3133286
8. Xie L, Han T, Zhou H, Zhang Z-R, Han B, Tang A (2021) Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. Comput Intel Neurosc 2021. https://doi.org/10.1155/2021/9210050
9. Chopra N, Ansari M-M (2022) Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. Expert Syst Appl 198:116924. https://doi.org/10.1016/j.eswa.2022.116924
10. Seyyedabbasi A, Kiani F (2022) Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. Eng Comput-Germany 1–25. https://doi.org/10.1007/s00366-022-01604-x
11. Zhang Y, Jin Z, Mirjalili S (2020) Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models. Energy Conversion and Management Energ Convers Manage 224:113301. https://doi.org/10.1016/j.enconman.2020.113301
12. De Jong K. (1988) Learning with genetic algorithms: An overview. Machine learning 3:121-138. https://doi.org/10.1007/bf00113894
13. Storn, R, Price, K (1997) Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. Journal of Global Optimization 11:341-359. https://doi.org/10.1023/A:1008202821328
14. D. Simon (2008) Biogeography-Based Optimization. in IEEE Transactions on Evolutionary Computation 12(6):702-713. https://doi.org/10.1109/TEVC.2008.919004
15. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. science 220(4598):671-680. https://doi.org/10.1117/12.256316
16. Erol O K, Eksin I (2016) A new optimization method: big bang-big crunch. Advances in Engineering Software 37(2):106-111. https://doi.org/10.1016/j.advengsoft.2005.04.005
17. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Information sciences 179(13):2232-2248. https://doi.org/10.1007/978-3-319-31683-3_6
18. Hatamlou A (2013) Black hole: A new heuristic optimization approach for data clustering. Information sciences 222:175-184. https://doi.org/10.1016/j.ins.2012.08.023
19. Rahman CM, Rashid TA (2021) A new evolutionary algorithm: Learner performance based behavior algorithm. Egyptian Informatics Journal 22(2):213-223. https://doi.org/10.1016/j.eij.2020.08.003
20. Abdulhameed S, Rashid TA(2022) Child drawing development optimization algorithm based on child's cognitive development.

Arabian Journal for Science and Engineering 47(2):1337-1351. https://doi.org/10.1007/s13369-021-05928-6
21. Kumar M, Kulkarni AJ, Satapathy SC (2018) Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology Future Generation Computer Systems 81:252-272. https://doi.org/10.1016/j.future.2017.10.052
22. Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. Applied Soft Computing, 2018, 64:161-185. https://doi.org/10.1016/j.asoc.2017.11.043
23. Kennedy J, Eberhart R (1995) Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks 4:1942-1948. https://doi.org/10.1109/ICNN.1995.488968
24. Mirjalili S, Mirjalili S M (2014) Lewis A. Grey wolf optimizer. Advances in engineering software 69:46-61. https://doi.org/10.1109/confluence51648.2021.9377194
25. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Advances in engineering software 95:51-67. https://doi.org/10.1007/s41870-019-00346-2
26. Połap D, Woźniak M (2021) Red fox optimization algorithm. Expert Systems with Applications 166:114107. https://doi.org/10.1016/j.eswa.2020.114107
27. Hama Rashid D N, Rashid T A, Mirjalili S (2021) ANA: Ant Nesting Algorithm for Optimizing Real-World Problems. Mathematics 9(23):3111. https://doi.org/10.3390/math9233111
28. Abdullah J M, Ahmed T(2019) Fitness dependent optimizer: inspired by the bee swarming reproductive process. IEEE Access 7:43473-43486. https://doi.org/10.1109/access.2019.2907012
29. Jain M, Saihjpal V, Singh N (2022) An Overview of Variants and Advancements of PSO Algorithm. Applied Sciences 12(17):8392. https://doi.org/10.3390/app12178392
30. Dong Y, Tang J, Xu B, Wang D (2005) An application of swarm optimization to nonlinear programming. Comput Mathappl 49(11-12):1655-1668. https://doi.org/10.1016/j.camwa.2005.02.006
31. Shami T M, El-Saleh A A, Alswaitti M (2022) Particle swarm optimization: A comprehensive survey. IEEE Access 10:10031-10061. https://doi.org/10.1109/access.2022.3142859
32. Morales-Castañeda B, Zaldivar D, Cuevas E, Fausto F, Rodríguez A (2020) A better balance in metaheuristic algorithms: Does it exist? Swarm Evol Comput 54:100671. https://doi.org/10.1016/j.swevo.2020.100671
33. Taherkhani M, Safabakhsh R (2016) A novel stability-based adaptive inertia weight for particle swarm optimization. Appl Soft Comput 38:281–295. https://doi.org/10.1016/j.asoc.2015.10.004
34. Ding S, Zhang Z, Sun Y, Shi S (2022) Multiple birth support vector machine based on dynamic quantum particle swarm optimization algorithm. Neurocomputing 480:146–156. https://doi.org/10.1016/j.neucom.2022.01.012
35. Liu W, Wang Z, Yuan Y, Zeng N, Hone K, Liu X (2019) A novel sigmoid-function-based adaptive weighted particle swarm optimizer. IEEE T cybernetics 51(2):1085–1093. https://doi.org/10.1109/tcyb.2019.2925015
36. Ding S, Du W, Zhao X, Wang L, Jia W (2019) A new asynchronous reinforcement learning algorithm based on improved parallel PSO. Appl Intell 49(12):4211–4222. https://doi.org/10.1007/s10489-019-01487-4
37. Liu Q, Wei W, Yuan H, Zhan Z-H, Li Y (2016) Topology selection for particle swarm optimization. Inform Sciences 363:154–173. https://doi.org/10.1016/j.ins.2016.04.050
38. Lin A, Sun W, Yu H, Wu G, Tang H (2019) Global genetic learning particle swarm optimization with diversity enhancement by ring topology. Swarm Evol Comput 44:571–583. https://doi.org/10.1016/j.swevo.2018.07.002
39. Ardizzon G, Cavazzini G, Pavesi G (2015) Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms. Inform Sciences 299:337–378. https://doi.org/10.1016/j.ins.2014.12.024
40. Xing Z, Zhu J, Zhang Z, Qin Y, Jia L (2022) Energy consumption optimization of tramway operation based on improved PSO algorithm. Energy 258:124848. https://doi.org/10.1016/j.energy.2022.124848
41. Nobile M-S, Cazzaniga P, Besozzi D, Colombo R, Mauri G, Pasi G (2018) Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. Swarm evol comput 39:70–85. https://doi.org/10.1016/j.swevo.2017.09.001
42. Li T, Shi J, Deng W, Hu Z (2022) Pyramid particle swarm optimization with novel strategies of competition and cooperation. Appl Soft Comput 121:108731. https://doi.org/10.1016/j.asoc.2022.108731
43. Zhang W, Li G, Zhang W, Liang J, Yen G-G (2019) A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization. Swarm Evol Comput 50:100569. https://doi.org/10.1016/j.swevo.2019.100569
44. Aziz N-A, Ibrahim Z, Mubin M, Nawawi S-W, Mohamad M-S (2018) Improving particle swarm optimization via adaptive switching asynchronous-synchronous update. Appl Soft Comput 72:298–311. https://doi.org/10.1016/j.asoc.2018.07.047
45. Roy C, Das D-K (2021) A hybrid genetic algorithm (GA)-particle swarm optimization (PSO) algorithm for demand side management in smart grid considering wind power for cost optimization. Sādhanā 46(2):1–12. https://doi.org/10.1007/s12046-021-01626-z
46. Tanweer M-R, Suresh S, Sundararajan N (2016) Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. Inform Sciences 326:1–24. https://doi.org/10.1016/j.ins.2015.07.035
47. Chegini S-N, Bagheri A, Najafi F (2018) PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. Appl Soft Comput 73:697–726. https://doi.org/10.1016/j.asoc.2018.09.019
48. Kang Q, Xiong C, Zhou M, Meng L (2018) Opposition-based hybrid strategy for particle swarm optimization in noisy environments. IEEE Access 6:21888–21900. https://doi.org/10.1109/ACCESS.2018.2809457
49. Tizhoosh H-R (2005) Opposition-based learning: a new scheme for machine intelligence. In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06) 1:695-701. https://doi.org/10.1109/CIMCA.2005.1631345
50. Zhou M, Zhao Z, Xiong C, Kang Q (2018) An opposition-based particle swarm optimization algorithm for noisy environments. 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC) 1-6. https://doi.org/10.1109/ICNSC.2018.8361279
51. Ul Hassan N, Bangyal W-H, Ali Khan M-S, Nisar K, Ibrahim A-A, Rawat D-B (2021) Improved Opposition-Based Particle Swarm Optimization Algorithm for Global Optimization. Symmetry 13(12):2280. https://doi.org/10.3390/sym13122280
52. Han L, He X (2007) A novel opposition-based particle swarm optimization for noisy problems. In Third International Conference on Natural Computation (ICNC 2007) 3:624-629. https://doi.org/10.1109/ICNC.2007.119
53. Wang H, Wu Z, Rahnamayan S, Liu Y, Ventresca M (2011) Enhancing particle swarm optimization using generalized opposition-based learning. Inform Sciences 181(20):4699–4714. https://doi.org/10.1016/j.ins.2011.03.016
54. Dong W, Kang L, Zhang W (2017) Opposition-based particle swarm optimization with adaptive mutation strategy. Soft Comput 21(17):5081–5090. https://doi.org/10.1007/s00500-016-2102-5
55. Tang J (2009) Zhao X (2009) An enhanced opposition-based particle swarm optimization. WRI Global Congress on Intelligent Systems 1:149–153. https://doi.org/10.1109/GCIS.2009.56

56. Shao P, Wu Z, Zhou X, Tran D-C (2017) FIR digital filter design using improved particle swarm optimization based on refraction principle. Soft Comput 21(10):2631–2642. https://doi.org/10.1007/s00500-015-1963-3
57. Li J, Gao Y, Wang K, Sun Y (2021) A dual opposition-based learning for differential evolution with protective mechanism for engineering optimization problems. Appl Soft Comput 113:107942. https://doi.org/10.1016/j.asoc.2021.107942
58. Rahnamayan S, Jesuthasan J, Bourennani., Salehinejad H, Naterer G-F, (2014) Computing opposition by involving entire population. IEEE congress on evolutionary computation (CEC) 2014:1800–1807. https://doi.org/10.1109/CEC.2014.6900329
59. Liu H, Xu G, Ding G (2015) Integrating opposition-based learning into the evolution equation of bare-bones particle swarm optimization. Soft Computing 19:2813–2836. https://doi.org/10.1007/s00500-014-1444-0
60. Tizhoosh H-R, Ventresca M (2008) Oppositional concepts in computational intelligence. Springer. https://doi.org/10.1007/978-3-540-70829-2
61. Seif Z, Ahmadi M-B (2015) An opposition-based algorithm for function optimization. Eng Applartif Intel 37:293–306. https://doi.org/10.1016/j.engappai.2014.09.009
62. Ergezer M, Simon D, Du D (2009) Oppositional biogeography-based optimization. 2009 IEEE international conference on systems, man and cybernetics 1009-1014. https://doi.org/10.1109/ICSMC.2009.5346043
63. Rahnamayan S, Tizhoosh H-R, Salama M-M (2007) Quasi-oppositional differential evolution. IEEE congress on evolutionary computation 2007:2229–2236. https://doi.org/10.1109/CEC.2007.4424748
64. Dong W, Kang L, Zhang W (2017) Opposition-based particle swarm optimization with adaptive mutation strategy. Soft Computing 21:5081–5090. https://doi.org/10.1007/s00500-016-2102-5
65. Rahnamayan S, Tizhoosh H-R, Salama M-M (2008) Opposition-based differential evolution. IEEE T Evolut Comput 12(1):64–79. https://doi.org/10.1007/978-3-540-68830-3_6
66. Rahnamayan S, Jesuthasan J, Bourennani ., Salehinejad H, Naterer G-F (2014) Computing opposition by involving entire population. 2014 IEEE congress on evolutionary computation (CEC) 1800-1807. https://doi.org/10.1109/CEC.2014.6900329
67. Zhao X, Feng S, Hao J (2021) Neighborhood opposition-based differential evolution with Gaussian perturbation. Soft Computing 25:27-46. https://doi.org/10.1007/s00500-020-05425-2
68. Si T, De A, Bhattacharjee AK (2014) Particle swarm optimization with generalized opposition based learning in particle's pbest position. Power and Computing Technologies 2014:1662-1667. https://doi.org/10.1109/iccpct.2014.7055039
69. Ahandani MA, Abbasfam J, Kharrati H (2022) Parameter identification of permanent magnet synchronous motors using quasi-opposition-based particle swarm optimization and hybrid chaotic particle swarm optimization algorithms. Applied Intelligence 52(11):13082-13096. https://doi.org/10.1007/s10489-022-03223-x
70. Khan R-A, Yang S, Khan S, Fahad S (2021) A Multimodal Improved Particle Swarm Optimization for High Dimensional Problems in Electromagnetic Devices. Energies 14(24):8575. https://doi.org/10.3390/en14248575
71. Cheng S, Shi Y, Qin Q, Zhang Q, Bai R (2014) Population diversity maintenance in brain storm optimization algorithm. Jartif Intell Soft 4(2):83–97. https://doi.org/10.1515/jaiscr-2015-0001
72. Huang Y, Li J-P, Wang P (2019) Unusual phenomenon of optimizing the Griewank function with the increase of dimension. Front Inform Techn El 20(10):1344–1360. https://doi.org/10.1631/fitee.1900155
73. Clerc M (2010) Particle swarm optimization. John Wiley & Sons. https://doi.org/10.1088/1674-1056/ac65ee
74. Shifeng O, Ying G, Gang J, Xuehui Z (2009) Variable step size algorithm for blind source separation using a combination of two adaptive separation systems. 2009 Fifth International Conference on Natural Computation 3:649-652. https://doi.org/10.1109/ICNC.2009.544
75. Dharmani B (2022) Gram-Charlier A Series Based Extended Rule-of-Thumb for Bandwidth Selection in Univariate Kernel Density Estimation. Aust J stat 51(3):141-163. https://doi.org/10.17713/ajs.v51i3.1204
76. Silverman B (1986) DensityEstimation Chapman and Hall. London
77. Rashed R (1990) A pioneer in anaclastics: Ibn Sahl on burning mirrors and lenses. Isis 81(3):464-491. https://doi.org/10.1086/355456
78. Hussain K, Salleh M-M, Cheng S, Shi Y (2019) On the exploration and exploitation in popular swarm-based metaheuristic algorithms. Neural Computappl 31(11):7665–7683. https://doi.org/10.1007/s00521-018-3592-0
79. Wang H, Wu Z, Liu Y, Wang J, Jiang D, Chen L (2009) Space transformation search: a new evolutionary technique. In Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation 537–544. https://doi.org/10.1145/1543834.1543907
80. Liang J-J, Qu B-Y, Suganthan P-N, Chen Q (2014) Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Technical Report201411A. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 29:625–640
81. Wu G, Mallipeddi R, Suganthan P-N (2017) Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report
82. Kumar A, Wu G, Ali MZ (2020) A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. Swarm and Evolutionary Computation 56:100693. https://doi.org/10.1016/j.swevo.2020.100693
83. Kumar A, Das S, Zelinka I (2020) A self-adaptive spherical search algorithm for real-world constrained optimization problems. Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion 2020:13-14. https://doi.org/10.1145/3377929.3398186
84. Gurrola-Ramos J, Hernàndez-Aguirre A (2020) Dalmau-Cedeño O (2020) COLSHADE for real-world single-objective constrained optimization problems. IEEE congress on evolutionary computation (CEC) 2020:1–8. https://doi.org/10.1109/cec48606.2020.9185583
85. Kumar A, Das S, Zelinka I(2020) A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion 2020:11-12. https://doi.org/10.1145/3377929.3398185
86. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Inst Stat Math 11(1):86–92
87. Dunn QJ (1961) Multiple comparisons among means. J Am Stat Assoc 56(293):52–64