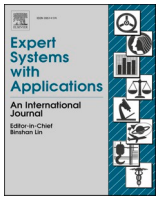
Expert Systems With Applications 232 (2023) 120886

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



PID-based search algorithm: A novel metaheuristic algorithm based on PID algorithm

Yuansheng Gao *

College of Science, Liaoning Technical University, Fuxin 123000, China

ARTICLE INFO

Keywords:
Metaheuristic
Global optimization
PID algorithm
PID-based search algorithm
Constrained problems

ABSTRACT

In this paper, a metaheuristic algorithm called PID-based search algorithm (PSA) is proposed for global optimization. The algorithm is based on an incremental PID algorithm that converges the entire population to an optimal state by continuously adjusting the system deviations. PSA is mathematically modeled and implemented to achieve optimization in a wide range of search spaces. PSA is used to solve CEC2017 benchmark test functions and six constrained problems. The optimization performance of PSA is verified by comparing it with seven metaheuristics proposed in recent years. The Kruskal-Wallis, Holm and Friedman tests verified the superiority of PSA in terms of statistical significance. The results show that PSA can be better balanced exploration and exploitation with strong optimization capability. Source codes of PSA are publicly available at https://ww2.mathworks.cn/matlabcentral/fileexchange/131534-pid-based-search-algorithm.

1. Introduction

With the development of science and technology, the complexity and difficulty of solving optimization problems have increased substantially. These optimization problems are usually non-convex, complex, and computationally expensive (Ahmadianfar et al., 2020). Traditional optimization methods are no longer good enough to solve these complex problems. However, in the past decade or so, metaheuristic algorithms have evolved rapidly to provide powerful optimization techniques for solving these complex real-world problems.

Most of the inspiration for metaheuristic algorithms comes from nature. Many phenomena in nature demonstrate intelligence, e.g., a colony of ants can usually find a shorter path when searching for food. It is by simulating these phenomena that the metaheuristic algorithm solves the problem. Compared with traditional optimization methods, metaheuristic algorithms have the advantages of flexibility, simplicity, and gradient-free (Dhiman & Kumar, 2019). In solving the problem, the metaheuristic algorithm does not need to solve the gradient of the problem, but only needs to calculate the value of the problem under some feasible solution, which greatly reduces the computational complexity of the metaheuristic algorithm. Moreover, when solving a different problem, only the function to be solved needs to be modified, not much modification to the core part of the algorithm is required. This simplicity and flexibility have led to metaheuristic algorithms being used in a wide range of applications in several fields (Bahreininejad, 2019).

Some of the metaheuristic algorithms that have been proposed are given in Fig. 1. Different metaheuristics model different phenomena or behaviors and behave differently on different problems. It is also because of their different performance that they are suitable for solving different optimization problems (Azizi et al., 2022). According to the no free lunch theorem (Wolpert & Macready, 1997), no single metaheuristic algorithm is suitable for solving all optimization problems. This indicates that for a given algorithm, it works better when solving a certain problem, but may work less well when solving another class of problems. Therefore, it is necessary to develop different metaheuristic algorithms to fill the gap in solving a particular problem. However, some metaheuristic algorithms exist that are based on metaphors. These metaphor-based algorithms do not facilitate the development of metaheuristics, and even hinder it. Therefore, it is more necessary to propose some metaheuristic algorithms that go beyond metaphors to provide a greater contribution to the optimization process.

PID algorithm is a classical algorithm in the field of control and is divided into incremental PID control and position PID control. In process control, the PID controller, which controls by proportional (P), integral (I) and differential (D) of deviation, is one of the most widely used automatic controllers (Wan et al., 2020). By reviewing the previous metaheuristics, it was found that there was no metaheuristic algorithm

* Corresponding author.
E-mail address: gaoyuansheng2021@163.com.

https://doi.org/10.1016/j.eswa.2023.120886
Received 12 December 2022; Received in revised form 6 June 2023; Accepted 18 June 2023
Available online 23 June 2023

proposed inspired by the PID algorithm. Therefore, this paper attempts to develop a metaheuristic algorithm based on incremental PID control, called PID-based search algorithm (PSA). The PSA was benchmarked by 29 benchmarking functions of CEC2017. Moreover, PSA was applied to six constrained optimization problems to verify the capability of PSA to solve constrained problems.

The structural framework following this paper is: In Section 2, some past studies on metaheuristic algorithms are introduced. Section 3 presents the inspiration for the proposed PSA. Section 4 details the mathematical model, pseudocode and computational complexity analysis of the PSA. In Section 5, the capabilities of PSA to solve benchmark problems are analyzed and discussed. PSA is applied to five engineering problems and verified the feasibility for solving engineering problems in Section 6. Finally, Section 7 summarizes and gives an outlook on the work done in this paper.

2. Related studies

Optimization is a popular research area, and hundreds of metaheuristic algorithms have been proposed. Although there are various metaheuristic algorithms available, they all follow a similar design (as in Algorithm 1). Metaheuristics usually set some necessary parameters in the initialization phase. After that, the initial position of each individual is generated randomly throughout the search space. This random approach leads to a large randomness and uncertainty of the initial individuals, which affects the optimization results. Therefore, many improved methods for initial populations have been investigated by related scholars, including chaotic mapping (Varol Altay & Alatas, 2020), reverse learning method (Li et al., 2018), Lévy flight (Viswanathan et al., 2000), and so on.

After the position of each individual is determined, the corresponding objective function value is calculated. Once this is done, the iterative optimization process begins. As shown in Algorithm 1, the iterative process usually has three steps. Among them, steps 1 and 3 are similar in different metaheuristic algorithms. Different metaheuristic algorithms are different in their search operations (step 2 in Algorithm 1), which is the most important reason for different optimization results. In step 1, the historical best individual of the population or the historical best individuals of each individual are usually selected to guide the search. In step 2, different types of algorithms have different search operations. According to the different search operations, this paper classifies metaheuristic algorithms into four categories: evolution-based algorithms, swarm-based algorithms, physics-based algorithms, and mathematics-based algorithms.

Evolution-based algorithms usually simulate the genetic evolution of organisms and the process of meritocracy in nature to perform search operations. Among these algorithms, the most classic one is the genetic algorithm (GA) (Holland, 1992). Genetic algorithm uses binary coding to generate new populations through operations such as selection, crossover, and mutation. After the initial population is generated, it evolves generation by generation to produce increasingly better approximate solutions according to the principles of survival of the fittest and survival of the fittest. Selection, crossover, and mutation are the search operations of the genetic algorithm. Other algorithms of this type use similar operations to implement the search, only simulating a different concept of evolution. In addition to genetic algorism, such algorithms include evolutionary programming (EP) (Yao et al., 1999), evolutionary strategy (ES) (Rechenberg, 1973), modal algorithm (MA) (Moscato, 1989), genetic programming (GP) (Koza, 1994), differential evolution (DE) (Storn & Price, 1997), backtracking search optimization algorithm (BSA) (Civicioglu, 2013), monkey king evolution (MKE) (Meng & Pan, 2016), wildebeests herd optimization (WHO) (Motevali et al., 2019), and human felicity algorithm (HFA) (Veysari, 2022).

Algorithm 1 General steps of the search process in metaheuristic algorithms

1. Begin (initialization)
2. P: Create the P-population randomly
3. **for** $i = 1$: n (the number of solution candidates)
4. F: Use the objective function and create the fitness vector of the members in P
5. **end**
6. **while** (search process lifecycle)
7. % *Step 1: Guide selection mechanism (create a mating pool)*
8. Selection of reference positions from the P
9. % *Step 2: Search operations*
10. Exploitation (neighborhood search around reference positions)
11. Exploration (diversification operations in P)
12. % *Step 3: Update mechanism*
13. Update the P-population depending on the fitness values or NSM scores (Kahraman, Katı, Aras, & Taşci, 2023) of solution candidates
14. next generation until termination criterion
15. **end**

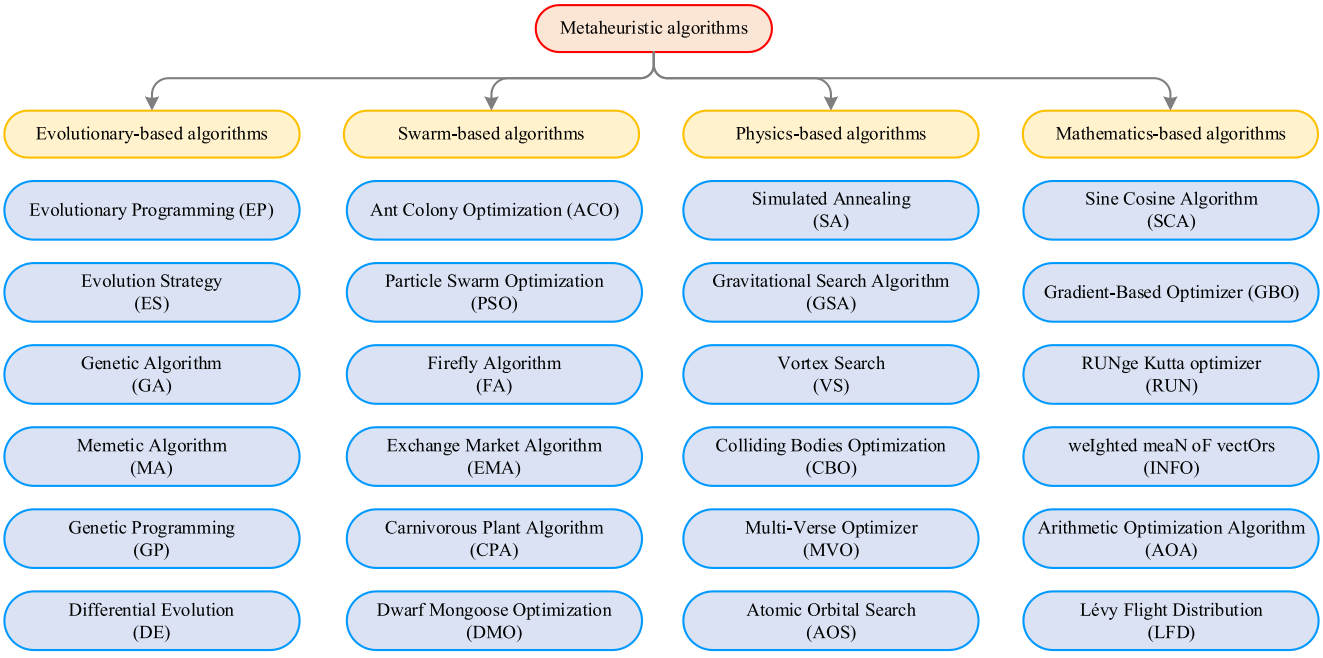


Fig. 1. Some of the metaheuristic algorithms that have been proposed.

Most swarm-based algorithms simulate the cluster behavior of organisms to perform search operations. These cluster behaviors are usually capable of exhibiting intelligence. For example, ant colony optimization (ACO) (Dorigo et al., 1996) simulates the behavior of ants forming the shortest path during foraging. The search operation of ACO is reflected in the simulation of ant foraging. During the foraging process, ants release a pheromone. For the same amount of time, more ants will travel the path with the shorter distance. Furthermore, the ants tend to choose the path with large pheromone concentration. Over time, this positive feedback causes the ant colony to eventually find a path with the shortest distance. Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) simulates the foraging behavior of birds to perform the search and uses both individual and population information to guide the search. This type of algorithm usually searches for the optimal solution through collaboration and information sharing among individuals in a population. In addition to these two algorithms, swarm-based algorithms include society and civilization algorithm (SCA) (Ray & Liew, 2003), artificial bee colony (ABC) (Karaboga & Basturk, 2007), imperialist competitive algorithm (ICA) (Atashpaz-Gargari and Lucas, 2007), firefly algorithm (FA) (Yang, 2009), bat algorithm (BA) (Yang, 2010), water cycle algorithm (WCA) (Eskandar et al., 2012), mine blast algorithm (MBA) (Sadollah et al., 2013), grey wolf optimizer (GWO) (Mirjalili et al., 2014), exchange market algorithm (EMA) (Ghorbani & Babaei, 2014), grasshopper optimisation algorithm (GOA) (Saremi et al., 2017), butterfly optimization algorithm (BOA) (Arora & Singh, 2019), poor and rich optimization algorithm (PRO) (Moosavi & Bardsiri, 2019), Harris hawk optimization (HHO) (Heidari et al., 2019), carnivorous plant algorithm (CPA) (Ong et al., 2021), social network search (SNS) (Talatahari et al., 2021), mayfly algorithm (MA) (Zervoudakis & Tsafarakis, 2020), tunicate swarm algorithm (TSA) (Kaur et al., 2020), northern goshawk optimization (NGO) (Dehghani et al., 2021), Coot algorithm (COOT) (Naruei & Keynia, 2021), aquila optimizer (AO) (Abualigah et al., 2021), African vultures optimization algorithm (AVOA) (Abdollahzadeh et al., 2021), escaping bird search (EBS) (Shahrouzi & Kaveh, 2022), golden jackal optimization (GJO) (Chopra & Ansari, 2022), Ali Baba and the forty thieves (AFT) (Braik et al., 2022), white shark optimizer (WSO) (Braik et al., 2022), and Flying Foxes Optimization (FFO) (Zervoudakis & Tsafarakis, 2022).

Physics-based algorithms usually use the concepts of electromagnetic, gravitational, and inertial forces in physics to perform spatial searches. Simulated annealing (SA) (Kirkpatrick et al., 1983) is the most classical algorithm in this category. The algorithm is derived from the principle of solid annealing and searches by simulating the process of cooling metals from high temperatures in life. The algorithm has a certain probability of accepting a solution that is worse than the current one, so it can jump out of the local optimum to some extent. In addition to simulated annealing algorithms, such algorithms include gravitational search algorithm (GSA) (Rashedi et al., 2009), black hole algorithm (BH) (Hatamlou, 2013), vortex search algorithm (VS) (Doğan & Ölmez, 2015), colliding bodies optimization (CBO) (Kaveh & Mahdavi, 2014), multi-verse optimizer (MVO) (Mirjalili et al., 2016), Henry gas solubility optimization (HGSO) (Hashim et al., 2019), thermal exchange optimization (TEO) (Kaveh & Dadras, 2017), and atomic orbital search (AOS) (Azizi, 2021).

Mathematics-based algorithms are a younger branch of metaheuristics, which do not search by modeling a certain phenomenon or behavior like other types of algorithms, but simply based on some mathematical formula. For example, sine cosine algorithm (SCA) (Mirjalili, 2016) makes each individual fluctuate outward or in the direction of the optimal solution based on the mathematical model of sine and cosine, thus performing a spatial search. These algorithms also include gradient-based optimizer (GBO) (Ahmadianfar et al., 2020), Runge Kutta optimizer (RUN) (Ahmadianfar et al., 2021), arithmetic optimization algorithm (AOA) (Abualigah et al., 2021), weighted mean of vectors (INFO) (Ahmadianfar et al., 2022), and Lévy flight distribution (LFD) (Houssein et al., 2020).

After the search is completed, different metaheuristic algorithms generally retain the individual with the better objective function value to complete the third step in Algorithm 1. From the present point of view, not more algorithms have been proposed due to the late start of the research on mathematics-based metaheuristics. However, this does not reduce the contribution of mathematics-based metaheuristic algorithms to the optimization process, since such algorithms are usually beyond metaphors due to the difference in mathematical formulations.

3. Inspiration

Incremental PID control makes the difference between the current moment's control quantity and the previous moment's control quantity, and the difference is the new control quantity, which is a recursive algorithm (Shen et al., 2009). Incremental PID control is a proportional-integral-derivative control, and its regulation process is shown in Fig. 2. After the user sets a target value, the incremental PID controller gives an output value to the actuator, which regulates the controlled object according to the output value. The real value of the controlled object after the adjustment is acquired by a sensor and passed again to the incremental PID controller.

The entire derivation process of the PID algorithm is shown in Appendix A. To accommodate the search mechanism of the metaheuristic algorithm, this study uses a discretized incremental PID algorithm. Assuming that the sampling is periodic, the sampling period is T, and the discrete independent variable is t, the discrete PID control can be expressed as

$$u(t) = K_p\left\{e(t) + \frac{T}{T_i}\sum_{i=0}^{t} e(i) + \frac{T_d}{T}[e(t) - e(t-1)]\right\} \tag{1}$$

where $u(t)$ is the control quantity at moment t, K_p denotes the proportionality factor, T_i shows the integral time constant, T_d represents the differential time constant, and $e(t)$ indicates the deviation between the user set value and the true value at moment t, i.e., the system deviation. Then the control volume at moment $t-1$ is

$$u(t-1) = K_p\left\{e(t-1) + \frac{T}{T_i}\sum_{i=0}^{t-1} e(i) + \frac{T_d}{T}[e(t-1) - e(t-2)]\right\} \tag{2}$$

Suppose the change in the control quantity $\Delta u(t)$ at moment t is $u(t) - u(t-1)$, then

$$\Delta u(t) = K_p[e(t) - e(t-1)] + K_i e(t) + K_d[e(t) - 2e(t-1) + e(t-2)] \tag{3}$$

where K_i is the integration coefficient and K_d refers the differentiation coefficient.

PID control is designed to effectively correct the deviation of the controlled object so that the system achieves a stable state. In this study, the historical best individual of the population is abstracted as the target value, and each individual is abstracted as the true value, then the PSA proposed in this paper is to simulate the regulation process of PID to correct the deviation of each individual relative to the best individual, so that the whole population can reach a better state.

4. PID-based search algorithm

4.1. Initialization

An optimization problem consists of a set of decision variables, constraints, and an objective function. It may be assumed that the number of decision variables in this group is d and the upper and lower bounds of the variables are $\mathbf{u}$ and $\mathbf{l}$, respectively. The control parameters of PSA include the maximum number of iterations T and the population size n. Then the initial population can be expressed as

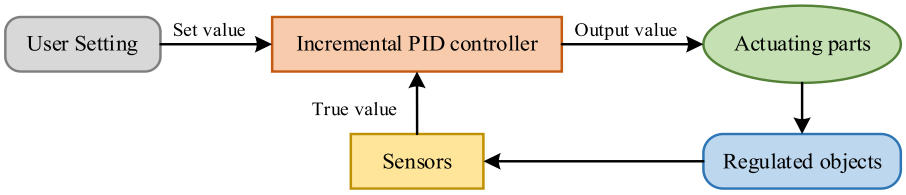


Fig. 2. Regulation process of incremental PID control.

$$\mathbf{x}_{ij} = (\mathbf{u}_j - \mathbf{l}_j)\cdot r_1 + \mathbf{l}_j, i = 1, 2, \cdots, n; j = 1, 2, \cdots, d \tag{4}$$

where $\mathbf{x}_{ij}$ represents the j th dimension of the i th individual; $\mathbf{u}_j$ and $\mathbf{l}_j$ are the upper and lower bounds of the j th variable (dimension), respectively; r_1 is a random number from 0 to 1.

Assuming that the objective function is F. The objective function value $\mathbf{f}_i$ for the i th individual can be calculated from Eq. (5).

$$\mathbf{f}_i = F(\mathbf{x}_i) \tag{5}$$

4.2. Incremental PID control

4.2.1. Calculating system deviations

For the minimization problem, the best individual $\mathbf{x}^*(t)$ at iteration number t is the individual corresponding to the population history minimum. The deviation $\mathbf{e}_k(t)$ of the population for a number of iterations t is

$$\mathbf{e}_k(t) = \mathbf{x}^*(t-1) - \mathbf{x}(t-1) \tag{6}$$

To facilitate the calculation and iterative update, the deviation of the population of the previous iteration is denoted by $\mathbf{e}_{k-1}(t)$ separately when the number of iterations is t, and the deviation of the population of the previous two iterations is denoted by $\mathbf{e}_{k-2}(t)$ separately. Then when $t = 1$, it is worthwhile to make $\mathbf{e}_{k-2}(t) = \mathbf{e}_{k-1}(t) = \mathbf{e}_k(t)$. When $t > 1$, $\mathbf{e}_{k-2}(t) = \mathbf{e}_{k-1}(t-1)$. In the PID algorithm, when $t > 1$, $\mathbf{e}_{k-1}$ should be equal to the deviation between the user-set value and the true value in the system at the previous moment. In PSA, the user-set value is the best value, but the best value after each iteration will not always remain the same; therefore, $\mathbf{e}_{k-1}(t)$ cannot simply be equal to $\mathbf{e}_k(t-1)$ in the proposed PSA. As shown in Fig. 3, $\mathbf{e}_{k-1}(t)$ should equal $x^*(t) - x(t-1)$, but this also means storing $x(t-1)$. In order to reduce the space complexity of the algorithm as much as possible, it is better to make $\mathbf{e}_k(t-1) + x^*(t) - x^*(t-1)$ instead of $x^*(t) - x(t-1)$ as the final $\mathbf{e}_{k-1}(t)$. So, $\mathbf{e}_{k-1}(t)$ can be expressed as

$$\mathbf{e}_{k-1}(t) = \mathbf{e}_k(t-1) + x^*(t) - x^*(t-1) \tag{7}$$

4.2.2. PID regulation

In real-world problems, the Proportion, Integral and Differential factors are adjusted according to different situations and problems. Then the output value $\Delta\mathbf{u}(t)$ of the PID regulation when the number of iterations is t is

$$\Delta\mathbf{u}(t) = K_p\cdot\mathbf{r}_2\cdot[\mathbf{e}_k(t) - \mathbf{e}_{k-1}(t)] + K_i\cdot\mathbf{r}_3\cdot\mathbf{e}_k(t) + K_d\cdot\mathbf{r}_4\cdot[\mathbf{e}_k(t) - 2\mathbf{e}_{k-1}(t) + \mathbf{e}_{k-2}(t)] \tag{8}$$

where $\mathbf{r}_2$, $\mathbf{r}_3$ and $\mathbf{r}_4$ are vectors of random numbers from 0 to 1 in n rows and 1 column; K_p, K_i, and K_d are the adjustment coefficients for the Proportion, Integral and Differential, respectively, which are set to 1, 0.5 and 1.2, respectively, in this paper.

In a conventional PID algorithm, the output regulation value of 0 means that the real value has reached the user's set value at some point. However, as time increases, the true value will soon not equal the set value if the controlled object is not adjusted. Hence, to prevent this phenomenon, a constant is usually added to the original regulation value. It is not good for PSA that all solutions are extremely close to the best solution, especially in the early iterations, and this phenomenon may lead the algorithm to fall into a local optimum. Therefore, on the original $\Delta\mathbf{u}(t)$, this paper also adds a conditioning factor called zero output to prevent the algorithm from falling into local optimum. The zero output is defined in Eq. (9).

$$\mathbf{o}(t) = (\cos(1 - t/T) + \lambda\mathbf{r}_5\cdot\mathbf{L})\cdot\mathbf{e}_k(t) \tag{9}$$

where $\mathbf{r}_5$ are vectors of random numbers from 0 to 1 in n rows and d columns; λ is an adjustment coefficient that is calculated in Eq. (10).

$$\lambda = [\ln(T - t + 2)/\ln(T)]^2 \tag{10}$$

The variation of λ with t is shown in Fig. 4. It can be seen that λ slowly decreases as t increases, which helps the algorithm to fully explore. At the later stage, λ decreases rapidly, which helps the algorithm turn from exploration to exploitation.

$\mathbf{L}$ in the Eq. (9) is a Lévy flight function, which is defined in Eq. (11).

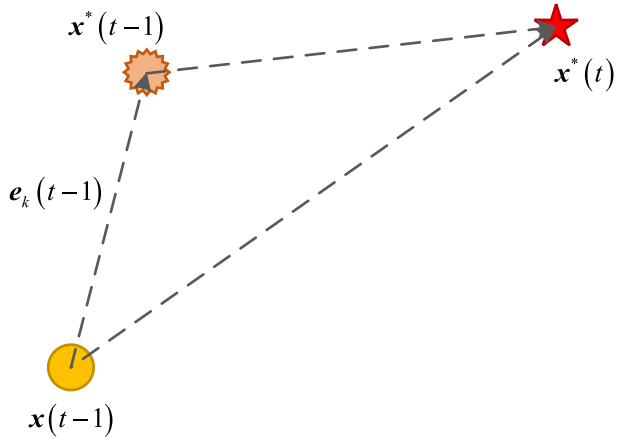


Fig. 3. Update of $\mathbf{e}_{k-1}(t)$.

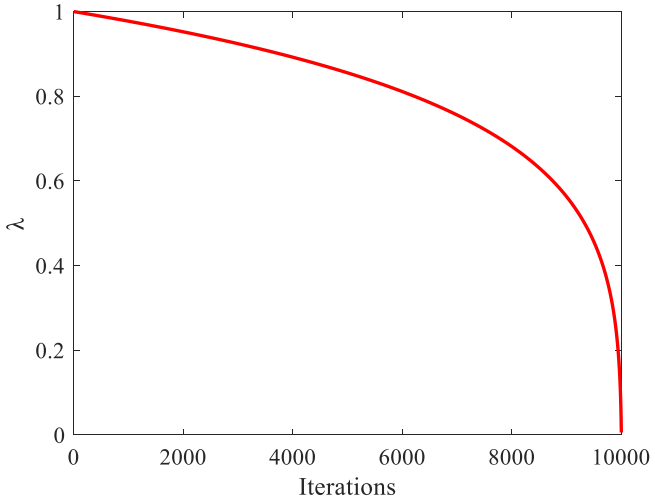


Fig. 4. The variation of λ with t (take $T = 10000$ for example).

$$\mathbf{L}=\frac{\mathbf{u}\sigma}{|\mathbf{v}|^{1/\beta}};\sigma=\left[\frac{\Gamma(1+\beta)\times\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\times\beta\times2^{(\beta-1)/2}}\right]^{1/\beta} \tag{11}$$

where $\mathbf{u}$ and $\mathbf{v}$ are matrices of n rows and d columns of random numbers that obey the standard normal distribution, respectively; β is a factor that is set to 1.5.

The updates of all individuals are related to $\Delta\mathbf{u}(t)$ and $\mathbf{o}(t)$. The population renewal formula is defined as

$$\mathbf{x}(t+1)=\mathbf{x}(t)+\eta\cdot\Delta\mathbf{u}(t)+(1-\eta)\cdot\mathbf{o}(t) \tag{12}$$

where η is a matrix of n rows and 1 column, which is expressed as

$$\eta=\mathbf{r}_6\cos(t/T) \tag{13}$$

where $\mathbf{r}_6$ is a matrix of 0 to 1 random numbers in n rows and 1 columns. The pseudo-code and flowchart of PSA are shown in Algorithm 2 and Fig. 5, respectively.

5. Experimental results and discussion

This section shows the optimization results of PSA on the CEC2017 benchmark test functions and analyzes the exploitation capability, exploration capability, capability to avoid local optima, capability to solve for optima and convergence capability of PSA. Later, Wilcoxon rank sum test and Friedman test are used to verify the superiority of PSA in terms of statistical significance.

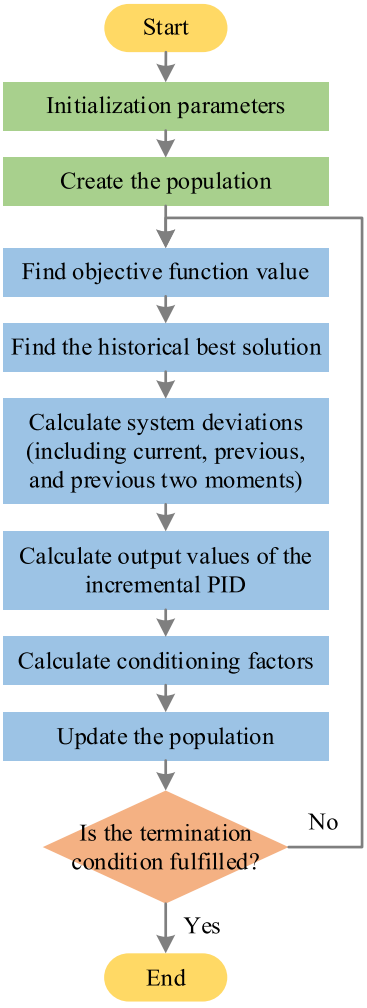


Fig. 5. Flowchart of the PID-based search algorithm.

5.1. Description of benchmark test functions

The CEC2017 benchmark test functions are considered to be a very challenging set of test functions. It contains two unimodal functions (F1 and F3), seven multimodal functions (F4 ~ F10), ten hybrid functions (F11 ~ F20) and ten composition functions (F21 ~ F30). The F2 function is no longer used for algorithm testing in most studies due to its large instability. These functions simulate the complexity of a real search space that contains a large number of local optima with functions of different shapes in different regions. For additional description see literature (Braik et al., 2022).

5.2. Experimental setup

All algorithms were implemented on a computer with 64-bit Windows 10 using MATLAB R2022b. Set the upper and lower bounds of the CEC2017 benchmark test functions to 100 and − 100, respectively, with 50 dimensions. To verify the performance of PSA, FHO (Azizi et al., 2022), GJO (Chopra & Ansari, 2022), AOS (Azizi, 2021), TSA (Kaur et al., 2020), SOA (Dhiman & Kumar, 2019), HHO (Heidari et al., 2019), and SHO (Dhiman & Kumar, 2017) are selected and compared with the proposed PSA. The population size of these algorithms is 50, the maximum number of function evaluations is 500,000 (10000*dimension), and other parameters are set in Table 1.

Algorithm 2 PID-based search algorithm

1. *% Initialization*
2. Initialize the population size n and the maximum number of iterations T
3. Initialize the number of iterations $t=1$
4. *% Create the population*
5. **for** $i=1:n$
6. **for** $j=1:d$
7. Create the j th dimension of the i th individual $\mathbf{x}_{ij}$ using Eq. (4)
8. **end**
9. **end**
10. *% Create the fitness vector*
11. **for** $i=1:n$
12. Calculate the value of the objective function for the i th individual $\mathbf{f}_i$ using Eq. (5)
13. **end**
14. *% Main loop*
15. **while** $t<T$
16. *% Step 1: Guide selection mechanism*
17. Select best individual $\mathbf{x}^*(t-1)$ from the population
18. *% Step 2: Search operations*
19. Calculate $\mathbf{e}_k(t)$ using Eq. (6)
20. **if** $t=1$
21. $\mathbf{e}_{k-2}(t)=\mathbf{e}_{k-1}(t)=\mathbf{e}_k(t)$
22. **else**
23. $\mathbf{e}_{k-2}(t)=\mathbf{e}_{k-1}(t-1)$
24. Calculate $\mathbf{e}_{k-1}(t)$ using Eq. (7)
25. **end**
26. Calculate $\Delta\mathbf{u}(t)$ using Eq. (8)

(*continued on next page*)

Table 1
Parameter setting of the algorithms.

Algorithms	Parameters	Value
PSA	Proportion factor K_p	1
	Integral factor K_i	0.5
	Differential factor K_d	1.2
	The factor of Levy flight β	1.5
FHO	Parameter-less	
GJO	Parameter-less	
AOS	Maximum number of layers around nucleus	10
	Photon Rate	0.1
TSA	Initial speed	1
	Subordinate speed	4
SOA	Control parameter (A)	[2, 0]
	f_c	2
HHO	Parameter-less	
SHO	Control Parameter $\mathbf{h}$	[1, 5]
	Constant $\mathbf{M}$	[0.5, 1]

(*continued*)

Algorithm 2 PID-based search algorithm

27. Calculate $\mathbf{o}(t)$ using Eqs. (9) ~ (11)
28. *% Step 3: Update mechanism*
29. Update the population × using Eq. (12) and (13)
30. *% Next generation until termination criterion*
31. $t = t + 1$
32. **end**

5.3. Exploitation analysis

The unimodal functions have only one extreme value point and are often used to test the exploitation performance of metaheuristic algorithms. The optimization results of the unimodal test functions are given in Table 2. On F1, PSA and AOS rank 1st and 2nd for both mean and variance values, respectively. On F3, PSA ranks 1st in mean value and 5th in variance value. Although the variance value of PSA on F3 is not good, it also illustrates that PSA has the ability to find better solutions (the premise is that PSA ranks 1st in mean value of F3). On balance, PSA has a strong exploitation capability.

5.4. Exploration analysis

In general, metaheuristic algorithms are explored upfront and exploited later. If the algorithm is poorly explored, it is extremely easy to fall into a local optimum in the early stage, thus failing to obtain satisfactory results. The multimodal functions have multiple local optima in the entire search space and are often used to test the exploration capability of the algorithm. The optimization results of the multimodal test functions are given in Table 3. PSA ranks 1st on F4 ~ F8, F10 and 2nd on F9 for the index of the mean value. From the data, it is clear that the variance of PSA is not dominant, but the mean is more prominent. This result may occur because PSA is more likely to jump out of the local optimum compared to the comparison algorithm, which makes the fluctuation of PSA solutions greater compared to other algorithms.

5.5. Analysis of the capability to avoid the local optimum

Hybrid and composition functions are considered to be the most challenging functions and are often used to test the capability of metaheuristic algorithms to avoid local optimums. The optimization results of the hybrid test functions are given in Table 4. In terms of mean value, PSA ranks 1st on F11 ~ F19. In terms of variance value, PSA ranks 1st on F11 ~ F15, F18 and F19. On F20, GJO ranks 1st in mean value and FHO ranks 1st in variance value. The optimization results of the composition functions are given in Table 5. From the data, the mean value of PSA ranks 1st on F21 ~ F23, F25, F27 ~ F30. In terms of variance, PSA ranks 1st on F28 ~ F30, and 2nd on F24, F25 and F27. It can be concluded that PSA is an effective algorithm for solving hybrid and composite functions, verifying the capability of PSA to avoid local optimum.

5.6. Analysis of the capability of solving the optimal value

Metaheuristic algorithms usually only yield higher quality solutions with a good balance of exploration and exploitation. Box plots of the optimization results of each algorithm on some of the CEC2017 benchmark test functions are given in Fig. 6. On F1, F4, F6, F11 ~ F15, F18, F19, F28 ~ F30, the quality of the solutions obtained by PSA is higher than that of the comparison algorithm and exhibits less fluctuations. In particular, on F1, the quality of the solutions obtained by PSA is much higher than that of the comparison algorithm. The capability of PSA to solve the optimal value is verified by statistically obtaining that PSA ranks 1st on the mean value of twenty-five functions and 2nd on the mean value of four functions.

5.7. Convergence analysis

The convergence analysis of metaheuristic algorithms is another avenue to understand the exploration and exploitation mechanisms of the algorithms. Fig. 7 shows the search history, trajectory, average fitness and convergence curves of PSA on some of the CEC2017 benchmark test functions (2D), where population size is 30 and maximum iteration is 500 (number of function evaluations is 15000). In the early iterations, PSA tends to be more exploration-oriented, and in the later iterations, PSA tends to be more exploitation-oriented.

Search history can show the historical location of the population. It can be seen from Fig. 7 that the density is greater at locations close to the global optimum and less at locations far from the global optimum, which indicates that PSA is able to explore regions close to the global optimum. The first individual's trace shows the value of the first variable in each iteration. For the unimodal function F1, the solution fluctuates more in the early iterations and gradually smooths out in the late iterations, and this behavior helps to speed up the convergence rate. Since the multimodal functions have more local extremes, the solution may fluctuate more in the middle of the iteration in order to jump out of the local extremes. This behavior is expressed in F5 and F7. However, on F10, PSA stabilizes quickly after the initial fluctuations, which indicates that PSA can find promising regions more quickly on the multimodal function. As can be seen from F21 and F28, the solution may also fluctuate greatly in early part of the iteration, which is due to the difficulty of solving the composite function and the presence of a large number of local extremes, making the solution fluctuate greatly around the global optimum. Moreover, this is showing that PSA is not prone to fall into local extremes, ensuring its capability to explore.

The average fitness reflects the average level of all individuals in each iteration. The average fitness all show a decreasing trend during the iterations, which proves that PSA improves the accuracy of understanding during the runtime. The convergence curve is the best value of the whole population in each iteration. It can be seen that the curve decreases faster and smooths out at the later stage, verifying the convergence of PSA.

The convergence curves of PSA and its comparison algorithms on partial functions are given in Fig. 8. It can be seen that PSA shows a fast convergence rate. Notably, PSA excels in functions F1, F6, F10, F12, F13, F14, F15, F18, F19, F22 and F30. On F4 and F28, both PSA and AOS achieve better convergence. It can be seen that PSA shows a better convergence speed throughout the iterations of F3 and F14. On the other functions, PSA levels off in the middle and late stages, but the faster convergence speed in the early stages ensures the optimization quality. These indicate that PSA has a strong convergence property.

Table 2
Optimization results of the unimodal and multimodal test functions.

No.	Index	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F1	Ave	**3.7051E + 03**	5.9800E + 10	3.2131E + 10	4.7735E + 06	6.3461E + 10	2.4385E + 10	2.7309E + 08	1.2231E + 11
	Var	**1.1170E + 07**	5.3680E + 20	4.1774E + 19	2.5013E + 12	1.5652E + 20	3.3519E + 19	1.6129E + 08	2.6457E + 19
F3	Ave	**4.9280E + 03**	2.6331E + 05	9.8734E + 04	2.9611E + 04	1.5428E + 05	8.1512E + 04	4.3797E + 08	2.9220E + 08
	Var	2.8499E + 08	4.3043E + 09	2.8336E + 08	**3.8604E + 07**	4.6076E + 08	2.0239E + 08	4.1505E + 15	1.2410E + 18

Bolded values represent the smallest values.

Table 3
Optimization results for multimodal test functions.

No.	Index	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F4	Ave	**4.9248E + 02**	3.3591E + 03	4.0484E + 03	6.0543E + 02	1.3443E + 04	1.9833E + 03	8.1170E + 02	4.4835E + 04
	Var	**2.4644E + 03**	5.6209E + 06	1.5982E + 06	3.2979E + 03	2.9172E + 07	4.7370E + 05	7.7685E + 03	2.1747E + 07
F5	Ave	**7.6598E + 02**	9.7549E + 02	8.7197E + 02	8.2717E + 02	1.0131E + 03	8.5242E + 02	9.1926E + 02	1.2737E + 03
	Var	3.7395E + 03	1.5957E + 03	2.5891E + 03	1.2542E + 03	1.4062E + 03	2.1537E + 03	1.6348E + 03	**5.8095E + 02**
F6	Ave	**6.0087E + 02**	6.4343E + 02	6.4698E + 02	6.6151E + 02	6.8397E + 02	6.5029E + 02	6.7983E + 02	7.2243E + 02
	Var	**8.9999E-01**	3.2198E + 00	6.8267E + 01	3.2686E + 01	4.9278E + 01	3.7005E + 01	1.6659E + 01	5.6923E + 01
F7	Ave	**1.1410E + 03**	1.8290E + 03	1.3536E + 03	1.6136E + 03	1.8576E + 03	1.4494E + 03	1.8516E + 03	2.1473E + 03
	Var	5.3518E + 03	8.7984E + 03	5.6794E + 03	1.0516E + 04	4.0149E + 03	9.1702E + 03	4.1004E + 03	**1.0662E + 03**
F8	Ave	**1.0709E + 03**	1.2513E + 03	1.1753E + 03	1.1467E + 03	1.3276E + 03	1.1590E + 03	1.2189E + 03	1.6138E + 03
	Var	3.4090E + 03	8.7651E + 02	2.6081E + 03	1.9665E + 03	1.5728E + 03	2.2581E + 03	1.1714E + 03	**7.6569E + 02**
F9	Ave	1.0360E + 04	**8.5075E + 03**	1.5363E + 04	1.3342E + 04	2.9099E + 04	1.4031E + 04	2.2327E + 04	5.5411E + 04
	Var	8.3075E + 06	**1.4960E + 06**	2.2786E + 07	2.6161E + 06	7.5212E + 06	9.0259E + 06	9.0179E + 06	4.6690E + 07
F10	Ave	**6.8323E + 03**	1.4658E + 04	9.8321E + 03	8.5315E + 03	1.2248E + 04	9.3479E + 03	1.0149E + 04	1.7087E + 04
	Var	6.5647E + 05	5.2741E + 05	3.1835E + 06	1.0441E + 06	7.4856E + 05	1.5936E + 06	1.7654E + 06	**4.4498E + 05**

Bolded values represent the smallest values.

Table 4
Optimization results for hybrid test functions.

No.	Index	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F11	Ave	**1.3798E + 03**	6.0020E + 03	7.3113E + 03	1.5049E + 03	1.7753E + 04	4.1174E + 03	1.8152E + 03	9.8369E + 04
	Var	**7.6284E + 03**	5.9807E + 05	1.0282E + 07	1.0720E + 04	2.8932E + 07	2.0036E + 06	2.6639E + 04	2.5221E + 10
F12	Ave	**6.1736E + 05**	4.8650E + 09	6.9896E + 09	1.1771E + 08	3.2495E + 10	3.2868E + 09	3.1807E + 08	1.0225E + 11
	Var	**1.8137E + 11**	4.4534E + 17	1.8040E + 19	2.6200E + 15	1.4150E + 20	2.9908E + 18	3.8905E + 16	2.2895E + 20
F13	Ave	**7.8052E + 03**	1.2518E + 09	1.2251E + 09	3.1147E + 05	1.4362E + 10	5.8656E + 08	9.9918E + 06	6.1797E + 10
	Var	**4.4404E + 07**	4.8608E + 16	4.3149E + 18	3.7403E + 10	8.3774E + 19	9.7098E + 17	4.3728E + 13	3.0524E + 20
F14	Ave	**3.5709E + 04**	5.5484E + 06	1.4930E + 06	2.5331E + 05	1.4814E + 07	3.4490E + 05	1.9661E + 06	1.9304E + 08
	Var	**9.5924E + 08**	1.1023E + 14	7.1342E + 12	2.2693E + 10	2.7433E + 14	1.1505E + 11	2.7371E + 12	1.1569E + 16
F15	Ave	**9.5857E + 03**	3.7087E + 08	3.1789E + 08	7.4475E + 04	2.3708E + 09	2.9006E + 07	1.1830E + 06	1.3977E + 10
	Var	**4.4095E + 07**	2.8673E + 17	3.2967E + 17	3.0812E + 09	4.8881E + 18	1.3934E + 15	5.7266E + 11	1.3655E + 19
F16	Ave	**3.3200E + 03**	5.1892E + 03	3.5667E + 03	4.0674E + 03	5.0458E + 03	3.6338E + 03	4.9596E + 03	1.2973E + 04
	Var	2.7420E + 05	2.6787E + 05	1.6896E + 05	2.5440E + 05	7.9606E + 05	**1.4860E + 05**	3.7564E + 05	4.7044E + 06
F17	Ave	**3.1994E + 03**	3.8519E + 03	3.2483E + 03	3.4361E + 03	4.6358E + 03	3.3344E + 03	3.8945E + 03	3.0648E + 04
	Var	1.0818E + 05	**2.2885E + 04**	7.3345E + 04	1.0954E + 05	1.9396E + 06	9.0052E + 04	1.5712E + 05	2.3319E + 08
F18	Ave	**1.4904E + 05**	1.6169E + 07	8.8890E + 06	1.5444E + 06	5.7646E + 07	3.3578E + 06	5.6459E + 06	7.1215E + 08
	Var	**4.6268E + 09**	2.1129E + 14	1.2395E + 14	5.9130E + 11	3.1990E + 15	3.8744E + 12	1.7381E + 13	1.2453E + 17
F19	Ave	**1.8073E + 04**	1.3259E + 08	1.1847E + 08	2.2295E + 06	1.2995E + 09	5.0939E + 06	2.2833E + 06	7.9398E + 09
	Var	**1.1772E + 08**	1.2709E + 15	5.3803E + 16	1.4661E + 12	1.2137E + 18	3.3786E + 14	6.3471E + 12	4.0422E + 18
F20	Ave	3.1829E + 03	3.4508E + 03	**3.1455E + 03**	3.2064E + 03	3.6061E + 03	3.2689E + 03	3.5559E + 03	4.9955E + 03
	Var	6.1579E + 04	**5.1427E + 04**	1.1579E + 05	6.9334E + 04	8.5663E + 04	9.8898E + 04	1.1402E + 05	7.3375E + 04

Bolded values represent the smallest values.

Table 5
Optimization results for composition test functions.

No.	Index	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F21	Ave	**2.5769E + 03**	2.7305E + 03	2.6570E + 03	2.7071E + 03	2.9371E + 03	2.6241E + 03	2.9287E + 03	3.4631E + 03
	Var	4.4000E + 03	**1.7939E + 02**	2.6155E + 03	3.2898E + 03	3.8988E + 03	1.2790E + 03	1.0718E + 04	1.1554E + 04
F22	Ave	**8.8878E + 03**	1.4442E + 04	1.2018E + 04	1.0105E + 04	1.4611E + 04	1.0858E + 04	1.2581E + 04	1.8965E + 04
	Var	1.5904E + 06	1.9799E + 07	2.7290E + 06	1.1891E + 06	8.6891E + 05	1.6622E + 06	1.5153E + 06	**4.4378E + 05**
F23	Ave	**3.0605E + 03**	3.2588E + 03	3.1937E + 03	3.6115E + 03	3.8674E + 03	3.0720E + 03	3.9463E + 03	4.8023E + 03
	Var	7.3548E + 03	**3.5768E + 03**	7.2968E + 03	2.6475E + 04	2.8175E + 04	3.9205E + 03	4.7956E + 04	7.4185E + 04
F24	Ave	3.2261E + 03	3.8519E + 03	3.4038E + 03	3.8514E + 03	4.1015E + 03	**3.1559E + 03**	4.2134E + 03	5.2708E + 03
	Var	4.0135E + 03	5.5713E + 04	8.2067E + 03	4.5495E + 04	6.9616E + 04	**2.1502E + 03**	4.1750E + 04	1.9640E + 05
F25	Ave	**3.0539E + 03**	8.7545E + 03	5.3008E + 03	3.0965E + 03	8.9433E + 03	4.4730E + 03	3.2566E + 03	1.7195E + 04
	Var	2.0281E + 03	5.3702E + 06	4.8371E + 05	**5.2307E + 02**	3.9760E + 06	1.6322E + 05	2.8166E + 03	6.9700E + 05
F26	Ave	7.3008E + 03	9.4763E + 03	8.8228E + 03	1.1325E + 04	1.4544E + 04	**7.2492E + 03**	1.0464E + 04	1.8696E + 04
	Var	2.2658E + 06	1.4693E + 06	7.7438E + 05	2.6198E + 06	1.2864E + 06	**2.1474E + 05**	5.5426E + 06	2.9147E + 05
F27	Ave	**3.5313E + 03**	4.4743E + 03	3.9522E + 03	4.2771E + 03	5.0973E + 03	3.5852E + 03	4.6055E + 03	7.7278E + 03
	Var	9.7802E + 03	7.7674E + 04	2.7212E + 04	1.5164E + 05	3.1096E + 05	**7.3580E + 03**	3.6042E + 05	1.0104E + 06
F28	Ave	**3.2952E + 03**	4.7463E + 03	5.5854E + 03	3.3570E + 03	8.4183E + 03	8.2940E + 03	3.6071E + 03	1.5779E + 04
	Var	**3.1560E + 02**	9.8311E + 05	4.0340E + 05	1.0677E + 03	1.4358E + 06	1.3793E + 06	6.3333E + 03	1.8692E + 06
F29	Ave	**4.3412E + 03**	7.4050E + 03	5.4718E + 03	6.6982E + 03	1.1044E + 04	5.4980E + 03	7.0756E + 03	7.2235E + 05
	Var	**7.5636E + 04**	5.4407E + 06	5.8302E + 05	5.9282E + 05	3.3001E + 07	3.5878E + 05	1.7306E + 06	6.9454E + 11
F30	Ave	**9.8030E + 05**	4.2644E + 08	2.7818E + 08	6.7956E + 07	2.0039E + 09	2.0253E + 08	1.0009E + 08	1.3762E + 10
	Var	**4.2743E + 10**	7.0399E + 15	1.8833E + 17	2.4850E + 14	3.3389E + 18	1.2309E + 17	1.6868E + 15	1.7881E + 19

Bolded values represent the smallest values.

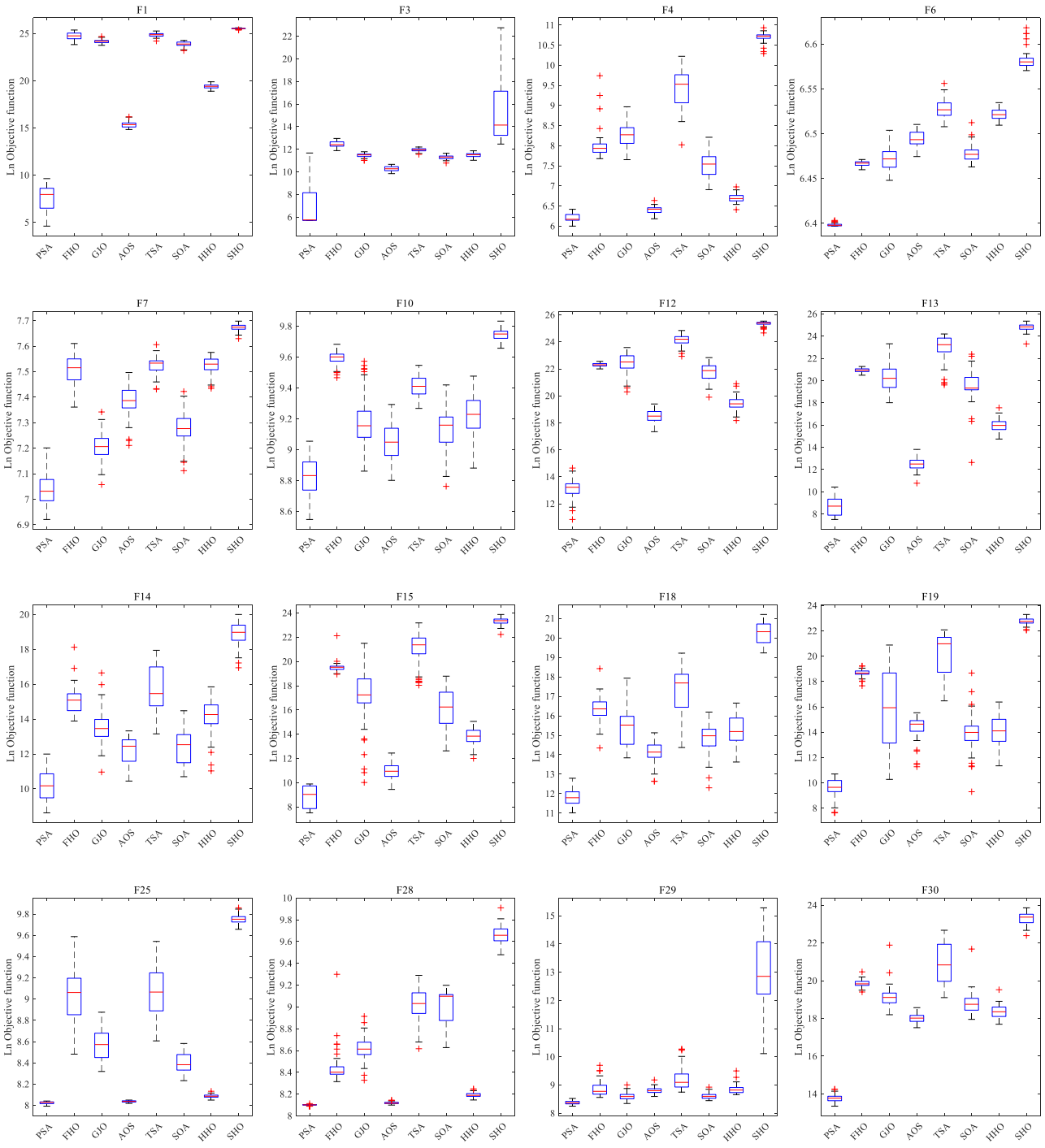


Fig. 6. Optimization results of some CEC2017 benchmark functions.

5.8. Statistical tests

It is well known that the optimization results of metaheuristic algorithms are highly stochastic, and only performing a comparison of two algorithms on a certain function does not fully describe the algorithm performance. Therefore, the Kruskal-Wallis test, Holm test, and Friedman test are used to verify the superiority of the proposed algorithm. The mean of the ranks obtained by Kruskal-Wallis test are given in Table 6. PSA exhibits the smallest mean of the ranks on all functions except on F9, F20, F24, and F26. Table 7 displays the p-values obtained by Holm test. From the data, most p-values are less than 0.05, verifying the excellent performance of PSA. The ranking results of each algorithm tested by Friedman test are shown in Table 8. The ranking results show that PSA is ranked 1st, AOS is ranked 2nd, SOA is ranked 3rd, GJO is ranked 4th, HHO is ranked 5th, FHO is ranked 6th, TSA is ranked 7th and SHO is ranked 8th.

5.9. Computational cost and complexity analysis

Computational complexity is an important metric to measure the merit of an algorithm. The computational cost analysis considers four specific calculations as defined by CEC2017. In these functions, T_0 denotes the time for 1,000,000 calculations of a particular mathematical process (as shown in Eq. (14)) when $x = 0.55$.

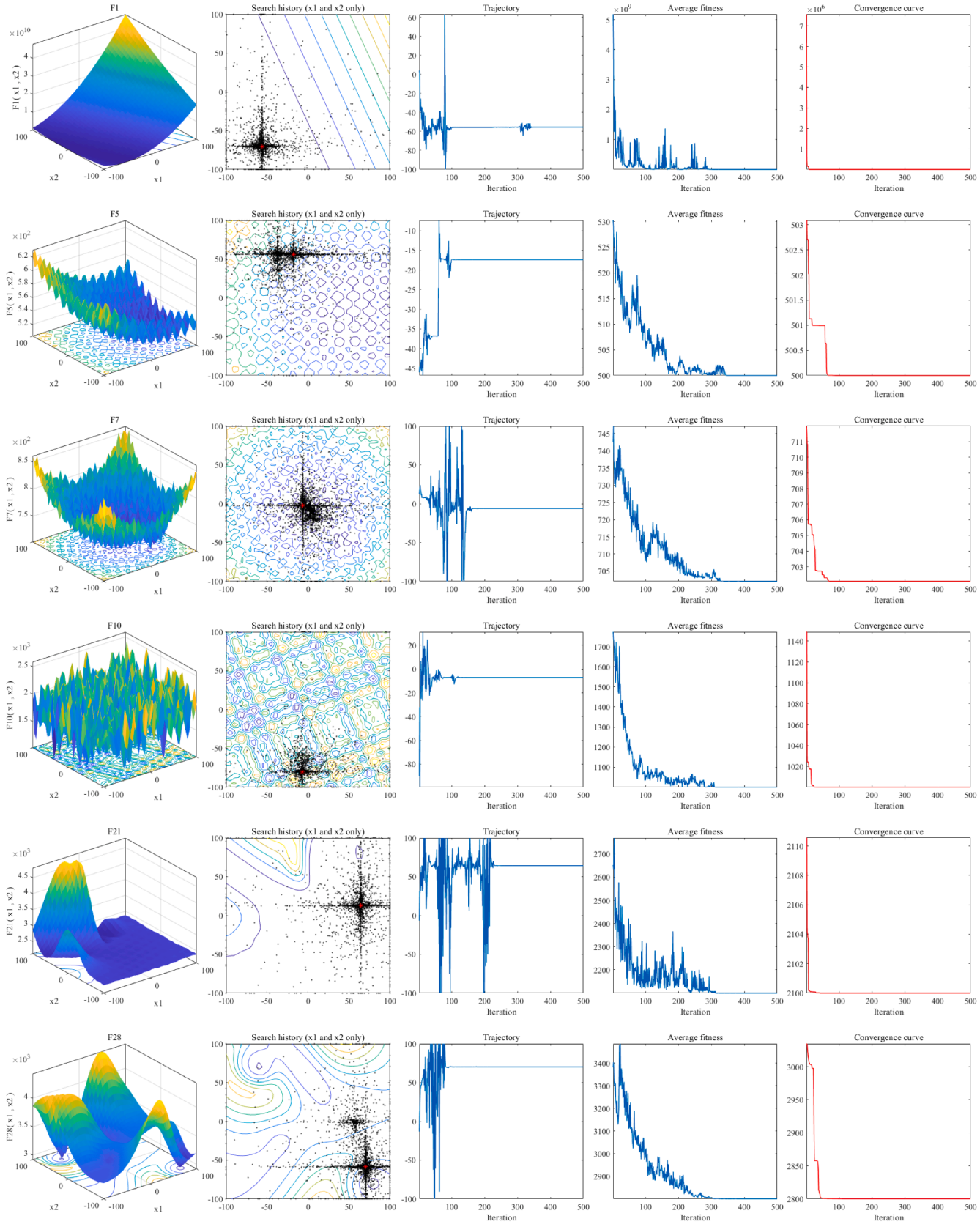


Fig. 7. Search history, trajectory, average fitness, and convergence curve of PSA on some CEC2017 benchmark test functions (2D).

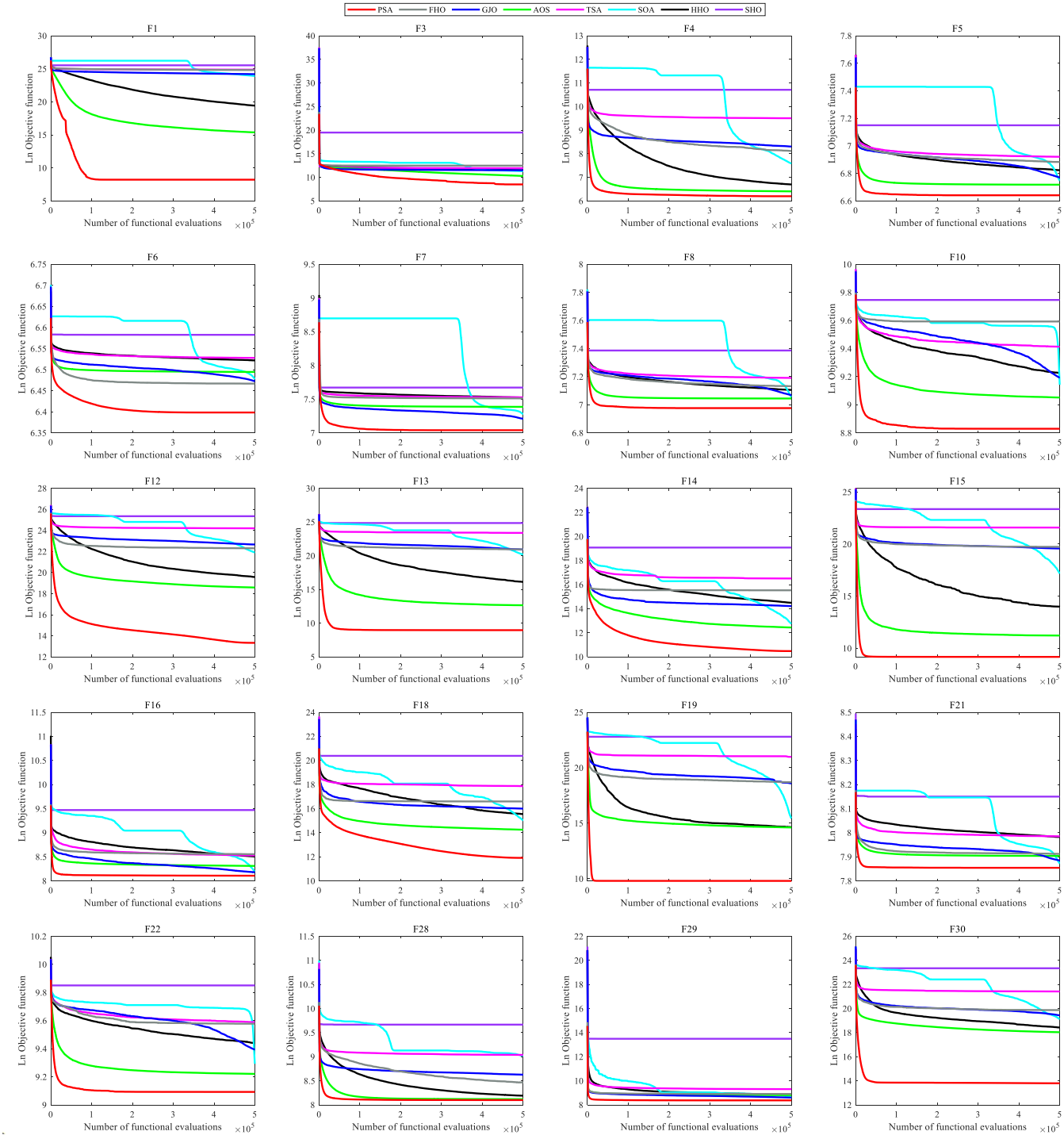


Fig. 8. Convergence curves of PSA and its comparative algorithms on partial functions.

$$x = x + x; x = \frac{x}{2}; x = x \times x; x = \sqrt{x}; x = \ln x; x = e^x; x = \frac{x}{x+2} \tag{14}$$

T_1 indicates the time to compute the Function 18 (F18) 200,000 times; T_2 is the time consumed by the considered algorithm at 200,000 function evaluations for Function 18 (F18); $\overline{T_2}$ is the mean value of the time required to calculate T_2 five times. Table 9 displays the computational cost of AOS and its competitors, which illustrates the superiority of the proposed algorithm.

For PSA, the time complexity depends mainly on the population size (n), the maximum number of iterations (T) and the number of variables to be solved (d). The time complexity of the initialization phase is O(nd). The iterative update phase takes O(ndT) to compute $\mathbf{e}_k(t)$, $\mathbf{e}_{k-1}(t)$ and $\mathbf{e}_{k-2}(t)$ and O(ndT) to update $\boldsymbol{x}$. Thus, the time complexity of PSA is O(ndT). The space complexity of PSA depends mainly on the population size (n) and the number of variables to be solved (d). It is not difficult to conclude that the space complexity of PSA is O(nd).

5.10. Comparison of PSA and strong algorithms

The description and setup of the experimental environment and benchmark functions are the same as in Section 5.1 and Section 5.2. The winners of the IEEE CEC2017 competition (including LSHADE (Tanabe

Table 6
Kruskal-Wallis test results including mean of the ranks.

No.	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F1	**25.5**	288.72	222.82	75.5	303.68	186.8	125.5	375.48
F3	**29.36**	325.34	189.48	74.42	274.32	145.28	191.8	374
F4	**28.94**	231.14	262.74	72.88	322.78	185.3	124.72	375.5
F5	**46.42**	275.34	154.12	97.5	310.4	130.62	214.1	375.5
F6	**25.5**	98.08	131.54	217.42	309.2	155.12	291.64	375.5
F7	**27.16**	264.24	85.88	172.46	279.3	121.3	278.16	375.5
F8	**45.46**	260.44	146.66	111.88	319.76	128.68	215.62	375.5
F9	89.26	**43.3**	178.52	157.9	322.66	166.36	270.5	375.5
F10	**34.78**	323.94	164.02	108.24	263.76	149.12	184.9	375.24
F11	**33.72**	239.3	251.06	69.98	322.02	189.62	122.8	375.5
F12	**25.5**	233.26	249.06	78.62	324.86	194.52	122.74	375.44
F13	**25.5**	262.54	227.6	75.84	317.36	193.34	126.82	375
F14	**29.08**	275.16	188.26	105.4	295.98	116.22	219.12	374.78
F15	**26.1**	276.24	217.32	77.04	308.06	187.36	136.76	375.12
F16	**68.82**	278.82	99.12	163.72	255.28	104.2	258.54	375.5
F17	**91.54**	251.7	97.56	145.68	275.6	122.82	243.64	375.46
F18	**25.7**	267.1	198.7	97.76	297.02	155.06	187.16	375.5
F19	**26.52**	272.22	191.82	156.78	308.1	133.06	140.02	375.48
F20	122.42	209.7	**115.82**	133.44	251.72	156.94	238.46	375.5
F21	**51.6**	201.98	119.92	171.04	304.46	83.9	295.64	375.46
F22	**52.14**	268.82	183.56	99.88	275.1	136.34	212.8	375.36
F23	**55.34**	161.4	126.32	236.66	288.46	59.9	300.76	375.16
F24	68.2	216.26	125.3	217.4	269.42	**35.22**	297.62	374.58
F25	**34.68**	295.46	221.56	66.36	300.9	184.08	125.46	375.5
F26	75.14	178.14	145.2	244.7	322.18	**53.8**	209.34	375.5
F27	**43.24**	240.72	141.7	199.76	301.3	59.56	243.2	374.52
F28	**26.48**	183.76	222.08	74.52	298.4	297.76	125.5	375.5
F29	**29.14**	219.16	114.1	216.54	300.86	115.88	232.9	375.42
F30	**25.5**	281.26	211.38	88.72	311.26	179.24	131.22	375.42

Bolded values represent the smallest values.

Table 7
The p-values of Holm test.

No.	PSA vs. FHO	PSA vs. GJO	PSA vs. AOS	PSA vs. TSA	PSA vs. SOA	PSA vs. HHO	PSA vs. SHO
F1	2.7315E-33	2.3079E-57	1.3550E-38	3.6914E-58	6.6041E-51	3.7465E-51	1.9271E-122
F3	3.8668E-47	2.5847E-48	5.3552E-16	3.7371E-61	1.2793E-43	1.8440E-48	6.6647E-02
F4	1.6973E-13	3.7437E-36	8.9765E-18	6.7111E-31	1.1310E-27	3.3886E-40	8.3305E-84
F5	7.5201E-37	2.1864E-15	1.9042E-08	2.2709E-43	3.0627E-12	1.0699E-26	3.4011E-75
F6	4.1559E-117	1.0560E-61	8.3448E-88	1.3461E-92	8.7772E-77	1.3942E-112	1.2489E-105
F7	2.1349E-63	9.5890E-26	1.6075E-46	1.9002E-73	5.2896E-33	6.4130E-73	1.8358E-95
F8	1.8006E-35	1.3536E-15	7.2873E-11	2.3032E-45	6.5268E-13	4.9075E-28	1.1449E-78
F9	6.2800E-05	6.9304E-09	5.8743E-09	3.0604E-55	1.1417E-08	6.2112E-37	2.2333E-65
F10	2.9613E-72	1.9816E-18	6.1698E-15	4.7474E-54	1.2018E-20	2.9148E-27	6.0912E-85
F11	1.8265E-64	3.2859E-23	2.9057E-09	6.4510E-39	2.1307E-24	2.6878E-30	3.7660E-05
F12	8.3206E-73	3.5921E-20	2.0258E-29	3.4906E-35	5.8564E-24	1.2609E-19	1.0530E-69
F13	1.1859E-62	6.5781E-05	5.1540E-19	5.1677E-19	5.6898E-05	4.1520E-18	2.4652E-44
F14	3.4062E-04	2.0504E-04	1.1816E-16	8.1437E-09	4.9106E-09	7.4394E-13	2.1128E-22
F15	3.8332E-06	1.6727E-04	9.1349E-13	1.9516E-11	3.1235E-07	9.9133E-19	7.9421E-47
F16	9.4958E-33	1.0186E-02	8.8681E-11	1.6289E-20	9.3749E-04	6.8909E-26	6.1409E-52
F17	1.6152E-22	4.1894E-01	5.2332E-04	2.0180E-10	3.4495E-02	1.1925E-15	1.9283E-22
F18	6.9960E-12	2.4276E-07	1.3497E-22	1.3083E-10	6.3358E-20	3.5887E-15	1.1769E-25
F19	3.4279E-46	4.8330E-04	7.1235E-23	4.7286E-13	5.3717E-02	6.5109E-09	1.8822E-48
F20	1.6813E-07	5.3173E-01	6.4814E-01	6.8095E-12	1.3234E-01	8.7296E-09	4.5780E-57
F21	3.4573E-29	9.7901E-10	9.9507E-18	1.7031E-48	2.4404E-05	9.2308E-37	3.0598E-71
F22	2.2330E-13	4.7330E-18	1.2887E-06	1.7117E-45	9.6944E-12	9.0940E-27	1.5709E-71
F23	6.5384E-24	7.4059E-12	2.3312E-38	1.5777E-51	4.4754E-01	1.1474E-46	1.5255E-65
F24	4.9906E-33	1.3711E-19	3.8035E-36	5.5206E-41	7.4607E-09	1.9102E-54	4.8808E-54
F25	1.0224E-31	5.8118E-41	3.8648E-08	7.4171E-38	7.5217E-44	2.2618E-37	5.1976E-108
F26	3.0898E-12	1.5195E-08	8.6708E-23	1.9946E-47	8.1746E-01	2.4782E-12	7.0534E-72
F27	1.4427E-40	4.6776E-28	2.5854E-23	1.3511E-35	4.4448E-03	5.6980E-22	2.2029E-50
F28	2.1391E-17	4.9513E-45	2.1085E-20	1.7652E-51	2.6286E-51	3.0220E-47	4.0789E-82
F29	5.8502E-15	2.5539E-16	5.0475E-37	7.7203E-13	8.1405E-22	6.7431E-26	2.1924E-08
F30	3.8105E-58	1.7542E-05	3.0802E-51	8.6001E-12	9.8076E-05	4.1611E-31	2.6684E-41

& Fukunaga, 2014) and LSHADE-spacma (Mohamed et al., 2017)) and the strong algorithms on that test set (including FDB-AGDE (Guvenc et al., 2021), FDB-TLABC (Duman et al., 2022), AFDB-SFS (Duman et al., 2023), and FDB-SOS (Kahraman et al., 2020)) are selected for comparison with the proposed PSA. The population size of these algorithms is 50, the maximum number of function evaluations is 500,000 (10000*dimension), and other parameters are set in Table 10 (the parameters of the PSA are the same as in Table 1).

Tables 11, 12, and 13 show the error values obtained by optimizing the CEC2017 benchmark test functions (mean result, the best result, and variance of the results, respectively). On the unimodal functions (F1 and F3), PSA can compete with FDB-TLABC and FDB-SOS. Notably, on F1,

Table 8
Friedman test for ranking results.

No.	PSA	FHO	GJO	AOS	TSA	SOA	HHO	SHO
F1	1	6	5	2	7	4	3	8
F3	1	7	4	2	6	3	5	8
F4	1	5	6	2	7	4	3	8
F5	1	6	4	2	7	3	5	8
F6	1	2	3	5	7	4	6	8
F7	1	5	2	4	7	3	6	8
F8	1	6	4	2	7	3	5	8
F9	2	1	5	3	7	4	6	8
F10	1	7	4	2	6	3	5	8
F11	1	5	6	2	7	4	3	8
F12	1	5	6	2	7	4	3	8
F13	1	6	5	2	7	4	3	8
F14	1	6	4	2	7	3	5	8
F15	1	6	5	2	7	4	3	8
F16	1	7	2	4	6	3	5	8
F17	1	5	2	4	7	3	6	8
F18	1	6	5	2	7	3	4	8
F19	1	6	5	2	7	4	3	8
F20	2	5	1	3	7	4	6	8
F21	1	5	3	4	7	2	6	8
F22	1	6	4	2	7	3	5	8
F23	1	4	3	5	6	2	7	8
F24	2	5	3	4	6	1	7	8
F25	1	6	5	2	7	4	3	8
F26	2	4	3	6	7	1	5	8
F27	1	5	3	4	7	2	6	8
F28	1	4	5	2	7	6	3	8
F29	1	6	2	4	7	3	5	8
F30	1	6	5	2	7	4	3	8
FAR	1.1379	5.2759	3.9310	2.8966	6.8276	3.2759	4.6552	8
Rank	1	6	4	2	7	3	5	8

FAR is Friedman average ranking.

Table 9
Computational cost of the algorithms.

Algorithm	Properties	Results	Algorithm	Properties	Results
PSA	T_0	0.0698 (s)	FHO	T_0	0.0670 (s)
	T_1	1.0123 (s)		T_1	1.0588 (s)
	$\overline{T_2}$	2.2703 (s)		$\overline{T_2}$	4.6495 (s)
	$(\overline{T_2}-T_1)/T_0$	18.0124 (unitless)		$(\overline{T_2}-T_1)/T_0$	53.5762 (unitless)
GJO	T_0	0.0677 (s)	AOS	T_0	0.0991 (s)
	T_1	1.0524 (s)		T_1	1.0635 (s)
	$\overline{T_2}$	3.6362 (s)		$\overline{T_2}$	5.5044 (s)
	$(\overline{T_2}-T_1)/T_0$	38.1898 (unitless)		$(\overline{T_2}-T_1)/T_0$	44.8078 (unitless)
TSA	T_0	0.0707 (s)	SOA	T_0	0.0673 (s)
	T_1	1.0271 (s)		T_1	1.0131 (s)
	$\overline{T_2}$	2.7464 (s)		$\overline{T_2}$	2.6429 (s)
	$(\overline{T_2}-T_1)/T_0$	24.3208 (unitless)		$(\overline{T_2}-T_1)/T_0$	24.2248 (unitless)
HHO	T_0	0.0672 (s)	SHO	T_0	0.0674 (s)
	T_1	1.0041 (s)		T_1	1.0198 (s)
	$\overline{T_2}$	1.9829 (s)		$\overline{T_2}$	3.2229 (s)
	$(\overline{T_2}-T_1)/T_0$	14.5688 (unitless)		$(\overline{T_2}-T_1)/T_0$	32.6691 (unitless)

Table 10
Parameter setting of the strong competitors.

Algorithms	Parameters	Value	Algorithms	Parameters	Value
LSHADE	Pbest rate	0.11	LSHADE-spacma	L rate	0.8
	Arc rate	1.4		Pbest rate	0.11
	Memory size	5		Arc rate	1.4
	Maximum population size	50		Memory size	5
	Minimum population size	4		Maximum population size	50
AFDB-SFS	Walk	1		Minimum population size	4
	Maximum diffusion	0		First calss percentage	0.5
FDB-SOS	Parameter-less		FDB-TLABC	Limit	200
FDB-AGDE	Parameter-less			*CR*	0.5

the mean result, the best result and the variance of PSA are better than those of FDB-SOS, FDB-TLABC and FDB-AGDE and FDB-SOS, respectively. On F3, PSA outperforms LSHADE, LSHADE-spacma, and FDB-TLABC in mean outcome; FDB-TLABC and FDB-SOS in best outcome; and LSHADE and LSHADE-spacma in variance. On the multimodal functions (F4 ∼ F10), PSA has some competition. It is worth noting that PSA outperforms FDB-SOS on F5 in terms of best results; on F6, PSA ranks 2nd in terms of best results; and PSA outperforms FDB-SOS on F10 in terms of the mean, the best results and the variance. On the hybrid functions (F11 to F20), PSA shows similar optimization results to the powerful algorithm on F11 and F13. In particular, PSA is superior to FDB-AGDE in the mean result and variance of F11; FDB-SOS is inferior to PSA in the mean result of F12 to F14; and PSA is superior to FDB-SOS in the mean result of F18 and F19. On the composition functions (F21 ∼

Table 11
Mean results of the CEC2017 benchmark test functions.

No.	PSA	LSHADE	LSHADE-spacma	FDB-AGDE	FDB-TLABC	AFDB-SFS	FDB-SOS
F1	3.6051E + 03	5.9145E-08	5.4854E-14	3.4018E + 03	2.7348E + 03	3.5707E-03	4.0438E + 03
F3	4.6280E + 03	3.5403E + 04	6.6534E + 04	2.7070E-01	6.7948E + 03	1.5784E-11	6.6770E + 02
F4	9.2485E + 01	4.1732E + 01	4.2236E + 01	9.2471E + 01	4.6808E + 01	4.6076E + 01	7.2716E + 01
F5	2.6598E + 02	5.5301E + 01	6.5191E + 01	1.0375E + 02	1.1434E + 02	9.2869E + 01	2.2252E + 02
F6	8.6844E-01	6.3964E-01	1.7386E + 00	3.4451E + 00	2.5890E + 00	1.2361E-05	3.4616E-01
F7	4.4102E + 02	1.4013E + 02	1.3590E + 02	2.1689E + 02	2.0571E + 02	1.3547E + 02	3.6350E + 02
F8	2.7091E + 02	5.3590E + 01	6.6409E + 01	1.0373E + 02	1.0584E + 02	9.2372E + 01	2.2467E + 02
F9	9.4596E + 03	2.7456E + 02	3.4096E + 02	8.3902E + 02	7.0389E + 02	2.9532E + 00	7.5253E + 02
F10	5.8323E + 03	3.6987E + 03	3.4246E + 03	4.7076E + 03	3.8630E + 03	4.0028E + 03	7.0928E + 03
F11	2.7978E + 02	1.9978E + 02	2.2574E + 02	2.9059E + 02	1.3312E + 02	6.3149E + 01	1.9636E + 02
F12	6.1616E + 05	9.9069E + 03	2.4039E + 03	7.0291E + 04	1.6027E + 05	5.1552E + 04	1.1277E + 06
F13	6.5052E + 03	1.1962E + 03	3.0169E + 03	3.7310E + 03	4.2747E + 03	8.9373E + 02	7.5524E + 03
F14	3.4309E + 04	3.1868E + 02	3.4995E + 02	2.3100E + 02	1.9854E + 04	6.1193E + 01	5.3595E + 04
F15	8.0857E + 03	4.3166E + 02	5.3071E + 02	3.0859E + 03	4.7852E + 03	8.7739E + 01	7.7671E + 03
F16	1.7200E + 03	8.2450E + 02	7.6170E + 02	1.0461E + 03	8.6835E + 02	8.7690E + 02	1.1859E + 03
F17	1.4994E + 03	6.9246E + 02	6.0619E + 02	7.1246E + 02	6.7307E + 02	5.7371E + 02	8.8680E + 02
F18	1.4724E + 05	1.7602E + 03	3.0332E + 02	8.0900E + 03	1.7893E + 05	9.0537E + 01	3.3106E + 05
F19	1.6173E + 04	1.5186E + 02	2.6360E + 02	7.6517E + 02	1.4364E + 04	3.1825E + 01	1.6578E + 04
F20	1.1829E + 03	4.8390E + 02	4.2599E + 02	5.8974E + 02	2.7573E + 02	4.2113E + 02	7.0820E + 02
F21	4.7691E + 02	2.5270E + 02	2.6376E + 02	3.0065E + 02	2.8681E + 02	2.8527E + 02	4.0361E + 02
F22	6.6878E + 03	3.7620E + 03	3.2560E + 03	4.5566E + 03	2.9069E + 03	3.9130E + 03	7.0130E + 03
F23	7.6050E + 02	4.9031E + 02	5.1381E + 02	5.6638E + 02	5.2496E + 02	5.2121E + 02	6.4821E + 02
F24	8.2614E + 02	5.5689E + 02	5.8447E + 02	6.5520E + 02	5.9512E + 02	6.0478E + 02	7.5792E + 02
F25	5.5388E + 02	5.2192E + 02	5.3010E + 02	5.4718E + 02	5.5532E + 02	5.3842E + 02	5.4223E + 02
F26	4.7008E + 03	1.8772E + 03	2.0528E + 03	2.8854E + 03	2.9441E + 03	2.2063E + 03	3.2876E + 03
F27	8.3131E + 02	6.4328E + 02	7.2255E + 02	7.5919E + 02	6.5618E + 02	5.6712E + 02	6.7881E + 02
F28	4.9521E + 02	4.8944E + 02	4.8992E + 02	9.9888E + 02	4.9773E + 02	4.8328E + 02	4.9662E + 02
F29	1.4412E + 03	6.3169E + 02	6.8313E + 02	7.1613E + 02	8.2692E + 02	5.3460E + 02	7.1857E + 02
F30	9.7730E + 05	7.3234E + 05	7.7484E + 05	1.2567E + 09	8.1635E + 05	7.8271E + 05	9.2177E + 05

Table 12
The best results of the CEC2017 benchmark test functions.

No.	PSA	LSHADE	LSHADE-spacma	FDB-AGDE	FDB-TLABC	AFDB-SFS	FDB-SOS
F1	3.9947E-01	3.4106E-13	2.8422E-14	1.3594E-01	1.3285E + 00	1.1140E-07	1.5420E-02
F3	7.0285E-03	6.8212E-13	2.8422E-13	1.0624E-03	2.3721E + 03	5.6843E-13	3.6185E + 01
F4	3.2481E + 00	1.3676E-04	1.1369E-13	4.6371E-02	6.8332E-05	6.8635E-08	1.0323E-01
F5	1.5521E + 02	3.4824E + 01	4.0793E + 01	5.6713E + 01	7.1637E + 01	4.1788E + 01	1.6006E + 02
F6	1.2594E-04	1.0883E-02	3.4744E-01	3.1539E-01	3.6304E-01	1.1369E-13	5.1177E-02
F7	3.1331E + 02	9.3510E + 01	1.0308E + 02	1.4071E + 02	1.4971E + 02	1.0654E + 02	2.9611E + 02
F8	1.7149E + 02	3.6814E + 01	4.5771E + 01	6.4672E + 01	6.6662E + 01	6.0692E + 01	1.3627E + 02
F9	2.6158E + 03	3.4897E + 01	5.7970E + 01	1.9429E + 02	1.8442E + 02	0.0000E + 00	1.4178E + 01
F10	4.1552E + 03	2.3720E + 03	2.6757E + 03	2.3952E + 03	2.5180E + 03	2.3077E + 03	5.4296E + 03
F11	1.2373E + 02	7.7905E + 01	1.1080E + 02	3.6773E + 01	7.2987E + 01	3.0844E + 01	5.3219E + 01
F12	4.9726E + 04	2.5823E + 03	1.2431E + 03	1.6663E + 04	2.4490E + 04	8.8155E + 03	8.7295E + 04
F13	5.3058E + 02	1.9020E + 02	1.1033E + 03	1.2899E + 02	1.9466E + 02	3.7191E + 01	2.4760E + 02
F14	4.1649E + 03	2.0336E + 02	1.6129E + 02	4.0333E + 01	4.6294E + 02	2.5937E + 01	1.8184E + 03
F15	3.4341E + 02	1.1152E + 02	2.3299E + 02	9.9363E + 01	1.2484E + 02	4.7093E + 01	1.2869E + 02
F16	4.6771E + 02	3.4937E + 02	2.5957E + 02	3.6581E + 02	3.6564E + 02	3.4839E + 02	3.9361E + 02
F17	8.7773E + 02	3.5034E + 02	2.2322E + 02	1.2096E + 02	2.8236E + 02	2.0913E + 02	3.2959E + 02
F18	5.9328E + 04	9.4215E + 01	9.1579E + 01	5.9274E + 02	4.7013E + 04	4.8021E + 01	3.8943E + 04
F19	1.6999E + 02	6.0779E + 01	1.0354E + 02	2.5311E + 01	8.1071E + 02	1.5784E + 01	1.9762E + 01
F20	5.9239E + 02	1.2485E + 02	8.3773E + 01	8.0615E + 01	6.4550E + 01	6.0660E + 01	7.7457E + 01
F21	3.8698E + 02	2.3434E + 02	2.4823E + 02	2.6148E + 02	2.5196E + 02	2.5026E + 02	3.0350E + 02
F22	1.0000E + 02	1.0000E + 02	1.0000E + 02	1.0000E + 02	1.0000E + 02	1.0000E + 02	1.0000E + 02
F23	6.1640E + 02	4.5718E + 02	4.6964E + 02	5.0984E + 02	4.7396E + 02	4.7416E + 02	4.9937E + 02
F24	6.6221E + 02	5.3341E + 02	5.3776E + 02	5.8983E + 02	5.5125E + 02	5.5739E + 02	6.7213E + 02
F25	4.6023E + 02	4.6060E + 02	4.6059E + 02	4.6018E + 02	4.6029E + 02	4.6101E + 02	4.6020E + 02
F26	3.0000E + 02	1.4904E + 03	3.0000E + 02	2.1819E + 03	2.3841E + 03	1.7437E + 03	3.0000E + 02
F27	6.6663E + 02	5.2942E + 02	5.7563E + 02	5.7531E + 02	5.4883E + 02	5.0888E + 02	5.4543E + 02
F28	4.5885E + 02	4.5335E + 02	4.5335E + 02	4.5860E + 02	4.5337E + 02	4.5362E + 02	4.5885E + 02
F29	9.3043E + 02	3.6796E + 02	3.5959E + 02	4.1113E + 02	4.0335E + 02	3.1791E + 02	3.7192E + 02
F30	6.2610E + 05	5.7941E + 05	5.7943E + 05	5.9026E + 05	5.9059E + 05	5.9375E + 05	6.0688E + 05

F30), most of the optimization results of PSA are under the same order of magnitude as those strong algorithms. It is noteworthy that PSA achieves similar results to the powerful algorithm for the best results of F22; PSA, LSHADE-spacma and FDB-SOS achieve the minimum values for the best results of F26. Furthermore, PSA is better than FDB-SOS on the best result of F24; LSHADE, LSHADE-spacma, FDB-TLABC and AFDB-SFS are worse than PSA on the best result of F25; PSA has a better mean result and variance than FDB-AGDE and FDB-TLABC on F28; and FDB-AGDE is worse than PSA on the mean and variance of F30.

It can be concluded that although PSA as a whole is not yet up to the optimization of these powerful algorithms, it is able to compete with or even outperform these algorithms in certain functions. Therefore, the proposed PSA is competitive.

Table 13
Variance of the optimization results of the CEC2017 benchmark test function.

No.	PSA	LSHADE	LSHADE-spacma	FDB-AGDE	FDB-TLABC	AFDB-SFS	FDB-SOS
F1	1.1170E + 07	2.4095E-14	1.8142E-28	1.5280E + 07	8.4928E + 06	1.1433E-04	3.3226E + 07
F3	2.8499E + 08	6.0128E + 09	8.1588E + 09	4.2218E-01	5.3223E + 06	6.9140E-22	4.5985E + 05
F4	2.4644E + 03	1.9822E + 03	2.2593E + 03	2.5591E + 04	1.7575E + 03	1.5649E + 03	2.4423E + 03
F5	3.7395E + 03	1.3718E + 02	1.1857E + 02	4.8022E + 02	3.1160E + 02	3.8610E + 02	8.2815E + 02
F6	8.9999E-01	3.0102E-01	9.4468E-01	3.3382E + 00	1.9373E + 00	7.3074E-09	2.9193E-02
F7	5.3518E + 03	4.7185E + 02	4.1307E + 02	1.4884E + 03	1.1375E + 03	3.3072E + 02	9.4554E + 02
F8	3.4090E + 03	1.1629E + 02	1.1234E + 02	4.0655E + 02	4.3403E + 02	3.5776E + 02	1.6800E + 03
F9	8.3075E + 06	4.0749E + 04	4.7279E + 04	1.5014E + 05	1.6409E + 05	4.6050E + 01	6.3575E + 05
F10	6.5647E + 05	4.0468E + 05	1.0239E + 05	1.1876E + 06	4.0075E + 05	3.7001E + 05	4.1591E + 05
F11	7.6284E + 03	3.2623E + 03	3.9794E + 03	1.1794E + 06	7.8860E + 02	1.4481E + 02	3.6979E + 03
F12	1.8137E + 11	4.2332E + 07	2.7827E + 05	1.3805E + 09	8.9830E + 09	1.2215E + 09	8.6930E + 11
F13	4.4404E + 07	6.4391E + 05	9.8074E + 05	1.0736E + 07	2.8910E + 07	4.6514E + 06	5.8028E + 07
F14	9.5924E + 08	7.0794E + 03	1.1985E + 04	6.8444E + 04	2.8165E + 08	2.3220E + 02	2.1617E + 09
F15	4.4095E + 07	2.9840E + 04	3.7232E + 04	1.6041E + 07	1.6759E + 07	4.2617E + 02	5.0896E + 07
F16	2.7420E + 05	4.0056E + 04	7.0668E + 04	1.1187E + 05	8.4684E + 04	5.6596E + 04	1.6590E + 05
F17	1.0818E + 05	2.8450E + 04	3.3329E + 04	4.7155E + 04	4.4196E + 04	2.9111E + 04	8.0057E + 04
F18	4.6268E + 09	3.1470E + 06	2.5227E + 04	4.8291E + 07	1.9444E + 10	9.0417E + 02	1.3849E + 11
F19	1.1772E + 08	3.0090E + 03	6.2722E + 03	4.7589E + 06	6.3513E + 07	9.8719E + 01	1.7260E + 08
F20	6.1579E + 04	1.8409E + 04	2.3085E + 04	5.8670E + 04	2.9132E + 04	3.4061E + 04	6.3637E + 04
F21	4.4000E + 03	1.4906E + 02	1.0407E + 02	3.6930E + 02	3.2679E + 02	3.0963E + 02	1.2070E + 03
F22	1.5904E + 06	1.6839E + 06	2.9628E + 06	4.4162E + 06	5.3727E + 06	3.2422E + 06	2.6819E + 06
F23	7.3548E + 03	3.9374E + 02	3.6712E + 02	8.0023E + 02	4.6292E + 02	4.8268E + 02	2.2380E + 03
F24	4.0135E + 03	1.7470E + 02	6.3131E + 02	5.6589E + 03	5.1672E + 02	4.4999E + 02	2.0111E + 03
F25	2.0281E + 03	1.5485E + 03	1.8533E + 03	3.7271E + 03	1.4006E + 03	1.0708E + 03	1.7857E + 03
F26	2.2658E + 06	3.9858E + 04	1.2036E + 05	1.3379E + 05	1.8681E + 05	6.9543E + 04	5.8122E + 05
F27	9.7802E + 03	3.9431E + 03	8.8562E + 03	1.2171E + 05	4.1870E + 03	1.4832E + 03	5.7789E + 03
F28	3.1560E + 02	8.2164E + 02	6.3188E + 02	1.8487E + 06	1.1221E + 03	4.7016E + 02	2.9070E + 02
F29	7.5636E + 04	2.0674E + 04	4.1034E + 04	4.2361E + 04	4.3565E + 04	2.3699E + 04	3.4376E + 04
F30	4.2743E + 10	1.3182E + 10	3.8226E + 10	7.8518E + 19	1.2260E + 10	8.9054E + 09	4.1966E + 10

6. Constrained optimization problems

In this section, the constrained optimization capability of PSA will be verified by six constrained problems, which are three-bar truss design, speed reducer design, tubular column design, car side impact design, pressure vessel design, and function 10 in CEC2006 constrained real-parameter optimization problem (CEC2006 constrained real-parameter optimization-G10). The population size for all algorithms is 50 and the maximum number of function evaluations is 25000. PSA is evaluated by the results of 30 independent runs.

6.1. Three-bar truss design

The objective of the three-bar truss problem is to minimize the volume of a statically loaded three-bar truss constrained by the stresses in each bar. The problem has two decision variables (vertebral canal cross-sectional area A_1, A_2) and three constraints (Ray & Saini, 2001), which are shown schematically in Fig. 9. The mathematical form of the problem is given in Eq. (15).

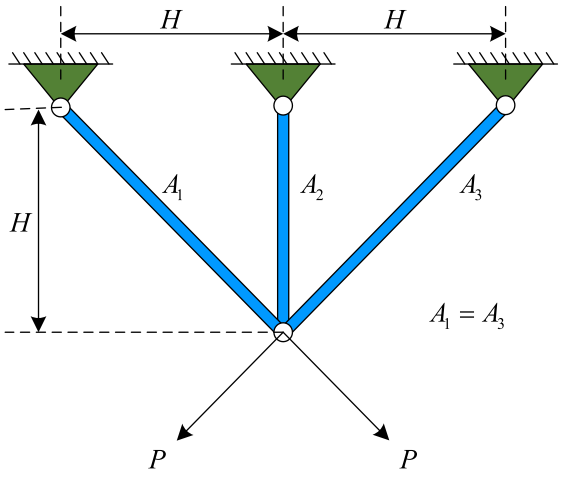


Fig. 9. Three-bar truss design.

Consider:

$$\mathbf{x} = [x_1, x_2] = [A_1, A_2]$$

Minimize:

$$f(\mathbf{x}) = \left(2\sqrt{2}x_1 + x_2\right) \times l$$

Subject to:

$$g_1(\mathbf{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0$$

$$g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0$$

$$g_3(\mathbf{x}) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \le 0 \tag{15}$$

Where:

$l = 100cm, P = 2kN/cm^3, \sigma = 2kN/cm^3$.

Variable range:

$$0 \le x_1, x_2 \le 1$$

The optimization results of each algorithm for the three-bar truss design problem are given in Table 14. From the data, PSA ranks 1st in all indicators. Ranking 2nd in all metrics is the GJO. Collectively, PSA is an effective algorithm to solve the problem.

6.2. Speed reducer design

The design of a speed reducer is a challenging benchmark problem because it contains seven variables as well as eleven constraints (Mezura-Montes & Coello, 2005). These variables include the face width of teeth (b), module of teeth (m), the number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft

Table 14
Optimization results for the three-bar truss design problem.

Algorithms	Mean	Best	Std
PSA	**2.6389588935E + 02**	**2.6389584339E + 02**	**5.6733156698E-05**
FHO	2.6390758451E + 02	2.6389675917E + 02	7.6588765836E-03
GJO	2.6389676063E + 02	2.6389586919E + 02	8.4545208762E-04
AOS	2.6389689189E + 02	2.6389589888E + 02	1.5589851616E-03
TSA	2.6391407414E + 02	2.6389604542E + 02	1.3770191088E-02
SOA	2.6390043003E + 02	2.6389654164E + 02	3.3275097430E-03
HHO	2.6394778118E + 02	2.6389603451E + 02	6.6343836708E-02
SHO	2.6410721058E + 02	2.6390313095E + 02	3.7031332247E-01

Bolded values represent the smallest values.

between bearings (l_2), the diameter of first shafts (d_1) and the diameter of second shafts (d_2). The schematic diagram of the problem is shown in Fig. 10, and its mathematical model is shown in Eq. (16).

Consider:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, p, l_1, l_2, d_1, d_2]$$

Minimize:

$$f(\mathbf{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$$

Subject to:

$$g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0$$

$$g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0$$

$$g_3(\mathbf{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0$$

$$g_4(\mathbf{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0 \tag{16}$$

$$g_5(\mathbf{x}) = \frac{\sqrt{(745x_4/x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \le 0$$

$$g_6(\mathbf{x}) = \frac{\sqrt{(745x_5/x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \le 0$$

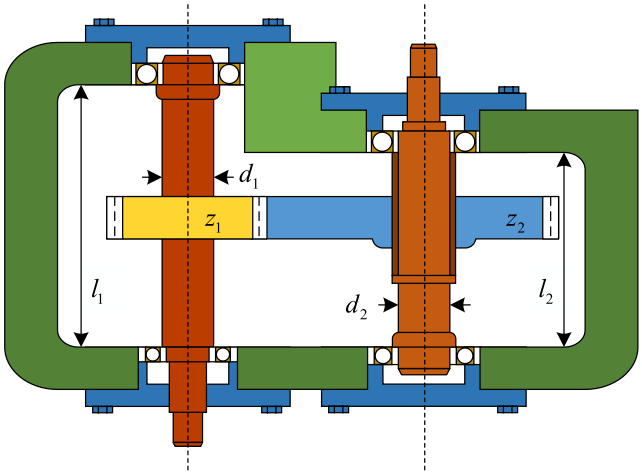


Fig. 10. Speed reducer design.

$$g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \le 0$$

$$g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \le 0$$

$$g_{10}(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(\mathbf{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

Variable range:

$2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $x_3 \in \{17, 18, 19, \ldots, 27, 28\}$, $7.3 \le x_4 \le 8.3$, $7.3 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5 \le x_7 \le 5.5$.

Table 15 shows the optimization results of all algorithms for the speed reducer design problem. From the results, it is clear that PSA ranks first in all indicators. It is worth noting that the GJO ranks 2nd on the index of the mean value and standard deviation. Moreover, the SHO performs the worst on this problem. On balance, PSA has an advantage on this problem.

6.3. Tubular column design

The tubular column problem aims to design a homogeneous column of tubular cross-section to carry compressive loads at minimum cost (Talatahari et al., 2021). The decision variables are the average diameter of the column d and the thickness of the tube t. The tubular column is made of a material with yield stress $\sigma_y = 500kgf/cm^2$ and modulus of elasticity $E = 0.85 \times 10^6 kgf/cm^2$. The problem is shown schematically in Fig. 11 and its mathematical form is given in Eq. (17).

Consider:

$$\mathbf{x} = [x_1, x_2] = [d, t]$$

Minimize:

$$f(\mathbf{x}) = 9.8x_1x_2 + 2x_1$$

Subject to:

$$g_1(\mathbf{x}) = \frac{P}{\pi x_1x_2\sigma_y} - 1 \le 0$$

$$g_2(\mathbf{x}) = \frac{8PL^2}{\pi^3Ex_1x_2(x_1^2 + x_2^2)} - 1 \le 0$$

Table 15
Optimization results for the speed reducer design problem.

Algorithms	Mean	Best	Std
PSA	**2.9944244664E + 03**	**2.9944244658E + 03**	**2.1180473926E-06**
FHO	3.3108831594E + 03	3.1368323268E + 03	6.5095787389E + 01
GJO	3.0066762786E + 03	2.9988119630E + 03	4.3539771595E + 00
AOS	3.0082978098E + 03	2.9983938694E + 03	5.4920338692E + 00
TSA	3.0617992659E + 03	3.0233889618E + 03	2.5684922693E + 01
SOA	3.0196648867E + 03	3.0040626762E + 03	7.8368674136E + 00
HHO	3.0281918272E + 03	2.9973183806E + 03	2.4497472223E + 01
SHO	3.3495307923E + 03	3.1665808844E + 03	1.4609193493E + 02

Bolded values represent the smallest values.

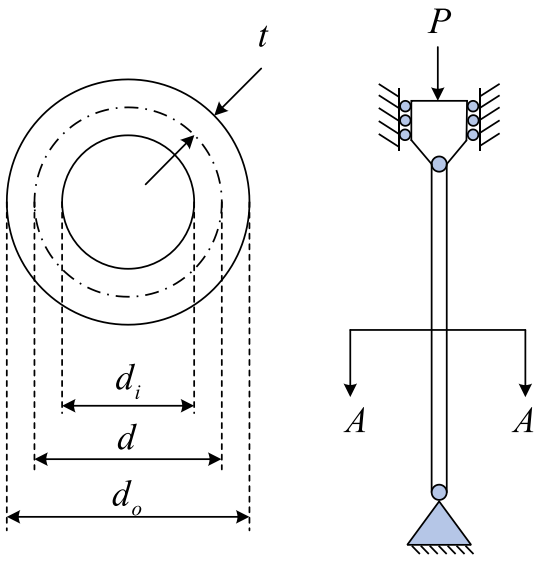


Fig. 11. Tubular column design.

$$g_3(\mathbf{x}) = \frac{2}{x_1} - 1 \le 0$$

$$g_4(\mathbf{x}) = \frac{x_1}{14} - 1 \le 0 \tag{17}$$

$$g_5(\mathbf{x}) = \frac{0.2}{x_2} - 1 \le 0$$

$$g_6(\mathbf{x}) = \frac{x_2}{8} - 1 \le 0$$

Variable range:

$$2 \le x_1 \le 140.2 \le x_2 \le 0.8$$

The optimization results of all algorithms for the tubular column design problem are given in Table 16. As can be seen from the data in the table, PSA is ranked 1st in all indicators. It is worth noting that HHO ranks 2nd in the best value metric; GJO ranks 2nd in both the mean and standard deviation metrics. On balance, PSA has a strong competitive position on the problem.

Table 16
Optimization results for the tubular column design problem.

Algorithms	Mean	Best	Std
PSA	**2.6486361472E + 01**	**2.6486361472E + 01**	**3.1639173292E-15**
FHO	2.6650453689E + 01	2.6504362465E + 01	8.9135431909E-02
GJO	2.6494134678E + 01	2.6488123979E + 01	4.0606510964E-03
AOS	2.6519578600E + 01	2.6487226637E + 01	6.6795264226E-02
TSA	2.6604657947E + 01	2.6519170015E + 01	6.1621525343E-02
SOA	2.6507980100E + 01	2.6491246449E + 01	1.0813564591E-02
HHO	2.6598650713E + 01	2.6486593236E + 01	9.6197420028E-02
SHO	2.8538329483E + 01	2.6717709676E + 01	1.6797639035E + 00

Bolded values represent the smallest values.

6.4. Car side impact design

The mathematical model for this problem is modeled in the literature according to (Gandomi et al., 2011), using the derived response surface method. Eleven decision variables are considered in this study, which are thicknesses of B-pillar inner (x_1), B-pillar reinforcement (x_2), floor side inner (x_3), cross members (x_4), door beam (x_5), door beltline reinforcement (x_6), roof rail (x_7), materials of B-pillar inner (x_8), floor side inner (x_9), barrier height (x_{10}), and hitting position (x_{11}). Fig. 12 shows a model of the car side impact problem. According to the simplified model, the mathematical form of the optimization problem is given in Eq. (18).

Consider:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}]$$

Minimize:

$$f(\mathbf{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7$$

Subject to:

$$g_1(\mathbf{x}) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} - 1 \le 0$$

$$g_2(\mathbf{x}) = 46.36 - 9.9x_2 - 12.9x_1x_2 + 0.1107x_3x_{10} - 32 \le 0$$

$$g_3(\mathbf{x}) = 33.86 + 2.95x_3 + 0.1792x_3 - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 - 32 \le 0$$

$$g_4(\mathbf{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} - 32 \le 0$$

$$g_5(\mathbf{x}) = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} - 0.32 \le 0\ ,$$

$$g_6(\mathbf{x}) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} - 0.02x_2^2 - 0.32 \le 0, \tag{18}$$

$$g_7(\mathbf{x}) = 0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 - 0.32 \le 0$$

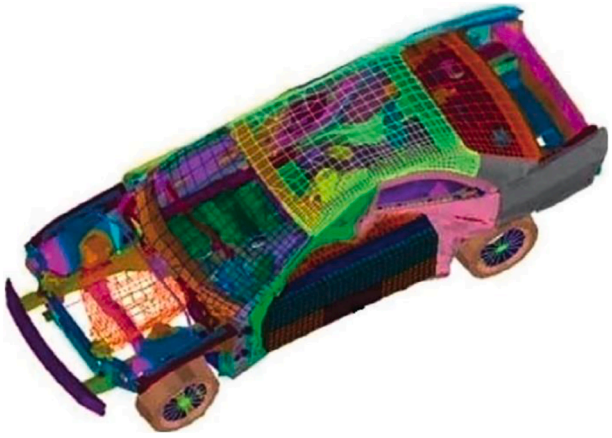


Fig. 12. Car side impact design (Youn & Choi, 2004).

$$g_8(\mathbf{x}) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 - 4 \le 0$$

$$g_9(\mathbf{x}) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \le 0$$

$$g_{10}(\mathbf{x}) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 - 15.7 \le 0$$

Variable range:

$$0.5 \le x_1, x_2, x_3, x_4, x_5, x_6, x_7 \le 1.5 x_8, x_9 \in \{0.192, 0.345\} - 30 \le x_{10}, x_{11} \le 30$$

Table 17 gives the optimization results of all algorithms for the car side impact problem. PSA ranks 1st in all metrics. The GJO, which also achieved excellent performance, ranks second in the mean and best value metrics, but third in the standard deviation. Taken together, PSA is an effective algorithm to solve the problem.

6.5. Pressure vessel design

The design problem of a pressure vessel has four variables and four constraints (Mirjalili et al., 2014). These four variables include head thickness (T_h), shell thickness (T_s), length (U) and inner radius (R), in which T_h and T_s are discrete variables; the values of U and R are continuous. Fig. 13 shows a schematic representation of the problem, whose mathematical model is shown in Eq. (19).

Consider:

$$\mathbf{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, U]$$

Minimize:

$$f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

$$\begin{aligned} g_1(\mathbf{x}) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(\mathbf{x}) &= -x_2 + 0.00954x_3 \le 0 \\ g_3(\mathbf{x}) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(\mathbf{x}) &= x_4 - 240 \le 0 \end{aligned} \tag{19}$$

Variable range:

$x_1, x_2 \in \{z | z = 0.0625i, i = 1, 2, \cdots, 99\}, 10 \le x_3, x_4 \le 200$.

Table 18 shows the optimization results of all algorithms for the pressure vessel design problem. From the results, it is clear that PSA ranks 1st in all indicators. It is noteworthy that GJO ranks 2nd in the mean value metric; SOA performed equally well, finishing 2nd in the index of the best value. On balance, PSA has an advantage on this problem.

Table 17
Optimization results for the car side impact design problem.

Algorithms	Mean	Best	Std
PSA	**2.3036235197E + 01**	**2.2843668419E + 01**	**2.0341880798E-01**
FHO	2.5647027367E + 01	2.3526488071E + 01	1.1665427675E + 00
GJO	2.3309867613E + 01	2.2858590816E + 01	2.6964487949E-01
AOS	2.3603065382E + 01	2.2895684714E + 01	4.7103439820E-01
TSA	2.3956436940E + 01	2.3031954928E + 01	4.6614652322E-01
SOA	2.3493404035E + 01	2.2871389050E + 01	2.0606057368E-01
HHO	2.3481380618E + 01	2.3102967893E + 01	4.4709116450E-01
SHO	2.6790274941E + 01	2.5777801548E + 01	6.4599950164E-01

Bolded values represent the smallest values.

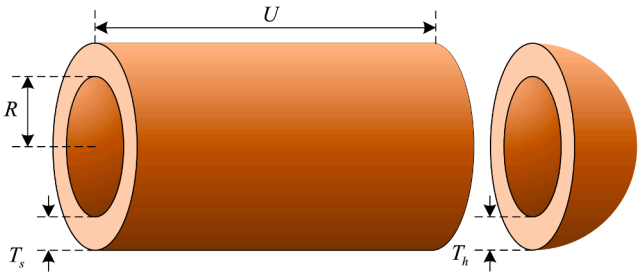


Fig. 13. Pressure vessel design.

Table 18
Optimization results for the pressure vessel design problem.

Algorithms	Mean	Best	Std
PSA	**6.3608293580E + 03**	**6.0597143350E + 03**	**3.0086501236E + 02**
FHO	6.8310440695E + 03	6.2215972777E + 03	3.9281412998E + 02
GJO	6.5811475362E + 03	6.0610103138E + 03	6.9242955276E + 02
AOS	7.1488590074E + 03	6.0697396160E + 03	6.7343897159E + 02
TSA	7.4721637295E + 03	6.1868191945E + 03	8.6317629409E + 02
SOA	6.6731102933E + 03	6.0598867458E + 03	7.3543782974E + 02
HHO	7.6451315336E + 03	6.0607265336E + 03	2.7347940774E + 03
SHO	1.5847314043E + 04	8.6455851841E + 03	5.2242457963E + 03

Bolded values represent the smallest values.

6.6. CEC2006 constrained real-parameter optimization-G10

The 2006 IEEE Conference on Evolutionary Computation (CEC) proposed 24 constrained optimization functions (Liang et al., 2006). Among these functions, function 10 (CEC2006-G10) is very challenging because of the eight decision variables and six constraints included. The mathematical expression of CEC2006-G10 is given in Eq. (20).

Consider:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$$

Minimize:

$$f(\mathbf{x}) = x_1 + x_2 + x_3$$

Subject to:

$$\begin{aligned} g_1(\mathbf{x}) &= -1 + 0.0025(x_4 + x_6) \le 0 \\ g_2(\mathbf{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \le 0 \\ g_3(\mathbf{x}) &= -1 + 0.01(x_8 - x_5) \le 0 \\ g_4(\mathbf{x}) &= -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0 \\ g_5(\mathbf{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0 \\ g_6(\mathbf{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0 \end{aligned} \tag{20}$$

Variable range:

$100 \le x_1 \le 10000, 1000 \le x_2, x_3 \le 10000, 10 \le x_i \le 1000 (i = 4, \ldots, 8)$.

Table 19
Optimization results for the CEC2006-G10.

Algorithms	Mean	Best	Std
PSA	**8.2871056839E + 03**	**7.0517752552E + 03**	1.4944255193E + 03
FHO	3.4156340184E + 04	1.0516310307E + 04	2.7201548866E + 04
GJO	1.1822876855E + 04	7.6305087866E + 03	1.9473566101E + 04
AOS	2.3514583340E + 04	8.2089194013E + 03	3.9281390590E + 04
TSA	7.9071213678E + 04	1.0058020435E + 04	9.6031252423E + 04
SOA	9.3893219897E + 03	7.9594002494E + 03	**8.3687414225E + 02**
HHO	2.3599956885E + 05	1.0068479811E + 04	2.1261926006E + 05
SHO	1.0002528206E + 06	3.9415657988E + 05	3.3144912786E + 05

Bolded values represent the smallest values.

The optimization results of CEC2006-G10 are shown in Table 19. The comparison shows that PSA has the smallest performance in mean and optimal values; PSA is inferior to SOA in standard deviation and ranks 2nd. Although SOA ranks 1st in standard deviation metrics, the other metrics are much worse than PSA. Therefore, it can be verified that PSA is more competitive in this problem.

7. Conclusion

In this paper, a novel metaheuristic algorithm called PID-based search algorithm (PSA) is designed based on the classical algorithm in the field of process control: the PID algorithm. The algorithm is based on an incremental PID that converges the entire population to the optimal state by adjusting the system deviations.

Advanced metaheuristic algorithms of recent years are selected for comparison with PSA, which include FHO, GJO, AOS, TSA, SOA, HHO, and SHO. The optimization capabilities of PSA are evaluated by CEC2017 benchmark test functions and six constrained optimization problems. Moreover, a complete computational cost and complexity analysis is performed for PSA and its competitors. In addition, some of the winners on the IEEE CEC2017 competition and some powerful algorithms that performed well on the CEC2017 benchmark set are used for comparison with PSA. The core results obtained from this study are summarized as follows:

(1) For most functions, PSA obtains better results than the comparison algorithms.

(2) Compared to the comparison algorithm, PSA converges to better results faster for most functions, especially for functions F1, F10, F12, F19, and F30.

(3) Most of the p-values obtained on Holm test (PSA vs. competitors) are less than 0.05.

(4) Kruskal-Wallis and Friedman test results show that PSA ranks first on the CEC 2017 benchmark test functions compared to the competitors.

(5) PSA has some advantages in terms of computational cost and complexity while guaranteeing optimization results.

(6) PSA is competitive in some ways compared to the winner at the IEEE CEC2017 competition and the powerful algorithms on the suite.

(7) PSA shows good optimization capabilities on constrained optimization problems.

The proposed PSA can be seen as a case where the PID algorithm is successfully designed as a metaheuristic algorithm, which provides a new search operation. Taking this paper as an inspiration, some other versions of PID algorithm can be tried to be designed as metaheuristics, which will provide more new search operations for the metaheuristics. Moreover, Proportion, Integral, and Differential regulation can be designed as optimization strategies that can be combined with other metaheuristics to improve optimization performance.

For the PSA itself, the Proportion, Integral, and Differential coefficients of the PSA are constant rather than adaptively adjusted, which may decrease the optimization performance of the PSA. Therefore, the proportional, integral, and differential coefficients of the PSA are to be further investigated, which may be designed as an adaptive regulation. Moreover, all the adaptive adjustment factors in the algorithm can be improved to a more appropriate degree. In addition to that, the design of zero output is free. Thus, there can be various designs of zero output to improve the performance of PSA. Currently, hundreds of metaheuristic algorithms have been proposed, which allow PSA to choose to combine with suitable algorithms to improve its own performance. In addition, PSA can be designed as a multi-target version and a binary version.

In terms of applications, PSA can be tried for solving neural networks, structural optimization, feature selection and some real-world problems (e.g., path planning and location problems).

Declaration of Competing Interest

The author declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the 2022 Liaoning College Students' Innovative Entrepreneurial Training Plan Program (Project Number: S202210147033).

Appendix A. . Brief description of PID algorithm

Assume that the target value of the controlled system is V_t and the actual value at the moment t is $V(t)$. Then the deviation of the control system at moment t is $E(t) = V_t - V(t)$. The formula for the Proportion can be derived as

$$P(t) = k_p E(t) + P_0 \tag{A1}$$

where $P(t)$ is the output value of the Proportion under the moment t; k_p is a factor that serves to adjust the deviation value of the controlled system; P_0 is a constant that serves to prevent the output value from going to zero.

Integral considers the past state of the controlled subject. The output value $I(t)$ of the Integral at moment t is calculated as

$$I(t) = k_i \int_0^t E(t)\ \mathrm{d}t + I_0 \tag{A2}$$

where k_i and I_0 are a coefficient and a constant, respectively. These have similar roles to k_p and P_0.

Differential takes into account the trend of the deviation. The output value $D(t)$ of the Integral at moment t is expressed as

$$D(t) = k_d \frac{\mathrm{d}E(t)}{\mathrm{d}t} + D_0 \tag{A3}$$

where k_d is a coefficient and D_0 is a constant. These have similar functions to k_p and P_0.

The PID algorithm is the synthesis of Eqs. (A1) to (A3). Thus, the conventional PID algorithm can be expressed as

$$O(t) = P(t) + I(t) + D(t) + O_0 \tag{A4}$$

where $O(t)$ is the control quantity output by the PID controller at the moment t; O_0 is the sum of P_0, I_0 and D_0. Expressing Eq. (A4) specifically as

$$O(t) = k_p E(t) + k_i \int_0^t E(t)\ \mathrm{d}t + k_d \frac{\mathrm{d}E(t)}{\mathrm{d}t} + O_0 \tag{A5}$$

In practice, only discrete data returned by the sensors are usually available. Therefore, the discretized form of the conventional PID algorithm (positional PID algorithm) is

$$O(t) = k_p E(t) + k_i \sum_{i=0}^{t} E(i) + k_d (E(t) - E(t-1)) + O_0 \tag{A6}$$

The incremental PID algorithm is then obtained as:

$$\Delta O(t) = k_p (E(t) - E(t-1)) + k_i E(t) + k_d [E(t) - 2E(t-1) + E(t-2)] \tag{A7}$$

References

Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering, 158*, Article 107408.

Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE Congress on Evolutionary Computation* (pp. 4661-4667). IEEE.

Ahmadianfar, I., Bozorg-Haddad, O., & Chu, X. (2020). Gradient-based optimizer: A new metaheuristic optimization algorithm. *Information Sciences, 540*, 131–159.

Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications, 181*, Article 115079.

Ahmadianfar, I., Heidari, A. A., Noshadian, S., Chen, H., & Gandomi, A. H. (2022). INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Systems with Applications, 195*, Article 116516.

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering, 376*, Article 113609.

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: A novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 157*, Article 107250.

Azizi, M. (2021). Atomic orbital search: A novel metaheuristic algorithm. *Applied Mathematical Modelling, 93*, 657–683.

Azizi, M., Talatahari, S., & Gandomi, A. H. (2022). Fire Hawk Optimizer: A novel metaheuristic algorithm. *Artificial Intelligence Review*, 1–77.

Arora, S., & Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. *Soft Computing, 23*(3), 715–734.

Bahreininejad, A. (2019). Improving the performance of water cycle algorithm using augmented Lagrangian method. *Advances in Engineering Software, 132*, 55–64.

Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., & Awadallah, M. A. (2022). White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *Knowledge-Based Systems, 243*, Article 108457.

Braik, M., Ryalat, M. H., & Al-Zoubi, H. (2022). A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the forty thieves. *Neural Computing and Applications, 34*(1), 409–455.

Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications, 198*, Article 116924.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation, 219*(15), 8121–8144.

Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences, 293*, 125–145.

Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software, 114*, 48–70.

Dhiman, G., & Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems, 165*, 169–196.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26*(1), 29–41.

Dehghani, M., Hubálovský, Š., & Trojovský, P. (2021). Northern goshawk optimization: A new swarm-based algorithm for solving optimization problems. *IEEE Access, 9*, 162059–162080.

Duman, S., Kahraman, H. T., Sonmez, Y., Guvenc, U., Kati, M., & Aras, S. (2022). A powerful meta-heuristic search algorithm for solving global optimization and real-world solar photovoltaic parameter estimation problems. *Engineering Applications of Artificial Intelligence, 111*, Article 104763.

Duman, S., Kahraman, H. T., & Kati, M. (2023). Economical operation of modern power grids incorporating uncertainties of renewable energy sources and load demand using the adaptive fitness-distance balance-based stochastic fractal search algorithm. *Engineering Applications of Artificial Intelligence, 117*, Article 105501.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures, 110*, 151–166.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures, 89*(23–24), 2325–2336.

Ghorbani, N., & Babaei, E. (2014). Exchange market algorithm. *Applied Soft Computing, 19*, 177–187.

Guvenc, U., Duman, S., Kahraman, H. T., Aras, S., & Katı, M. (2021). Fitness-Distance Balance based adaptive guided differential evolution algorithm for security-constrained optimal power flow problem incorporating renewable energy sources. *Applied Soft Computing, 108*, Article 107421.

Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences, 222*, 175–184.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97*, 849–872.

Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 94*, Article 103731.

Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems, 101*, 646–667.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures, 139*, 18–27.

Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software, 110*, 69–84.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.

Kahraman, H. T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms. *Knowledge-Based Systems, 190*, Article 105169.

Kahraman, H. T., Katı, M., Aras, S., & Taşci, D. A. (2023). Development of the Natural Survivor Method (NSM) for designing an updating mechanism in metaheuristic search algorithms. *Engineering Applications of Artificial Intelligence, 122*, Article 106121.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing, 4*(2), 87–112.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4, pp. 1942-1948). IEEE.

Kirkpatrick, S., Gelatt, C. D., Jr, & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence, 90*, Article 103541.

Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., Coello, C. C., & Deb, K. (2006). Problem definitions and evaluation criteria for the

CEC 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics, 41*(8), 8–31.

Li, S., Mu, K., Lin, W., & Sun, D. (2018, May). Research on path optimization of ant colony algorithm Improved Particle Swarm Optimization and Reverse Learning. In *2018 International Conference on Mechanical, Electrical, Electronic Engineering & Science (MEEES 2018)* (pp. 283-289). Atlantis Press.

Mohamed, A. W., Hadi, A. A., Fattouh, A. M., & Jambi, K. M. (2017, June). LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 145-152). IEEE.

Mezura-Montes, E., & Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *MICAI 2005: Advances in Artificial Intelligence: 4th Mexican International Conference on Artificial Intelligence, Monterrey, Mexico, November 14-18, 2005. Proceedings 4* (pp. 652-662). Springer Berlin Heidelberg.

Motevali, M. M., Shanghooshabad, A. M., Aram, R. Z., & Keshavarz, H. (2019). WHO: A new evolutionary algorithm bio-inspired by wildebeests with a case study on bank customer segmentation. *International Journal of Pattern Recognition and Artificial Intelligence, 33*(05), 1959017.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report, 826*, 1989.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*(2), 495–513.

Moosavi, S. H. S., & Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence, 86*, 165–181.

Meng, Z., & Pan, J. S. (2016). Monkey king evolution: A new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization. *Knowledge-Based Systems, 97*, 144–157.

Naruei, I., & Keynia, F. (2021). A new optimization method based on COOT bird natural life model. *Expert Systems with Applications, 183*, Article 115352.

Ong, K. M., Ong, P., & Sia, C. K. (2021). A carnivorous plant algorithm for solving global optimization problems. *Applied Soft Computing, 98*, Article 106833.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248.

Rechenberg, I. (1973). Evolution Strategy: Optimization of Technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart, 104*, 15–16.

Ray, T., & Saini, P. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization, 33*(6), 735–748.

Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation, 7*(4), 386–396.

Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*(5), 2592–2612.

Shen, L., Liu, Z., Zhang, Z., & Shi, X. (2009). Frame-level bit allocation based on incremental PID algorithm and frame complexity estimation. *Journal of Visual Communication and Image Representation, 20*(1), 28–34.

Shahrouzi, M., & Kaveh, A. (2022). An efficient derivative-free optimization algorithm inspired by avian life-saving manoeuvres. *Journal of Computational Science, 57*, Article 101483.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47.

Tanabe, R., & Fukunaga, A. S. (2014). In *July). Improving the search performance of SHADE using linear population size reduction* (pp. 1658–1665). IEEE.

Talatahari, S., Bayzidi, H., & Saraee, M. (2021). Social network search for global optimization. *IEEE Access, 9*, 92815–92863.

Varol Altay, E., & Alatas, B. (2020). Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review, 53*, 1373–1414.

Veysari, E. F. (2022). A new optimization algorithm inspired by the quest for the evolution of human society: Human felicity algorithm. *Expert Systems with Applications, 193*, Article 116468.

Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., Havlin, S., Da Luz, M. G. E., Raposo, E. P., & Stanley, H. E. (2000). Lévy flights in random searches. *Physica A: Statistical Mechanics and its Applications, 282*(1–2), 1–12.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.

Wan, S., Xu, P., Wang, K., Yang, J., & Li, S. (2020). Real-time estimation of thermal boundary of unsteady heat conduction system using PID algorithm. *International Journal of Thermal Sciences, 153*, Article 106395.

Youn, B. D., & Choi, K. K. (2004). A new response surface methodology for reliability-based design optimization. *Computers & Structures, 82*(2–3), 241–256.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82–102.

Yang, X. S. (2009). In *October). Firefly algorithms for multimodal optimization* (pp. 169–178). Berlin, Heidelberg: Springer.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.

Zervoudakis, K., & Tsafarakis, S. (2020). A mayfly optimization algorithm. *Computers & Industrial Engineering, 145*, Article 106559.

Zervoudakis, K., & Tsafarakis, S. (2022). A global optimizer inspired from the survival strategies of flying foxes. *Engineering with Computers*, 1–34.