Cluster Computing (2025)28:81
https://doi.org/10.1007/s10586-024-04761-4

Escape after love: *Philoponella prominens* optimizer and its application to 3D path planning

Yuansheng Gao[1,2] · Jinpeng Wang[1] · Changlin Li[1,3]

Received: 14 May 2024 / Revised: 28 July 2024 / Accepted: 5 August 2024

Abstract
According to the No Free Lunch theorem, we need to propose new metaheuristic algorithms with different optimization behaviors to better solve global optimization problems. Therefore, this paper presents a novel metaheuristic called *Philoponella prominens* Optimizer (PPO). It is inspired by the special mating behavior of the *P. prominens*, and simulates escape, sexual cannibalism and predation behaviors of males after mating. The superiority of the PPO is verified by comparing with seven recently proposed metaheuristics on CEC2017 benchmark functions. Furthermore, the Wilcoxon signed-rank sum test confirms significance in 93.60%, the Friedman ranking test places the PPO first, and the Kruskal–Wallis test verifies that the PPO's values is smallest on about 2/3 functions. Further tests on CEC2022 benchmark functions demonstrates PPO's competitiveness against six state-of-the-art metaheuristics. Additionally, the PPO is applied to four engineering problems and 3D UAV path planning problem, where it consistently outperforms competitors. Source codes of the PPO are publicly available at https://ww2.mathworks.cn/matlabcentral/fileexchange/171624-philoponella-prominens-optimizer.

Keywords Optimization · Metaheuristic · *Philoponella prominens* optimizer · 3D path planning

1 Introduction

The advancement of human science and technology has led to the proliferation of unmanned aerial vehicles (UAVs) across numerous fields, including inspection, monitoring, and rescue. The flexibility, convenience, and compact size of the UAVs have made them invaluable assets in these domains. In related fields such as power inspection, the inspection path of the UAV directly impacts the inspection efficiency. Consequently, the UAV inspection path planning has emerged as a crucial aspect in applications [1].

The path planning for the UAV is usually to identify an efficient path that enables the UAV to effectively avoid obstacles and minimize the total cost of the inspection path [2], etc. In this problem, there are usually multiple optimization objectives, which are typically weighted according to prior experience. This approach transforms the multi-objective optimization problem into a single-objective optimization problem.

Classical path planning methods rely on obstacles and the entire free space, and that the optimization objectives should include a reduction in path length. These algorithms include the artificial potential field algorithm [3], the A* algorithm [4], the rapidly-exploring random tree algorithm (RRT) [5], etc. Some path planning for complex terrain, which usually features irregular obstacles and rugged terrain, classical path planning methods do not hold advantages. The metaheuristic algorithm is gradually garnering attention from researchers and is being increasingly employed in diverse fields such as path planning, signal processing, and engineering design due to its straightforward structure, adaptability, and reduced likelihood of converging towards locally optimal.

In order to solve the path planning problem of the UAV in power inspection, many optimization methods have been

✉ Yuansheng Gao
gaoyuansheng2021@163.com

[1] College of Science, Liaoning Technical University, Fuxin 123000, China

[2] Present Address: College of Computer Science and Technology, Zhejiang University, Hangzhou 310000, China

[3] Present Address: iSoftStone Information Technology (Group) Co., Ltd., Xian 710000, China

Published online: 26 November 2024

proposed by previous researchers in this field. Recently, Lv et al. [6] proposed a new hybrid algorithm HGEOGWO based on the golden eagle optimizer (GEO) and the gray wolf optimizer (GWO) for solving the 3D path planning problem faced by multiple UAVs in power inspection. Moreover, Pan et al. [7] proposed a golden eagle optimizer (GEO-DLS) with a dual learning strategy. Personal instance learning can enhance the search capability of the GEO and reduce the possibility of the GEO falling into local optimal. Specular reflection learning can improve the optimization accuracy of the GEO and accelerate the convergence speed of the GEO. For the special needs of transmission line detection, Yang et al. [8] proposed a fusion chemical reaction optimization algorithm based on random molecules (RMCRO). This algorithm incorporates the concept of repellent-attractant rule, addresses the shortcomings of the chemical reaction algorithm, and accelerates convergence using the difference algorithm. In order to improve the efficacy of the algorithmic planning process, Zhang et al. [9] proposed an UAV path planning algorithm based on the improved harris hawks optimization. The algorithm exhibits high optimization accuracy, convergence speed, and robustness. Guo et al. [10] proposed a flight cost-based rapidly-exploring random tree star (FC-RRT*) that extends the standard rapidly-exploring random tree star (RRT*) to handle the safety requirements and flight constraints of UAVs in complex systems. It can be observed that, in addition to some classical path planning methods, metaheuristic algorithms are more active in this field. Bai et al. introduced a new mapping method called pareto-optimal-matching that solves multi-objective binary optimization problems and constructs a 3-objective sensor-weapon-target assignment model [11].

The presented path planning models illustrate the potential of metaheuristic algorithms in optimizing complex systems. Nevertheless, the application of metaheuristic algorithms is not confined to path planning. For problems that cannot be solved by traditional methods, metaheuristic algorithms are equally applicable. In particular, for black-box models, metaheuristic algorithms focus exclusively on the inputs and outputs of the problem. This property makes metaheuristic algorithms equally efficacious in hyperparameter optimization [12] and engineering simulation [13].

For metaheuristics, regulating the exploitation and exploration behavior of the algorithm is extremely challenging [14]. The exploration allows the algorithm to perform a broad search of the unknown space, facilitating global optimization and avoiding local optimal solutions. The exploitation allows the algorithm to gradually approach the optimal solution in the vicinity of the global optimum, which helps to improve the accuracy of the solution. According to the no free lunch (NFL) theorem [15], no single metaheuristic algorithm is suitable for all optimization problems, different algorithms apply to different optimization problems. In other words, the different characteristics exhibited in different problems make the metaheuristic algorithms perform differently in the optimization process, which leads to differences in the performance of the metaheuristic algorithms. Therefore, even though many metaheuristics already exist, we still need to develop new metaheuristics to solve the problems that existing metaheuristics cannot solve efficiently.

A review of the metaheuristics proposed in the past reveals that there is no metaheuristic algorithm that simulates the specific mating behavior of the *Philoponella prominens*. To this end, an attempt has been made to simplify and mathematically model the special mating behavior of the *P. prominens*. The CEC2017 benchmark functions and four engineering problems are used to test the PPO and compare it with seven metaheuristic algorithms. Also, the competitiveness of PPO is verified against six SOTAs using the CEC2022 benchmark functions. Besides, the PPO is applied to 3D path planning of the UAV in power inspection.

The principal contributions of this paper are outlined below:

(1) A novel metaheuristic algorithm is proposed: the *P. prominens* Optimizer (PPO).
(2) The performance of the PPO is tested on 41 benchmark functions.
(3) The PPO is used to solve 4 real-world engineering problems.
(4) The proposed algorithm is specifically used to solve a 3D path planning problem.

The remainder of this paper is as follows: related studies on metaheuristic algorithms are shown in Sect. 2. Section 3 provides a detailed description of the PPO. The capabilities of the PPO to solve benchmark problems are tested and analyzed in Sect. 4. Besides, Sect. 5 applies the PPO to engineering problems and discusses its capability to solve real-world problems. In Sect. 6, the PPO is applied to the UAV path planning. Moreover, in a summary of the paper and an outlook on future research for this study are presented in Sect. 7.

2 Related works on metaheuristics

Metaheuristic algorithms can usually be classified into four main categories: genetic evolution-based algorithms (GEA), physics/chemistry-based algorithms (PCA), mathematical formulae/theorems-based algorithms (MA), and swarm-based algorithms (SA). A brief overview of the

Table 1 A brief review of metaheuristic algorithms

Type	Algorithm	Inspiration	Year
GEA	Evolutionary programming (EP) [16]	Finite state machine	1966
	Evolution strategy (ES) [17]	Biological evolution	1973
	Genetic algorithm (GA) [18]	The concept of genetic evolution	1975
	Genetic programming (GP) [19]	The concept of biological evolution	1992
	Differential evolution (DE) [20]	Darwinian evolutionary theory	1997
	Wildebeests herd optimization (WHO) [21]	Wildebeest herding behavior	2019
	Human felicity algorithm (HFA) [22]	The quest for the evolution of human society	2022
	Love evolution algorithm (LEA) [23]	Stimulus-value-role theory	2024
PCA	Simulated annealing (SA) [24]	Principle of solid annealing	1983
	Electromagnetism-like mechanism (EM) [25]	Electromagnetic field charged particles	2003
	Chemical reaction optimization (CRO) [9]	Interaction between molecules in a chemical reaction	2012
	Artificial electric field algorithm (AEFA) [26]	Coulomb's law	2019
	Equilibrium optimizer (EO) [27]	Controlled volume mass balance	2019
	Transit search (TS) [28]	Exoplanet detection methods	2022
	Optical microscope algorithm (OMA) [29]	Microscope magnification	2023
	Energy valley optimizer (EVO) [30]	Regarding stability and different modes of particle decay	2023
	Prism refraction search (PRS) [31]	The refraction of light through a triangular prism	2024
MA	Sine cosine algorithm (SCA) [32]	Mathematical models of sine and cosine	2016
	Golden sine algorithm (GSA) [33]	Scan in the unit circle of the sine function	2017
	Gradient-based optimizer (GBO) [34]	Gradient-based Newtonian methods	2020
	Arithmetic optimization algorithm (AOA) [35]	Quadratic mixed operations	2021
	Weighted mean of vectors (INFO) [36]	Different weighted averaging rules for vectors	2022
	PID-based search algorithm (PSA) [37]	An incremental PID algorithm	2023
	Newton–Raphson-based optimizer (NRBO) [38]	The Newton–Raphson method	2024
SA	Ant colony optimization (ACO) [39]	Foraging behavior of ant colonies	1992
	Particle swarm optimization (PSO) [40]	Foraging behavior of flocks of birds	1995
	Harmony search (HS) [41]	The music-making process	2001
	Teaching–learning-based algorithm (TLBO) [42]	The teaching–learning process of the class room	2012
	Spotted hyena optimizer (SHO) [43]	Spotted hyena hunting behavior	2017
	Seagull optimization algorithm (SOA) [44]	Migration and aggressive behavior of seagulls	2019
	Harris hawks optimization (HHO) [45]	Harris hawks hunting for prey	2019
	Tunicate swarm algorithm (TSA) [46]	Behavior of the membrane-clustered	2020
	Immune plasma algorithm (IPA) [47]	The COVID-19 coronavirus	2020
	Wild horse optimizer (WHO) [48]	Social life behavior of wild horses	2021
	Social network search (SNS) [49]	The behavior of users when expressing their opinions	2021
	Golden jackal optimization (GJO) [50]	Collaborative hunting behavior of the golden jackal	2022
	Snake optimizer (SO) [51]	Mating behavior of snakes	2022
	Crayfish optimization algorithm (COA) [52]	The habits of crayfish as affected by temperature	2023
	Pied kingfisher optimizer (PKO) [53]	Hunting and symbiotic behaviors in the spotted kingfisher	2024
	Puma optimizer (PO) [54]	The intelligence and life of pumas	2024

metaheuristic algorithm is shown in Table 1. In addition, a classification of metaheuristic algorithms is given in Fig. 1.

Genetic evolution-based algorithms usually simulate the operations of chromosome selection, crossover, and mutation during genetic evolution. This is done with the aim of producing superior individuals and optimizing the overall population in an iterative manner. The popular algorithms include genetic algorithm (GA) [18] and differential evolution (DE) [20]. The GA simulates the biological evolutionary process by natural selection and genetics mechanism of biological evolution. The primary steps include coding, population initialization, calculation

Fig. 1 A classification of metaheuristic algorithms

of individual fitness values, evolutionary calculations (selection, crossover, variation), and decoding. The DE is a heuristic stochastic search algorithm based on population differences. The variance vector of the DE is generated from the parent differential vector and crossed with the parent individual vector to generate a new individual vector, which is then selected against its parent individuals. Furthermore, the recently proposed the love evolution algorithm (LEA) by Gao et al. [23] is a stimulus-value-role theory inspired evolutionary algorithm. The most distinctive feature of the LEA is its distinctive search operations, which include convolution between variables, multiple multiplication, and division for crossover and mutation.

Physics/chemistry-based algorithms usually perform individual position updates based on a range of forces, including electromagnetic, inertial, gravitational, and chemical reactions, etc. The most classical algorithm is simulated annealing (SA) [24]. The SA draws on the solid annealing principle and is a probability-based algorithm. It starts from a certain higher initial temperature, accompanied by the decreasing temperature parameter, and combines the probabilistic sudden jump property to find the global optimal solution of the objective function randomly in the solution space. In addition, the rime optimization algorithm (RIME) was proposed by Su et al. in 2023 [33]. The RIME simulates the growth rate of each frost bar by taking into account the effects of wind speed, freezing coefficients, area of the attached material, and growth time. By simulating the growth process of soft-time and hard-time of frosting time, its puncture mechanism is constructed to realize exploration and exploitation in the optimization process.

Mathematical formulae/theorem-based algorithms do not optimize by simulating some behavior unlike other metaheuristic algorithms, which perform optimization by purely mathematical operations. To illustrate, the sine cosine algorithm (SCA) [32] causes the candidate solutions to fluctuate outward or in the direction of the optimal solution based on the mathematical model of sine and cosine, thus performing the search. The golden sine algorithm (GSA) [55] introduces the golden mean coefficient and narrows the search space by the golden mean to approximate the optimal solution of the algorithm. In the meantime, the PID-based Search Algorithm [37] simulates the regulation process of the discretized incremental PID control algorithm. This algorithm converges the entire population to the optimal state by continuously adjusting the system deviation.

Most of the swarm-based algorithms simulate the collective behavior of groups. The most classical algorithms are the ant colony optimization (ACO) [39] and the particle swarm optimization (PSO) [40]. The ACO simulates the behavior of ants in discovering paths during foraging. Ants choose their walking direction based on the pheromone concentration during their walk and eventually reach the place where the food is. The ACO as an efficient population intelligence algorithm has been widely used in problems such as path planning and the TSP. The PSO simulates the foraging behavior of birds. In the PSO, each individual searches for the optimal solution under the influence of the population and the individual historical best individual. Most of these algorithms simulate the behavior of group attack, foraging, mating, etc. Furthermore, most of these groups are flocks of birds, insects, fauna, etc. This intelligence, which is based on group behavior, provides greater motivation for solving optimization problems and is therefore of wide interest to researchers. Meanwhile, swarm-based algorithms are inspired by a wider range of proposed algorithms, and are most active in metaheuristics. The black-winged kite algorithm [56] simulates behaviors such as migration and attack of the black kite. At the same time, a combination of the cauchy mutation strategy and the leader strategy is employed. This novel combination strikes a favorable balance between exploring global solutions and utilizing local information, enhancing both the search capability and convergence speed.

Moreover, numerous existing metaheuristic algorithms have been successfully enhanced and employed to address genuine engineering challenges. To illustrate, Adegboye et al. optimized the artificial electric field algorithm (AEFA) with diverse approaches [57–60]. For example, gaussian specular (GS) reflection learning and local escape operator (LEO) are added to the basic step of AEFA called GSLEO-AEFA. This improves its convergence speed and capability of avoiding local optimal for a given problem. Similarly, Gharehchopogh et al. improved different metaheuristics and implemented applications of the improved algorithms to engineering problems [61–67]. For instance, the chaotic quasi-oppositional farmland fertility algorithm

(CQFFA) enhances the exploration speed and convergence rate of the algorithm by integrating chaotic mapping and quasi-oppositional-based learning (QOBL) mechanisms into the farmland fertility algorithm (FFA) [68]. Aslan et al. improved the immune plasma algorithm (IPA) with different schemes to obtain pIPA [69] and rIPA [70] and implemented them in engineering optimization. Zhong et al. proposed the hierarchical multi-leadership SCA (HMLSCA) [71] and applied it for use in diagnosing COVID-19.

A metaheuristic algorithm may be applicable to different engineering optimization problems through different variants. The woodpecker mating algorithm (WMA), proposed by Parizi, simulates the mating behavior of woodpeckers. Female woodpeckers approach male woodpeckers based on the intensity of drumming [72]. Additionally, it has diverse variants of the algorithm and is specifically applied to different engineering problems. For example, OWMA for solving optimization problems [73], a hybrid SCA-WMA algorithm for solving optimization problems [74], WMA for optimal economic load dispatch [75], a hybrid WMA-WOA for global optimization and data classification [76], and GWMA on the GPU [77]. In light of these considerations, the presentation of the PPO is expected to contribute favorably to the advancement of metaheuristic algorithms. Similarly, algorithms such as cuckoo search [78, 79] and salp swarm algorithm [80] have been improved accordingly by Abed-alguni et al. and are also efficient for solving single-objective problems.

3 *Philoponella prominens* optimizer

This section describes the inspiration for the PPO and the concrete model abstracted through the escape by ejecting, sexual cannibalism and predation behavior of *P. prominens*.

3.1 Inspiration

The *P. prominens* is widely distributed in China, Japan and Korea [81]. For many animals, there is a risk of death with each mating. This is because females often eat males after mating, a behavior known in biology as "sexual cannibalism". Figure 2 illustrates this behavior of them.

Figure 3 shows the ejection behavior of males after they have finished mating. In the case of the *P. prominens*, the male is able to eject quickly after mating, thus avoiding being eaten by the female [82]. And those males that cannot stay away from the females are eaten by them. In this way, the more able ejector males have a better chance of passing on their good genes. This sexual selection is such that among the offspring, more and more males acquire stronger ejection skills. Thus, driven by this sexual selection, populations move towards greater strength.

Fig. 2 Mating behavior of the *P. prominens* [81]

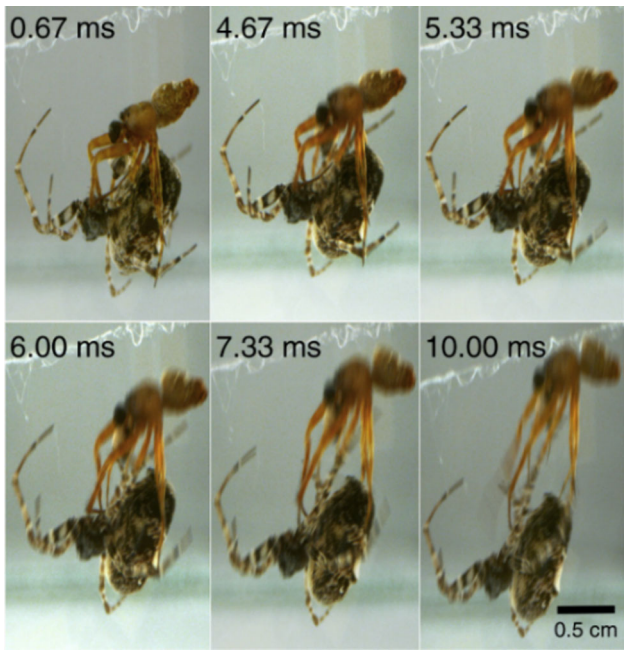


Fig. 3 Ejection behavior of males after mating [83]

For females, males that are unable to eject or are less able to eject are eaten by themselves. In this way, the more able ejector males have a better chance of passing on their good genes. This sexual selection is such that among the offspring, more and more males acquire stronger ejection skills. Thus, driven by this sexual selection, populations move towards greater strength.

Inspired by the above behavior and in order to facilitate subsequent model construction, the following rules are simplified and generalized in this paper.

(1) The positions of the males are the solution to the problem being solved.
(2) The positions of the females are randomized permutations of the males' historical best positions.
(3) Objective function value is defined as energy. For a minimization problem, the larger the energy the smaller the value of the objective function.
(4) The food's position represents the currently best solution.

The PPO mainly simulates the special mating behavior of *P. prominens*. This specific mating behavior serves as positive feedback that can enhance the quality of them to some extent. Consequently, within the context of a metaheuristic algorithm, we are able to continuously refine the quality of solutions by emulating the distinctive mating behavior of them. It is noteworthy that this specific mating behavior has contributed to the accelerated improvement in the quality of the *P. prominens* population. Consequently, in a manner analogous to the swarm intelligence demonstrated in behaviors such as predation and aggression, this particular mating behavior of them can also be regarded as an example of swarm intelligence.

3.2 Mathematical models

3.2.1 Initialization

A single-objective optimization problem usually consists of a set of decision variables, constraints, and an objective function. It may be useful to set the number of decision variables as d. The population size and the maximum number of function evaluations of the algorithm are denoted by n and T, respectively. Then the position of the male is

$$\boldsymbol{X}_{i,j} = \left(\boldsymbol{u}_j - \boldsymbol{l}_j\right) r_1 + \boldsymbol{l}_j,\ i = 1, 2, \cdots, n \tag{1}$$

where $\boldsymbol{X}_{i,j}$ denotes the jth dimension ($j = 1, 2, \ldots, d$) of the ith male. r_1 is a random number from 0 to 1; $\boldsymbol{u}_j$ and $\boldsymbol{l}_j$ are the upper and lower boundaries of the jth dimension.

Assume that the value of the objective functions for all males (candidate solutions) is $\boldsymbol{f}$. For a minimization problem, we define the energy $\boldsymbol{E}$ of the males to be

$$\begin{cases} \boldsymbol{F} = \max\{\boldsymbol{f}\} + \min\{\boldsymbol{f}\} - \boldsymbol{f} \\ \boldsymbol{E} = \dfrac{\boldsymbol{F}}{\max\{\boldsymbol{F}\} + \varepsilon} \end{cases} \tag{2}$$

where ε is a very small number greater than 0. It can be analyzed that the smaller the value of $\boldsymbol{f}$, the larger the value of $\boldsymbol{E}$.

In each iteration, a male will select a female for mating. The position of the corresponding female for each male can be expressed as

$$\begin{cases} \boldsymbol{c} = \text{randperm}(n) \\ \boldsymbol{Y} = \boldsymbol{H}_{\boldsymbol{c}} \end{cases} \tag{3}$$

where randperm(n) indicates the random permutation of the generated integers from 1 to n. $\boldsymbol{Y}$ is the positions of the females. $\boldsymbol{H}$ is the historical best positions for all males. $\boldsymbol{Y} = \boldsymbol{H}_{\boldsymbol{c}}$ means assigning $\boldsymbol{H}$ after row alignment according to $\boldsymbol{c}$ to $\boldsymbol{Y}^{(t)}$. Equation (3) indicates that the females' positions are a random permutation of the males' historical best positions. Figure 4 shows one-to-one matching of the males and females.

We define the average distance between males and females in all dimensions ω to be

$$\omega = \frac{1}{nd} \sum_{k=1}^{n} \|\boldsymbol{X}_k - \boldsymbol{Y}_k\|_2 \tag{4}$$

As the optimization proceeds, the distance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ will become smaller and smaller, which will facilitate the shift of Eq. (8) from exploration to exploitation.

3.2.2 Escape by ejecting

Males will eject immediately after mating to avoid being eaten by females. The ejection behavior of males can be seen as a projectile motion. If a male is assumed to have an initial position x_0 in a certain direction, a position x_1 after ejection, an ejection velocity is v, and the ejection angle is θ, then there is $x_1 = x_0 + (v\cos\theta)t$. For the two-dimensional plane, males eject at an angle of 0 to π. Inspired by this formula, this ejection behavior of males is expressed as

$$\begin{cases} v = \boldsymbol{E}_i \|\boldsymbol{X}_i - \boldsymbol{Y}_i\|_2 \\ \boldsymbol{X}_i = \boldsymbol{Y}_i + v\cos\boldsymbol{\theta} \end{cases} \tag{5}$$

where $\boldsymbol{\theta}$ is a vector $(1 \times d)$ of 0 to π random numbers. Males are driven by this formula to quickly congregate near females in a localized search. A schematic representation of this behavior is given in Fig. 5. ($^a\boldsymbol{X}_i, a = 1, 2, 3, 4$ indicate four potential positions of the ith male).

3.2.3 Sexual cannibalism and predation

After ejection, the females will eat the males with shorter ejection distances and give birth to their young, while the males with longer ejection distances will feed to maintain their strength. The distance of the ith male and female after escape by ejecting is given as

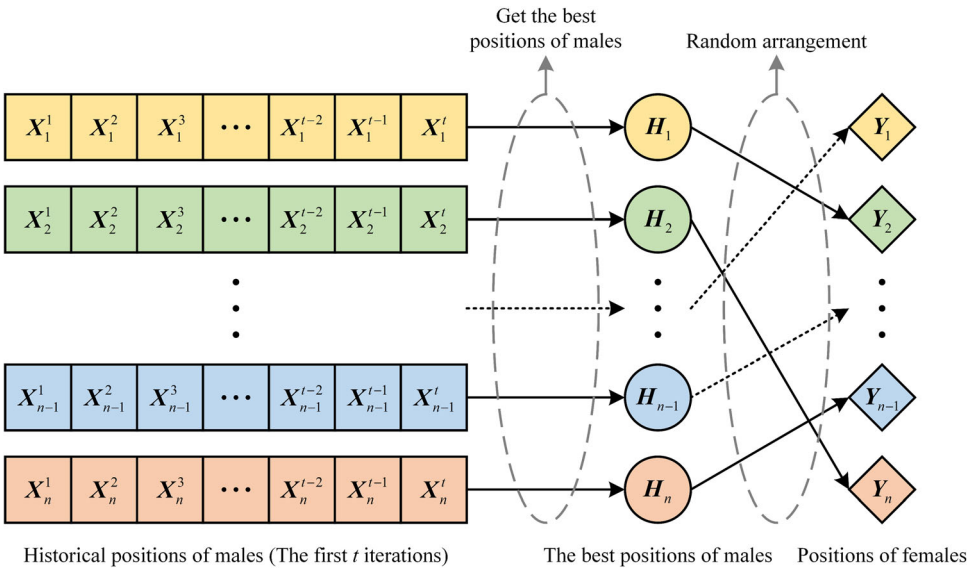


Fig. 4 One-on-one matching of males and females

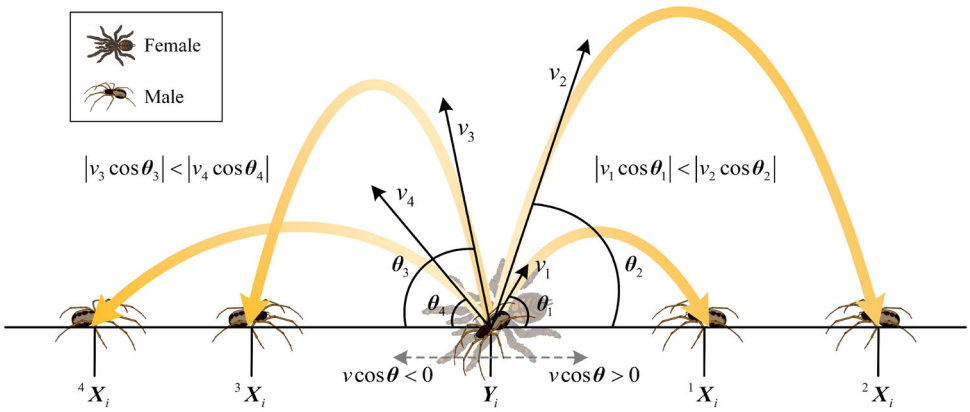


Fig. 5 Ejection behavior of males after mating

$$\boldsymbol{D}_i = \|\boldsymbol{X}_i - \boldsymbol{Y}_i\|_2 \tag{6}$$

After that, to determine whether the males survive, we define the survival factor σ, i.e. $\sigma = \sum_{i=1}^{n} \boldsymbol{D}_i / n \cdot \left[(1 - \frac{t}{T}) + 0.5\right]$. When the distance between the ith male and female is less than the survival factor, the male is eaten, and the female gives birth to a young. Before the males are eaten, females and males will engage in a chase and flee. Obviously, the greater the energy of the males, the greater the distance of escape, and the greater the change in position produced by the females in order to catch up with the males. Therefore, the position of the ith female after the chase is

$$\boldsymbol{Y}_i = \boldsymbol{Y}_i + r_2 \boldsymbol{E}_i (\boldsymbol{X}_i - \boldsymbol{Y}_i) \tag{7}$$

where r_2 is a random number from 0 to 1. It may be assumed that the ith female gives birth to young that will become a new male, and the position of the ith male is updated to be

$$\boldsymbol{X}_i = \boldsymbol{Y}_i + e^{1-t/T} \boldsymbol{L} \omega \tag{8}$$

where $\boldsymbol{L}$ is a vector of 1 row and d columns computed by the Lévy flight function and is denoted as

$$\boldsymbol{L} = \frac{\boldsymbol{h}}{|\boldsymbol{\eta}|^{1/\beta}} \left[\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta \times 2^{(\beta-1)/2}} \right]^{1/\beta} \tag{9}$$

where $\boldsymbol{h}$ and $\boldsymbol{\eta}$ are vectors of 1 row and d columns composed of random numbers obeying standard normal distribution. β is a constant taking the value of 1.5. The act of generating juveniles increases and maintains the diversity of the population. As the number of iterations increases, the value of $e^{1-t/T}$ gradually decreases, which facilitates the algorithm to shift from global search to local search. This behavior is illustrated in Fig. 6.

When the distance between the ith male and the female is greater than or equal to the average distance, the ith male escapes successfully and feeds to recover it strength. This behavior is defined as

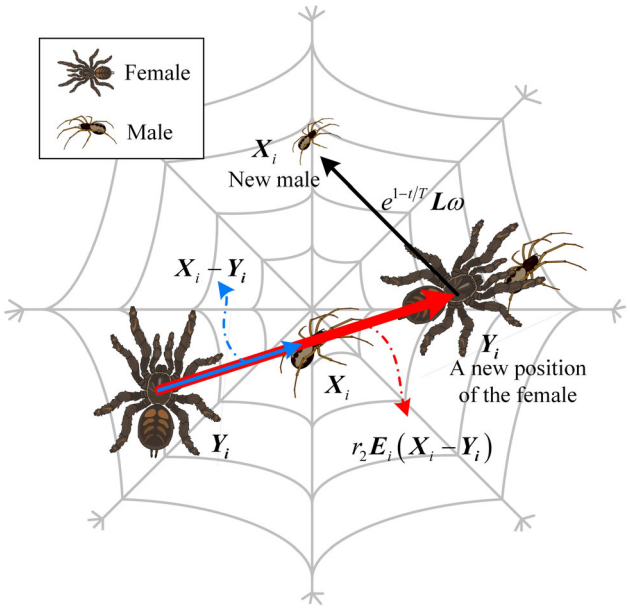


Fig. 6 The behavior of sexual cannibalism

Algorithm 1 *Philoponella prominens* optimizer

Input: The population size n; the maximum iterations T; the objective function $F(\cdot)$
Output: the food ξ (the historical best solution)
Initialize the number of iterations $t = 1$
Initialize the positions of males $\boldsymbol{X}$
Initialize the historical best positions of males $\boldsymbol{H}$
while $t < T$ **do**
  Calculate the energy $\boldsymbol{E}$ of the males using Eq. (2)
  Update the position of females $\boldsymbol{Y}$ using Eq. (3)
  Calculate the average distance ω between males and females in all dimensions using Eq. (4)
  % Escape by ejecting
  for $i = 1: n$ **do**
    Update the position of the ith male using Eq. (5)
    Calculate the distance between the ith male and female $\boldsymbol{D}_i$ using Eq. (6)
  end for
  Calculate the survival factor σ
  % Sexual cannibalism and predation
  for $i = 1: n$ **do**
    if $\boldsymbol{D}_i < \sigma$ **do**
      Update the position of the ith female $\boldsymbol{Y}_i$ using Eq. (7)
      Update the position of the ith male $\boldsymbol{X}_i$ using Eq. (8)
    else
      Update the position of the ith male $\boldsymbol{X}_i$ using Eq. (10)
    end if
  end for
  Update the positions of food ξ
  $t = t+1$
end while

$$\boldsymbol{X}_i = \boldsymbol{\xi} + \cos(r_3\pi)(\boldsymbol{X}_i - \boldsymbol{\xi}) \tag{10}$$

where r_3 refers to a random number from 0 to 1; $\boldsymbol{\xi}$ is the food (the historical best solution). Obviously, this formula can effectively improve the quality of the solution and is a local search mechanism. The spider does not reach the target location directly in the web, but has to zigzag along the silk (Fig. 7). Equation (10) uses the cosine function to portray this behavior.

3.3 Flowchart and pseudocode

Based on the above mathematical model, this paper proposes a novel metaheuristic algorithm inspired by the special mating behavior of the *P. prominens*. The pseudocode of the PPO is shown in Algorithm 1. A specific flowchart of the PPO is given in Fig. 8 to help intuitively understand the search process.

3.4 Time and space complexities

The time complexity of an algorithm is an important indicator to evaluate how good the algorithm is. The time complexity of the PPO mainly depends on the number of males (n), the maximum number of function evaluations (T) and the number of variables (d). It is evident that the time complexity required for the initialization is O(nd). In T loops, fitness computation and position updating need to be performed for each individual, and the total time required for these operations is O(ndT). The final time complexity of the PPO is O($nd + ndT$). By simplifying the result, one obtains O(ndT). Furthermore, the storage of males in the location occupies the main space and requires O(nd). Therefore, the space complexity of the PPO is O(nd).

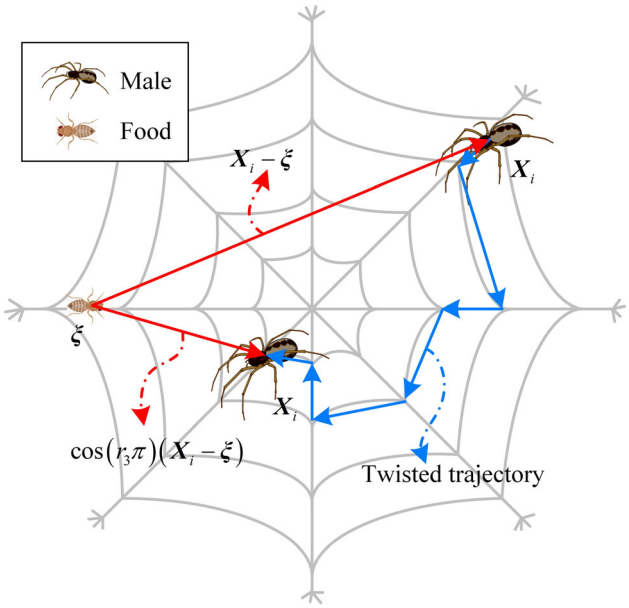


Fig. 7 Predation behavior of males

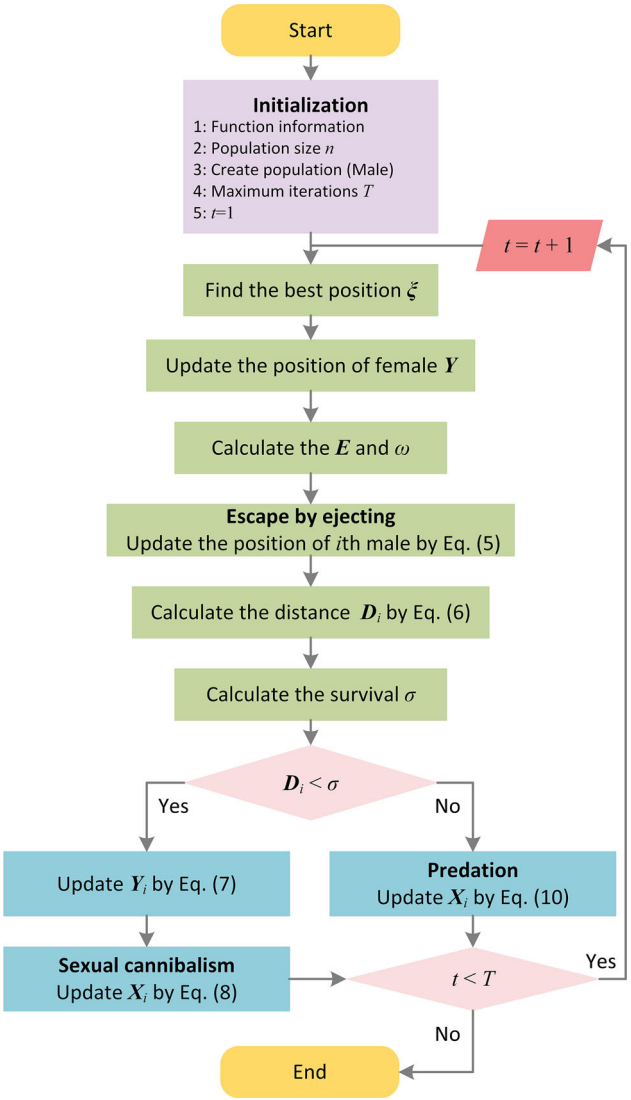


Fig. 8 Flowchart of the PPO

3.5 Differences between the PPO and some similar algorithms

Many of the metaheuristic algorithms inspired by the mating behavior of animals are unique, like the PPO. To illustrate the distinction between the PPO and existing metaheuristic algorithms of a similar nature, we will provide a brief overview below. The mating metaheuristic algorithms being compared includes the snake optimizer (SO) [51], the wild horse optimizer (WHO) [48], the water strider algorithm (WSA) [84], and the firefly algorithm (FA) [85].

3.5.1 Differences between the PPO and the SO

The SO is one of the more effective mating metaheuristics that have been developed in recent years. The SO simulates the diverse behaviors of snakes under varying temperatures and food conditions. The movement capabilities of the snakes in SO are all determined by the fitness values, while the movement capabilities of the spiders in PPO are determined by the distance on each dimension between the male and female spiders and the fitness values. In the search for food stage, the SO employs a straight-line wandering mode, whereas the PPO utilizes a circle-walking mode as illustrated in Fig. 7. Furthermore, the search mode judgment operators (temperature and food quantity) employed by the SO are determined by both the fitness value and the number of iterations. In contrast, the operator utilized by the PPO for this purpose is based on the Euclidean distance between the female and male spiders.

3.5.2 Differences between the PPO and the WHO

The WHO simulates the social grouping, bonding, grazing, and mating behaviors of wild horse populations. The most obvious difference between the PPO and the WHO is that the WHO has grouping behaviors. Typically, the WHO divides n horses into G social groups, and the groups exchange information with each other by mating. In contrast, there is no grouping phenomenon in the PPO, but the male and female spiders mate randomly, which is more favorable for information sharing between individuals. In addition, in the process of information transfer, new horses of the WHO are generated through the information crossover between the mares and stallions. New spiders of the PPO are then obtained from female spiders by certain positions changes. Furthermore, in the WHO, there is one guidance individual (the Stallion) for each group, so there are G guidance individuals totally in the WHO, but each guiding individual only has a guiding role in its group. However, the PPO has two guidance individuals including a best individual (food) and a random individual (female).

3.5.3 Differences between the PPO and the WSA

The WSA is inspired by a range of social behaviors in water striders, including territorial awareness, communication systems, foraging and mating. During mating, male water striders signal to females. Male water striders will move a distance that is r times or $(1 + r)$ times the Euclidean distance between male and female water striders (r is a random number between 0 and 1). In contrast, mating in the PPO is relatively simple, with each male spider randomly mating with a female spider, after which it escapes by ejecting according to a certain curve. It is worth noting that female water striders in the WSA are selected by a proportional fitness selection mechanism, while the female spiders in the PPO are obtained by randomly ranking male spiders at random. Another difference is that

the generation of new water striders in the WSA is determined by the upper and lower boundaries of the variable. In contrast, the new spiders of the PPO are generated in the neighborhood of female spiders. This ensures that the new spiders can inherit some of the information from their parents and can speed up the convergence of the algorithm.

3.5.4 Differences between the PPO and the FA

The characteristic of the FA is that it searches globally by simulating the brightness attraction behavior between fireflies. In the FA, all sexes of fireflies are considered solutions. Besides, the luminance magnitude of fireflies is determined by the distance between two fireflies. The PPO also calculates the distance between male and female spiders during the mating process, but in contrast to the FA, the distance of the PPO is used to determine the Sexual cannibalism or predation of spiders. Moreover, in order to avoid the locally optimal, the FA in the updating process added random vectors. However, when facing more complex optimization problems, a simple addition of random vectors may not be able to avoid local optimal efficiently. The behavior of escape by ejecting in the PPO give the algorithm more chances to get out of locally optimal.

4 Benchmark problems

In order to verify the performance of the proposed algorithm in this paper, 41 benchmark functions from CEC2017 [86] and CEC2022 [87] are selected to test the PPO and its comparative algorithms in this section. Then the results are analyzed and discussed. Among the CEC2017 benchmark functions, F1 and F3 are unimodal functions, F4 $\sim$ F10 are multimodal functions, F11 $\sim$ F20 are hybrid functions, and F21 $\sim$ F30 are composition functions. A brief description of these functions is given in Table 2. Similarly, there are unimodal function (H1), basic functions (H2 $\sim$ H5), hybrid functions (H6 $\sim$ H8) and composite functions (H9 $\sim$ H12) in the CEC2022 benchmark set. Table 3 provides a brief description for the CEC2022.

In this study, all algorithms are implemented using MATLAB R2023b on a computer with 64-bit Windows 11, and values such as the mean (Mean) and variance (Var) of each algorithm is analyzed by running each algorithm 51 times independently. To verify the superiority of the PPO, we selected seven recently proposed metaheuristic algorithms to be compared and analyzed with the PPO by taking into account factors such as the time of proposal of the algorithms and the performance of the algorithms. These algorithms include the African vultures optimization algorithm (AVOA) [88], the golden jackal optimization (GJO) [51], the tunicate swarm algorithm (TSA) [47], the love evolution algorithm (LEA) [23], the artificial gorilla troops optimizer (GTO) [89], the mountain gazelle optimizer (MGO) [90], and the nutcracker optimization algorithm (NOA) [91]. Set the population size of these algorithms to 50 and the maximum number of function evaluations to 500,000 (10,000* dimension). The dimension of the benchmark functions is set to 50. The settings of the remaining parameters are shown in Table 4. In addition, to verify the competition of the proposed algorithms, we analyze the optimization results of the PPO with six state-of-the-art (SOTA) algorithms on CEC2022 benchmark set including the JADE [92], the LSHADE-spacma [93], the LSHADE-cnEpSin [86], the CMA-ES [94], the A-LSHADE [93], and the LSHADE [95]. In this experiment, the maximum number of function evaluations is set to 1,000,000 and the population size is 50 (No Special Circumstances). The dimension on the CEC2022 benchmark functions is set to 20. Meanwhile, the PPO and the SOTAs are run independently for 30 times on each function. Other parameters of the PPO and the SOTAs are shown in Table 5.

4.1 Analysis of exploitation and exploration capabilities

Unimodal functions are real-valued functions with only one strictly local extreme value in the considered interval and are often used to test the exploitation capability of metaheuristic algorithms. The presence of a large number of local extremes in the multimodal functions place a higher demand on the exploration capability of algorithms. Hence, the multimodal functions are often used to evaluate the exploration capability of the metaheuristic algorithm.

In terms of exploration capability, Table 6 gives the test results of the PPO with its comparison algorithms on the unimodal functions (F1 and F3). In addition, the bolded numbers in the table indicate the minimum values. In terms of the Mean and Var, the PPO ranks first on F3. In addition, on F1, although the PPO does not get the first place, the accuracy of its Mean is 5 to 6 orders of magnitude higher than the algorithms that are worse than it. The experimental results verify that the PPO has excellent exploitation capability. Meanwhile, its excellent exploits may benefit from the constant reduction of energy $\boldsymbol{E}$, which weakens the spider's mobility. Table 6 gives the results of the PPO and its comparison algorithms on the multimodal functions (F4 $\sim$ F10). As far as the Mean is concerned, the PPO is ranks first on all except F10. Although, the PPO does not rank first on F10, it is in the same order of magnitude as the LEA which is ranks first, with a relative error of only about 0.0477. For the Var, the PPO ranks first on F6, F7, and F9; second on F4 and F10; and forth on F5 and F8. Although not all rank first, the relative error is only 0.1937 on F5,

Table 2 Overview of the CEC2017 benchmark functions

No.	Type	Function	Minimum
F1	Unimodal function	Shifted and Rotated Bent Cigar Function	100
F3	Unimodal function	Shifted and Rotated Zakharov Function	300
F4	Multimodal function	Shifted and Rotated Rosenbrock's Function	400
F5	Multimodal function	Shifted and Rotated Rastrigin's Function	500
F6	Multimodal function	Shifted and Rotated Expanded Scaffer's F6 Function	600
F7	Multimodal function	Shifted and Rotated Lunacek Bi_Rastrigin Function	700
F8	Multimodal function	Shifted and Rotated Non-Continuous Rastrigin's Function	800
F9	Multimodal function	Shifted and Rotated Levy Function	900
F10	Multimodal function	Shifted and Rotated Schwefel's Function	1000
F11	Hybrid function	Hybrid Function 1 (N = 3)	1100
F12	Hybrid function	Hybrid Function 2 (N = 3)	1200
F13	Hybrid function	Hybrid Function 3 (N = 3)	1300
F14	Hybrid function	Hybrid Function 4 (N = 4)	1400
F15	Hybrid function	Hybrid Function 5 (N = 4)	1500
F16	Hybrid function	Hybrid Function 6 (N = 4)	1600
F17	Hybrid function	Hybrid Function 6 (N = 5)	1700
F18	Hybrid function	Hybrid Function 6 (N = 5)	1800
F19	Hybrid function	Hybrid Function 6 (N = 5)	1900
F20	Hybrid function	Hybrid Function 6 (N = 6)	2000
F21	Composition function	Composition Function 1 (N = 3)	2100
F22	Composition function	Composition Function 2 (N = 3)	2200
F23	Composition function	Composition Function 3 (N = 4)	2300
F24	Composition function	Composition Function 4 (N = 4)	2400
F25	Composition function	Composition Function 5 (N = 5)	2500
F26	Composition function	Composition Function 6 (N = 5)	2600
F27	Composition function	Composition Function 7 (N = 6)	2700
F28	Composition function	Composition Function 8 (N = 6)	2800
F29	Composition function	Composition Function 9 (N = 3)	2900
F30	Composition function	Composition Function 10 (N = 3)	3000

Search range: $[-100, 100]^D$

Table 3 Overview of the CEC2022 benchmark functions

No.	Type	Function	Minimum
H1	Unimodal function	Shifted and full Rotated Zakharov Function	300
H2	Basic function	Shifted and full Rotated Rosenbrock's Function	400
H3	Basic function	Shifted and full Rotated Expanded Schaffer's F6 Function	600
H4	Basic function	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
H5	Basic function	Shifted and full Rotated Levy Function	900
H6	Hybrid function	Hybrid Function 1 (N = 3)	1800
H7	Hybrid function	Hybrid Function 2 (N = 6)	2000
H8	Hybrid function	Hybrid Function 3 (N = 5)	2200
H9	Composition function	Composition Function 1 (N = 5)	2300
H10	Composition function	Composition Function 2 (N = 4)	2400
H11	Composition function	Composition Function 3 (N = 5)	2600
H12	Composition function	Composition Function 4 (N = 6)	2700

Search range: $[-100, 100]^D$

Table 4 Setting the parameters of the PPO and its comparison algorithms

Algorithm	Parameter	Value	Algorithm	Parameter	Value
PPO	Parameter-less		LEA	Convergence constant $h_{\max}$	0.7
MGO	Parameter-less			Convergence constant $h_{\min}$	0
AVOA	L_1	0.8		Acceptance rate λ_c	0.5
	L_2	0.2		Adaptation rate λ_p	0.5
	w	2.5	GJO	Constant c_1	1.5
	P_1	0.6		Constant of Levy flight β	1.5
	P_2	0.4	TSA	Initial speed $P_{\min}$	1
	P_3	0.6		Subordinate speed $P_{\max}$	4
NOA	Probability δ	0.05	GTO	β	3
	Probability P_{a2}	0.2		W	0.8
	Probability P_{rp}	0.2		p	0.03

Table 5 Setting the parameters of the PPO and SOTAs

Algorithm	Parameter	Value	Algorithm	Parameter	Value
PPO	Parameter-less		JADE	c	0.1
CMA-ES	coordinate wise standard deviation σ	0.3		p	0.05
L-SHADE-spacma	L rate	0.8		CRm	0.5
	Maximum population size	50		Fm	0.5
	Minimum population size	4		Afactor	1
	Historical memory size	5	L-SHADE	Maximum population size	50
	p best rate	0.11		Minimum population size	4
	Archive rate	1.4		Historical memory size	5
	First class percentage	0.5		p best rate	0.11
LSHADE-cnEpSin	pd	0.4		Archive rate	1.4
	ps	0.55	AL-SHADE	Maximum population size	50
	Change freq	0.5		Minimum population size	4
	Maximum population size	100		Historical memory M_{CR}	0.5
	Minimum population size	4		Historical memory M_F	0.5
	Historical memory size	5		Historical memory size	6
	p best rate	0.11		p best rate	0.11
	Archive rate	1.4		Archive rate	2.6

which has the largest relative error. It is verified that the PPO has excellent exploration capability. This may be due to the fact that the PPO has an escape by ejecting during each iteration, allowing the PPO to explore the solution space more broadly.

4.2 Capability of avoiding locally optimal solutions

The hybrid and composition functions are considered to be extremely challenging functions. If an algorithm does not balance exploitation and exploration, then the algorithm can easily fall into local optimal and thus fail to obtain better optimization results. Thus, the analysis of the capability to avoid local optimal is particularly important. Tables 7 and 8 give the results of the PPO and its competitors on the hybrid and composition functions, respectively.

Considering only the Mean, the optimization results of the PPO rank first on F15, F16, F19, and F20 ∼ F29; second on F12, F14, F18, and F20; third on F13, F17, and F30; and only fourth on F11, where the result is rather poor. Considering the Var, the PPO ranks first on about half of the functions. Moreover, for the functions that do not rank first, the Var of the PPO is of the same order of magnitude as the values of the majority of the first ranked functions, with some of them differing by as much as an order of magnitude. In summary, the PPO has an excellent

Table 6 Test results of the PPO with its comparison algorithm on CEC2017 unimodal and multimodal functions

No.	Metrics	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
F1	Mean	8.5942E+03	3.0653E+10	6.1616E+10	1.5607E+04	4.3692E+03	5.0182E+03	**3.8268E+03**	1.4336E+11
	Var	7.6413E+07	4.0278E+19	1.4831E+20	1.3992E+08	4.2268E+07	3.2924E+07	**1.5628E+07**	5.1463E+19
F3	Mean	**3.0000E+02**	1.0015E+05	1.5309E+05	3.0000E+02	3.0184E+02	2.6684E+03	9.4545E+02	2.4543E+05
	Var	**2.3145E−17**	2.6016E+08	3.0892E+08	1.8975E−07	4.0333E+00	1.1342E+06	6.9483E+05	4.9069E+08
F4	Mean	**4.8609E+02**	4.1417E+03	1.3371E+04	5.3748E+02	5.2513E+02	5.1226E+02	5.2468E+02	3.7359E+04
	Var	2.6761E+03	2.0877E+06	2.5930E+07	2.9192E+03	3.1546E+03	**2.4855E+03**	2.5142E+03	2.5236E+07
F5	Mean	**6.3755E+02**	8.7297E+02	1.0145E+03	7.9996E+02	8.3231E+02	7.4534E+02	8.1432E+02	1.3535E+03
	Var	1.7846E+03	3.4560E+03	1.6107E+03	3.3950E+03	1.4950E+03	1.8649E+03	1.6943E+03	**1.1618E+03**
F6	Mean	**6.0792E+02**	6.4730E+02	6.8570E+02	6.2973E+02	6.5254E+02	6.2872E+02	6.3875E+02	7.0719E+02
	Var	**5.7968E+00**	6.7622E+01	2.1788E+01	1.0022E+02	5.5468E+01	5.1662E+01	3.2299E+01	2.4692E+01
F7	Mean	**9.7089E+02**	1.3429E+03	1.8306E+03	1.1130E+03	1.4829E+03	1.1430E+03	1.4097E+03	4.1592E+03
	Var	**2.0099E+03**	5.9450E+03	3.7236E+03	4.3686E+03	1.5129E+04	6.1534E+03	1.0884E+04	4.7468E+04
F8	Mean	**9.3525E+02**	1.1852E+03	1.3432E+03	1.0886E+03	1.1255E+03	1.0457E+03	1.1265E+03	1.6480E+03
	Var	1.4192E+03	3.5696E+03	1.1910E+03	3.3292E+03	2.2223E+03	1.8265E+03	1.3820E+03	**6.5222E+02**
F9	Mean	**1.6247E+03**	1.4947E+04	2.9134E+04	1.7982E+04	1.0725E+04	6.0311E+03	1.2264E+04	5.0726E+04
	Var	**1.4126E+05**	1.9421E+07	8.7350E+06	5.8221E+07	2.3756E+06	1.6817E+06	1.5001E+06	1.2727E+07
F10	Mean	7.4422E+03	9.6639E+03	1.2083E+04	**7.1030E+03**	8.5320E+03	7.2686E+03	7.8328E+03	1.4675E+04
	Var	5.0226E+05	3.7789E+06	1.6571E+06	7.0323E+05	1.8895E+06	9.8264E+05	1.2854E+06	**1.0497E+05**

Table 7 Test results of the PPO with its comparison algorithm on CEC2017 hybrid functions

No.	Metrics	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
F11	Mean	1.3541E+03	7.9230E+03	1.8790E+04	1.4507E+03	**1.2912E+03**	1.3091E+03	1.3166E+03	2.4556E+04
	Var	4.5805E+03	4.4367E+06	2.2669E+07	7.2144E+03	**2.7410E+03**	3.2926E+03	3.5587E+03	1.1883E+07
F12	Mean	8.9091E+05	6.6626E+09	3.5425E+10	1.0725E+07	**6.3486E+05**	1.7297E+06	4.3041E+06	4.5472E+10
	Var	**2.2628E+11**	1.4333E+19	2.0572E+20	2.7250E+13	3.2177E+11	1.1453E+12	4.3839E+12	3.7404E+19
F13	Mean	1.5265E+04	1.4399E+09	1.4855E+10	1.6418E+05	1.1346E+04	**9.2597E+03**	5.9618E+04	1.6484E+10
	Var	7.8349E+07	4.3921E+18	9.3207E+19	4.8470E+09	9.4481E+07	**7.5955E+07**	8.0691E+08	1.0214E+19
F14	Mean	1.3352E+04	1.1413E+06	1.2960E+07	8.7190E+04	**4.1936E+03**	1.0439E+05	6.1454E+04	1.0407E+07
	Var	7.5928E+07	1.7574E+12	2.3298E+14	2.4207E+09	**3.1830E+06**	9.6169E+09	2.4306E+09	1.9902E+13
F15	Mean	**1.1701E+04**	1.6035E+08	2.2114E+09	7.2090E+04	1.4200E+04	1.1844E+04	2.5192E+04	3.9955E+09
	Var	1.0756E+08	9.2859E+16	7.1777E+18	7.5786E+08	4.7873E+07	**4.7117E+07**	9.9921E+07	7.1442E+17
F16	Mean	**3.1951E+03**	3.5151E+03	4.9598E+03	3.5040E+03	3.4939E+03	3.3235E+03	3.7267E+03	7.5783E+03
	Var	2.2104E+05	1.4648E+05	5.4471E+05	1.8316E+05	2.1270E+05	1.4239E+05	2.3027E+05	1.4622E+05
F17	Mean	3.2178E+03	3.2919E+03	4.6840E+03	3.1967E+03	3.2602E+03	**3.0357E+03**	3.4971E+03	7.3148E+03
	Var	1.3761E+05	1.3313E+05	2.7325E+06	1.2747E+05	1.1948E+05	1.4143E+05	1.1453E+05	7.2884E+05
F18	Mean	7.9065E+04	8.0905E+06	4.9991E+07	4.9950E+05	**1.7310E+04**	2.3124E+05	2.7482E+05	5.9798E+07
	Var	9.7505E+08	1.3067E+14	3.2460E+15	4.7230E+10	**7.5605E+07**	1.7285E+10	2.1116E+10	3.2531E+14
F19	Mean	**1.3237E+04**	3.7577E+07	1.4150E+09	2.6138E+04	1.8096E+04	1.9590E+04	2.1891E+04	1.6340E+09
	Var	1.4809E+08	8.5594E+15	2.3435E+18	1.9224E+08	1.1943E+08	**6.7724E+07**	1.5781E+08	1.7133E+17
F20	Mean	3.0453E+03	3.1436E+03	3.6309E+03	3.1173E+03	3.1322E+03	**3.0312E+03**	3.2937E+03	4.0913E+03
	Var	6.9092E+04	9.1421E+04	1.0456E+05	9.0884E+04	8.2730E+04	7.3477E+04	1.0686E+05	**2.5537E+04**

Bold represents the minimum value

Table 8 Test results of the PPO with its comparison algorithm on CEC2017 composition test functions

No.	Metrics	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
F21	Mean	**2.4601E+03**	2.6733E+03	2.9340E+03	2.5837E+03	2.6140E+03	2.5048E+03	2.7198E+03	3.1446E+03
	Var	**1.4676E+03**	3.2789E+03	3.6577E+03	2.8009E+03	4.4606E+03	1.4966E+03	4.3456E+03	1.6745E+03
F22	Mean	**8.8302E+03**	1.1616E+04	1.4608E+04	8.8814E+03	1.0185E+04	8.9970E+03	1.0204E+04	1.6365E+04
	Var	8.6595E+05	5.5581E+06	**6.0076E+05**	1.5287E+06	6.9514E+06	4.6030E+06	7.7482E+05	1.5044E+05
F23	Mean	**2.8859E+03**	3.2211E+03	3.8069E+03	3.0324E+03	3.1972E+03	3.0070E+03	3.3512E+03	4.0994E+03
	Var	**1.2544E+03**	6.4672E+03	2.0832E+04	3.9274E+03	1.2254E+04	3.1146E+03	1.5120E+04	3.7981E+03
F24	Mean	**3.0483E+03**	3.4251E+03	4.1100E+03	3.2104E+03	3.3340E+03	3.1542E+03	3.6160E+03	4.3312E+03
	Var	**1.2086E+03**	1.0850E+04	3.2058E+04	6.7744E+03	1.3691E+04	3.5271E+03	2.1085E+04	8.5310E+03
F25	Mean	**3.0255E+03**	5.2881E+03	9.0969E+03	3.0157E+03	3.0709E+03	3.0610E+03	3.0607E+03	2.4778E+04
	Var	1.3258E+03	5.4184E+05	2.4152E+06	1.8126E+03	**7.4373E+02**	1.5127E+03	1.4884E+03	5.7448E+06
F26	Mean	**5.3541E+03**	8.6243E+03	1.4335E+04	6.9506E+03	6.1515E+03	8.4981E+03	8.6875E+03	1.8326E+04
	Var	**1.4186E+05**	5.4839E+05	1.2378E+06	3.0833E+05	1.2596E+07	3.1424E+06	8.2595E+06	1.3413E+06
F27	Mean	**3.3950E+03**	3.8941E+03	5.0274E+03	3.4702E+03	3.6815E+03	3.6122E+03	3.6320E+03	5.5328E+03
	Var	**4.8657E+03**	1.9148E+04	3.0624E+05	1.3450E+04	2.9278E+04	1.8148E+04	2.4046E+04	6.0386E+04
F28	Mean	**3.2906E+03**	5.7709E+03	8.6665E+03	3.2928E+03	3.3045E+03	3.3018E+03	3.3050E+03	1.3359E+04
	Var	4.3718E+02	4.0715E+05	1.7293E+06	8.0346E+02	4.1623E+02	4.6885E+02	**3.2267E+02**	8.3244E+05
F29	Mean	**4.3618E+03**	5.5805E+03	1.1635E+04	4.3918E+03	4.9786E+03	4.4680E+03	4.8145E+03	1.2654E+04
	Var	**1.0960E+05**	4.2359E+05	4.7571E+07	1.2413E+05	2.7794E+05	1.3019E+05	1.5311E+05	2.7766E+06
F30	Mean	1.0136E+06	2.8121E+08	2.0069E+09	1.7734E+06	**8.5860E+05**	9.3376E+05	1.5708E+06	3.0002E+09
	Var	4.5319E+10	5.2653E+16	4.4490E+18	2.1461E+11	3.0109E+10	**2.9295E+10**	2.0360E+11	3.3814E+17

Bold represents the minimum value

capability to avoid locally optimal solutions. The adaptive operator $e^{1-t/T}$ keeps decreasing during iteration in the Sexual cannibalism phase; the ξ operator keeps decreasing during solution convergence and in the Predation phase, these operations are key factors in the capability of the PPO to strike a good balance between exploration and exploitation. Meanwhile, considering the important feature of escape by ejecting of the PPO, its optimization results may show relative instability. But this loss is acceptable in order to obtain the excellent capability to avoid locally optimal.

4.3 Analysis of convergence capability

The metaheuristic algorithms approach the optimal solution by continuous iteration. In the early iterations, the solution may change considerably to explore the unknown space. Conversely, the solution will not change much later in the iteration. Figure 9 displays the 2D images, the search history, the trajectories, and the average fitness curves of the PPO on F1, F5, F8, F22, and F26.

The search history records the optimality search process of the algorithm in the search space. Higher densities represent exploitation and lower densities indicate exploration. As can be seen from Fig. 9, the PPO is able to explore the unknown space extensively and exploit it near the global optimum, gradually forcing it to the optimal solution.

The trajectory plot illustrates the fluctuation of the position of the algorithm solution during the iterative process, thus being able to reflect the convergence of the algorithm. In the early iterations, all solutions exhibit large fluctuations and gradually decrease as the number of iterations increases, eventually being able to reach a relatively stable state.

The average fitness and the convergence curve can reflect the convergence speed of an algorithm. The large fluctuations of the average fitness curve in the early iterations reflect the exploration of the PPO. After that, the average fitness curves are able to reach a smooth state. For the convergence curve, as the number of iterations increases, the convergence curves of all functions are able to reach a steady state and converge to the final result. It can be seen that the convergence of the PPO is faster and the optimization is more efficient.

The convergence curves of the PPO with its competitors on some functions are given in Fig. 10. It can be seen that the convergence speed of the PPO is better than other competitors, especially on F5, F6, F7, F8, F9, F16, F23 and F26, verifying that the PPO can better balance exploration

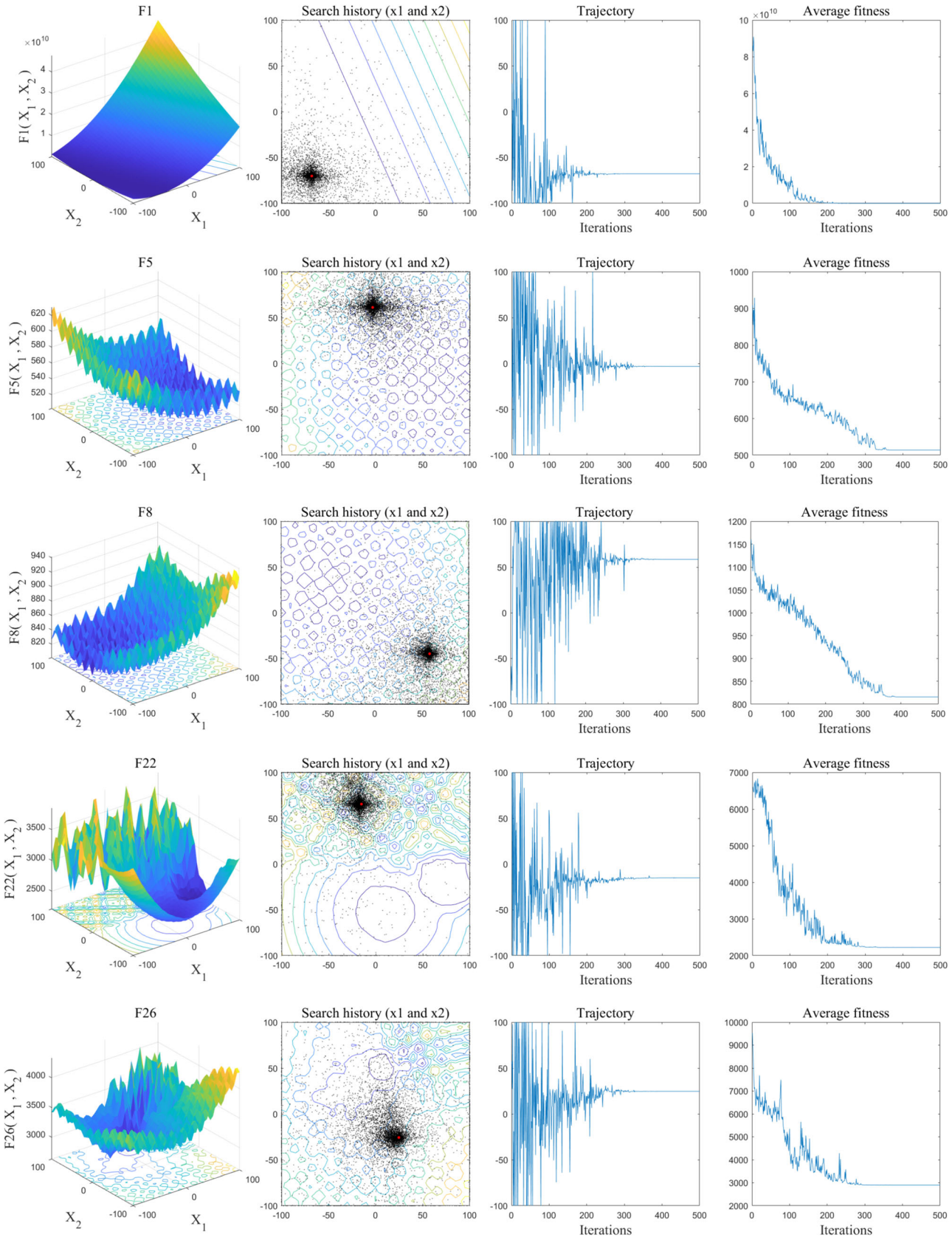


Fig. 9 2D benchmark function images, search history, trajectories, and average fitness curves

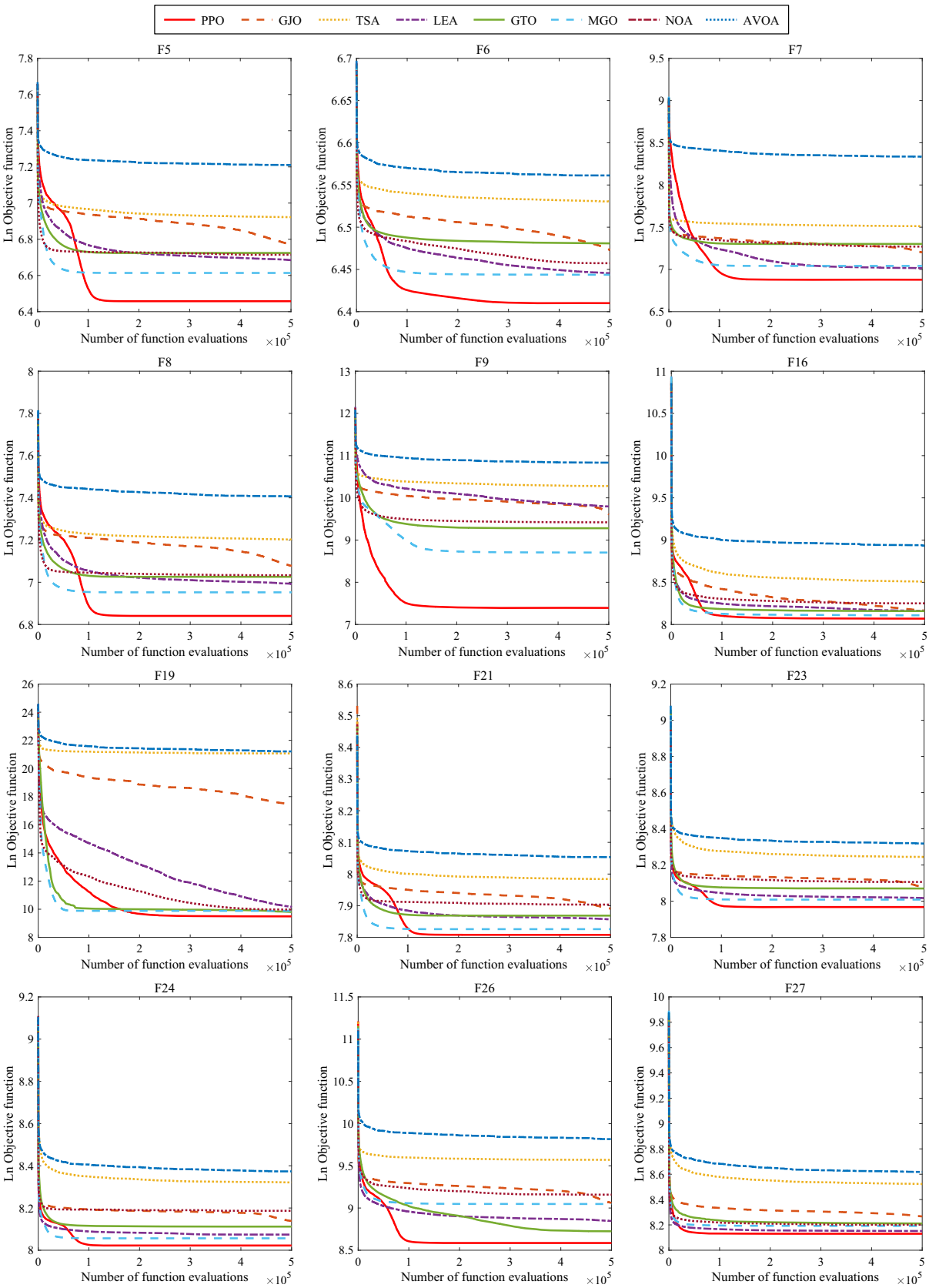


Fig. 10 Convergence curves of the PPO and its comparison algorithms on some test functions

and exploitation. On F5, all the algorithms perform poorly except for the PPO. On F9, the PPO shows a faster convergence rate. On F16, F21, and F27, in addition to the PPO, the MGO also perform well and achieve similar optimization results as PPO. On most functions, the NOA performs poorly, and all other algorithms achieve similar optimization results. Overall, the convergence speed of PPO is fast, which can verify the excellent performance of the PPO.

4.4 Statistical tests

The stochastic nature of the metaheuristic algorithm leads to a randomized nature of good and bad optimization results. The possibility of obtaining better results by chance cannot be ignored. For this reason, the Wilcoxon signed rank sum test was used for statistical testing at a significance level of 5%. Table 9 gives the p-values obtained by Wilcoxon signed rank sum test for the PPO and each comparison algorithm at a significance level of 5%. From the results, the vast majority of p-values are less than 5%, verifying the superiority of the PPO.

Table 10 gives the results of the Friedman ranking test of the PPO and comparison algorithm. The "FAR" in the table indicates the Friedman average ranking. The PPO ranks first; the MGO ranks second; the GTO ranks third; the LEA ranks fourth; the AVOA ranks fifth; the GJO ranks sixth; the TSA ranks seventh; and the NOA ranks eighth.

The mean of ranks obtained by Kruskal-Wallis test are given in Table 11. From the results, the PPO has the smallest mean value of rank on most of the functions. With the above three statistical methods, it is verified that the PPO is indeed superior to its competitive algorithms.

Table 9 The p-values obtained by Wilcoxon signed rank sum test at a significance level of 0.05

No.	PPO vs. GJO	PPO vs. TSA	PPO vs. LEA	PPO vs. GTO	PPO vs. MGO	PPO vs. AVOA	PPO vs. NOA
F1	5.1453E−10	5.1453E−10	3.2477E−03	1.0499E−02	2.0092E−02	1.5839E−03	5.1453E−10
F3	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10
F4	5.1453E−10	5.1453E−10	1.3110E−04	1.3472E−03	2.9657E−02	1.2621E−04	5.1453E−10
F5	5.1453E−10	5.1453E−10	6.1519E−10	5.1453E−10	1.3226E−09	5.4615E−10	5.1453E−10
F6	5.1453E−10	5.1453E−10	5.7967E−10	5.1453E−10	5.4615E−10	5.1453E−10	5.1453E−10
F7	5.1453E−10	5.1453E−10	1.1098E−09	5.1453E−10	5.4615E−10	5.1453E−10	5.1453E−10
F8	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	6.5284E−10	5.1453E−10	5.1453E−10
F9	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10
F10	1.1402E−08	5.1453E−10	4.3874E−02	4.1771E−06	2.4132E−01	3.5552E−03	5.1453E−10
F11	5.1453E−10	5.1453E−10	2.3099E−06	1.6894E−05	2.1086E−03	1.0785E−02	5.1453E−10
F12	5.1453E−10	5.1453E−10	5.1453E−10	5.8549E−03	2.4641E−05	6.5284E−10	5.1453E−10
F13	5.1453E−10	5.1453E−10	5.1453E−10	2.5074E−02	7.9164E−04	9.3064E−10	5.1453E−10
F14	5.1453E−10	5.1453E−10	7.3499E−10	1.0790E−08	2.1030E−09	2.2277E−09	5.1453E−10
F15	5.1453E−10	5.1453E−10	5.4615E−10	1.1105E−01	6.7317E−01	9.1483E−08	5.1453E−10
F16	4.7175E−04	6.1519E−10	1.2620E−03	2.7885E−03	8.2902E−02	7.1016E−07	5.1453E−10
F17	4.2560E−01	2.6464E−09	7.3578E−01	5.4234E−01	8.6758E−03	3.9192E−02	5.1453E−10
F18	5.1453E−10	5.1453E−10	5.1453E−10	5.4615E−10	8.2719E−10	6.1519E−10	5.1453E−10
F19	5.1453E−10	5.1453E−10	8.9246E−05	5.3491E−02	4.3783E−03	2.6221E−03	5.1453E−10
F20	1.8011E−01	8.6484E−09	7.2873E−01	1.2889E−01	6.6634E−01	1.0925E−06	5.1453E−10
F21	5.1453E−10	5.1453E−10	5.7967E−10	6.9272E−10	8.1285E−06	5.1453E−10	5.1453E−10
F22	3.3713E−08	5.1453E−10	3.8849E−01	3.5697E−05	2.1647E−02	5.1520E−08	5.1453E−10
F23	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10
F24	5.1453E−10	5.1453E−10	5.1453E−10	5.1453E−10	7.3499E−10	5.1453E−10	5.1453E−10
F25	5.1453E−10	5.1453E−10	2.1948E−01	2.0785E−08	1.8383E−05	1.8323E−06	5.1453E−10
F26	5.1453E−10	5.1453E−10	5.1453E−10	5.3491E−02	4.4164E−09	1.9851E−09	5.1453E−10
F27	5.1453E−10	5.1453E−10	2.6607E−04	6.9272E−10	1.0467E−09	6.1519E−10	5.1453E−10
F28	5.1453E−10	5.1453E−10	4.5899E−01	1.0704E−03	3.6635E−03	1.5839E−03	5.1453E−10
F29	5.4615E−10	5.1453E−10	7.8575E−01	6.7669E−07	2.8951E−01	7.1260E−06	5.1453E−10
F30	5.1453E−10	5.1453E−10	6.9272E−10	2.9671E−04	2.8961E−02	9.1414E−09	5.1453E−10

Table 10 Ranking the algorithm by Friedman test

No.	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
F1	4	6	7	5	2	3	1	8
F3	1	6	7	2	3	5	4	8
F4	1	6	7	5	4	2	3	8
F5	1	6	7	3	5	2	4	8
F6	1	5	7	3	6	2	4	8
F7	1	4	7	2	6	3	5	8
F8	1	6	7	3	4	2	5	8
F9	1	5	7	6	3	2	4	8
F10	3	6	7	1	5	2	4	8
F11	4	6	7	5	1	2	3	8
F12	2	6	7	5	1	3	4	8
F13	3	6	7	5	2	1	4	8
F14	2	6	8	4	1	5	3	7
F15	1	6	7	5	3	2	4	8
F16	1	5	7	4	3	2	6	8
F17	3	5	7	2	4	1	6	8
F18	2	6	7	5	1	3	4	8
F19	1	6	7	5	2	3	4	8
F20	2	5	7	3	4	1	6	8
F21	1	5	7	3	4	2	6	8
F22	1	6	7	2	4	3	5	8
F23	1	5	7	3	4	2	6	8
F24	1	5	7	3	4	2	6	8
F25	2	6	7	1	5	4	3	8
F26	1	5	7	3	2	4	6	8
F27	1	6	7	2	5	3	4	8
F28	1	6	7	2	4	3	5	8
F29	1	6	7	2	5	3	4	8
F30	3	6	7	5	1	2	4	8
FAR	1.6552	5.6207	7.0345	3.4138	3.3793	2.5517	4.3795	7.9655
Rank	1	6	7	4	3	2	5	8

4.5 Scalability analysis

Scalability is used to describe the capability of meta-heuristics to remain efficient when faced with increasing problem size and complexity. This section presents a scalability analysis, which compares the results of the PPO in different dimensions (10, 30, 50, and 100 dimensions). These results are illustrated in Fig. 11. The results indicate that the optimization performance of the PPO declines with the increase in dimensionality for the majority of functions. Furthermore, on F3 and F6, the optimization numerical of PPO do not exhibit a notable increase with the addition of dimensions. It is noteworthy that the performance of the PPO on F1, F10, F12, and F14 is considerably diminished due to the increase in dimensionality. Furthermore, on F15, F19, and F30, the PPO exhibits superior numerical results at higher dimensions compared to lower dimensions.

The optimization results of the PPO in distinct dimensions are illustrated in Table 12. In general, the Mean and the Var of the results increase to varying degrees as the dimensionality increases. Nevertheless, these increments remain within an acceptable range. In conclusion, the PPO exhibits relatively excellent scalability.

4.6 Sensitivity analysis

Two hyperparameters of the PPO are population size and maximum number of function evaluations. The remaining prerequisites are held constant, and the optimization results under different population sizes are analyzed to study the sensitivity of different populations to the PPO. The literature [86] defines the method of setting maximum number of function evaluations, i.e., MaxFEs = 10000*dimension.

Table 11 Kruskal–Wallis test results including mean of the ranks

No.	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
F1	138.9608	281.3137	331.6863	188.9412	94.37255	113.1373	104.5882	383
F3	26	281.9608	331.0392	77	128	227.1373	181.8627	383
F4	87.23529	281.5882	331.451	154.1176	132.2745	120.0784	146.2941	382.9608
F5	29.23529	244.7647	329.9412	160.1176	201.0784	95.68627	192.1765	383
F6	26.31373	231.3922	332.1373	123.5098	260.1569	111.7255	167.902	382.8627
F7	28.03922	197.3137	331.7451	97.09804	251.1569	110.7451	236.902	383
F8	28.29412	254.4118	330.098	144.9804	193.098	98.78431	203.3333	383
F9	26	222.8235	326.2941	241.3725	159.4314	80.4902	196.5882	383
F10	123.549	246.1765	323.451	97.15686	191.6667	105.5098	167.7451	380.7451
F11	145.8235	282.0196	339.3529	209.6667	77.70588	99.58824	107.2157	374.6275
F12	76.27451	281.9804	344.1569	225.0196	51.45098	115.8824	171.3725	369.8627
F13	100.6667	281.5882	349.2353	227.451	73.90196	59.54902	180.3137	363.2941
F14	81.15686	279.5294	345.902	185.7451	33	176.2353	171.9608	362.4706
F15	77.66667	283.098	336.6667	231.9412	99.52941	80.37255	154.1373	372.5882
F16	104.5098	162.4706	321.2549	162.7255	158.6667	127.8235	215.6667	382.8824
F17	158.0588	170.4118	310.451	148.098	161.549	109.2549	199.2941	378.8824
F18	79.41176	287.1765	339.6078	205.0196	26.80392	144.2157	186.8235	366.9412
F19	84.68627	283.902	347.1961	163.1569	120.3922	135.4706	136.2941	364.902
F20	132.8235	160.2745	293.7843	153.098	159.6078	127.1961	232.1373	377.0784
F21	38.03922	228.9216	331.9412	150.3333	177.7451	75.98039	250.2549	382.7843
F22	88.96078	237.4706	324.9216	98.17647	194.7843	131.2353	179.7451	380.7059
F23	28.78431	220.2745	333.1961	113.2745	204.8039	97.88235	256.1765	381.6078
F24	29.78431	221.1765	337.0392	124.6275	184.0588	92.84314	268.7843	377.6863
F25	83.27451	281.3529	331.6471	76.05882	163.1765	150.7255	166.7647	383
F26	60.4902	193.7059	332.7059	121.4314	120.0784	190.0196	235.2941	382.2745
F27	48.66667	262.8627	342.0784	87.35294	186.5294	159.3137	176.3137	372.8824
F28	85.37255	281.9216	331.2157	114.8235	157.1765	142.1961	140.4314	382.8627
F29	94.17647	258.0392	342.3529	98.5098	195.6275	113.0392	162.0588	372.1961
F30	102.3922	285.6471	342.5098	208.3137	62.2549	80.78431	186.2549	367.8431

Therefore, we do not consider the results of the maximum number of function evaluations.

In order to investigate the impact of population size on the performance of the PPO, we conducted an experiment on the CEC2017 benchmark functions, setting the population size to 10, 30, 50, 70, and 100, respectively. Figure 12 provides the convergence curves of some of the functions for different population sizes of the PPO. From the results, it is almost always the case that the small size of the population converges faster than the large size of the population. However, the accuracy of the results obtained for all population sizes of the PPO is relatively high. That is, changing the size of the population has little effect on the optimization results of PPO, and the results of PPO are relatively good at all population sizes.

4.7 Computational complexity

Computational complexity is an important metric for judging metaheuristic algorithms. The calculation of the computational cost of the algorithm is defined in the literature [86]. The specific calculation method is shown below.

(1) Calculate the program defined in Eq. (11) 1,000,000 times ($x = 0.55$), and record the running time as T_0.
(2) Evaluate F18 in the CEC2017 benchmark function 200,000 times under the condition of dimension 50. And the running time is recorded as T_1.
(3) Evaluate 200,000 times for F18 using the evaluated algorithm and record the time as T_2.
(4) Run step (3) independently 5 times and calculate the average value for T_2 to obtain T_{mean}.
(5) Calculate $(T_{\text{mean}}\text{-}T_1)/T_0$ and analyze the results.

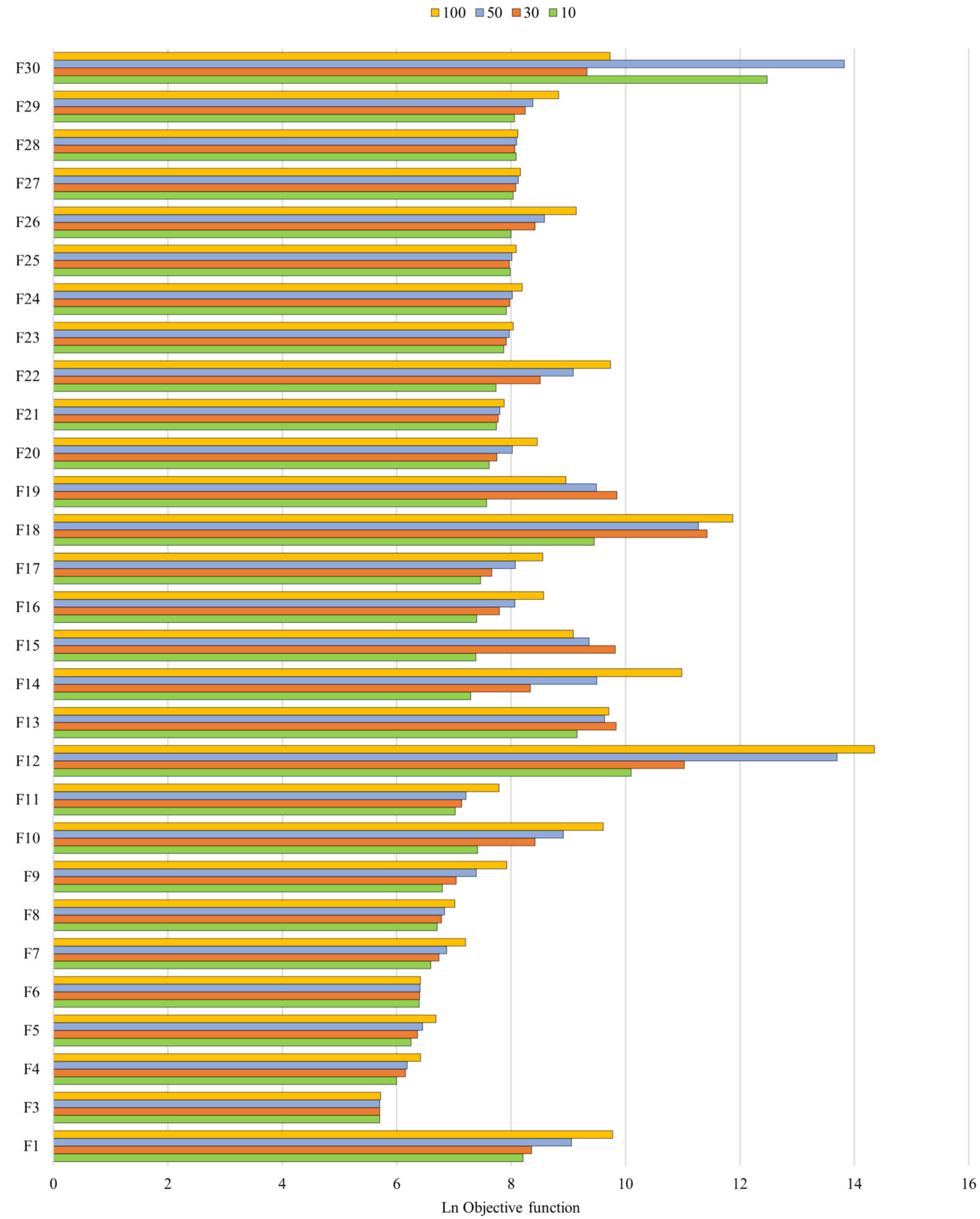


Fig. 11 The optimization results of the PPO in different dimensions on CEC2017 benchmark functions

Table 12 Optimization results of the PPO on different dimensions on CEC2017 benchmark functions

Mean					Var				
Dim	10	30	50	100	Dim	10	30	50	100
F1	3.6732E+03	4.2567E+03	8.5942E+03	1.7562E+04	F1	1.2500E+07	2.1518E+07	7.6413E+07	3.5876E+08
F3	3.0000E+02	3.0000E+02	3.0000E+02	3.0555E+02	F3	1.3305E−27	3.9154E−26	2.3145E−17	2.0050E+02
F4	4.0152E+02	4.7030E+02	4.8609E+02	6.1328E+02	F4	9.6138E+01	3.6958E+02	2.6761E+03	1.2445E+03
F5	5.2104E+02	5.7922E+02	6.3755E+02	7.9974E+02	F5	5.7272E+01	6.9589E+02	1.7846E+03	4.4022E+03
F6	6.0010E+02	6.0455E+02	6.0792E+02	6.1297E+02	F6	1.2073E−01	5.8606E+00	5.7968E+00	3.5830E+00
F7	7.3381E+02	8.4649E+02	9.7089E+02	1.3454E+03	F7	7.7887E+01	1.3568E+03	2.0099E+03	9.6305E+03
F8	8.2248E+02	8.8603E+02	9.3525E+02	1.1142E+03	F8	7.5334E+01	5.8729E+02	1.4192E+03	3.0977E+03
F9	9.0032E+02	1.1386E+03	1.6247E+03	2.7627E+03	F9	3.8072E+00	2.3721E+04	1.4126E+05	4.7774E+05
F10	1.6603E+03	4.5219E+03	7.4422E+03	1.4883E+04	F10	5.2456E+04	4.7240E+05	5.0226E+05	1.9556E+06
F11	1.1206E+03	1.2618E+03	1.3541E+03	2.4119E+03	F11	1.6109E+02	2.0410E+03	4.5805E+03	5.3942E+04
F12	2.4259E+04	6.1619E+04	8.9091E+05	1.7115E+06	F12	3.9287E+08	9.7327E+08	2.2628E+11	8.1234E+11
F13	9.4641E+03	1.8767E+04	1.5265E+04	1.6478E+04	F13	6.5034E+07	2.8075E+08	7.8349E+07	6.4385E+07
F14	1.4681E+03	4.1667E+03	1.3352E+04	5.8923E+04	F14	9.7378E+02	3.3563E+06	7.5928E+07	8.2687E+08
F15	1.6106E+03	1.8430E+04	1.1701E+04	8.8680E+03	F15	5.0046E+03	1.9980E+08	1.0756E+08	6.7852E+07
F16	1.6408E+03	2.4366E+03	3.1951E+03	5.2809E+03	F16	2.3474E+03	4.4420E+04	2.2104E+05	4.2589E+05
F17	1.7483E+03	2.1324E+03	3.2178E+03	5.1888E+03	F17	3.5124E+02	4.2246E+04	1.3761E+05	3.1323E+05
F18	1.2793E+04	9.1911E+04	7.9065E+04	1.4352E+05	F18	8.7482E+07	2.0978E+09	9.7505E+08	4.6663E+09
F19	1.9480E+03	1.9032E+04	1.3237E+04	7.7675E+03	F19	1.1702E+03	3.4141E+08	1.4809E+08	5.0730E+07
F20	2.0338E+03	2.3235E+03	3.0453E+03	4.7216E+03	F20	1.8361E+02	2.3069E+04	6.9092E+04	2.1109E+05
F21	2.3129E+03	2.3817E+03	2.4601E+03	2.6503E+03	F21	1.3908E+03	6.1603E+02	1.4676E+03	3.9740E+03
F22	2.2995E+03	4.9589E+03	8.8302E+03	1.6994E+04	F22	2.4296E+02	3.1750E+06	8.6595E+05	2.1995E+06
F23	2.6210E+03	2.7343E+03	2.8859E+03	3.1009E+03	F23	6.7760E+01	5.9921E+02	1.2544E+03	2.2492E+03
F24	2.7506E+03	2.9026E+03	3.0483E+03	3.6195E+03	F24	1.3308E+03	4.5356E+02	1.2086E+03	3.3251E+03
F25	2.9275E+03	2.8894E+03	3.0255E+03	3.2615E+03	F25	6.2577E+02	6.0536E+01	1.3258E+03	4.3017E+03
F26	2.9682E+03	4.5408E+03	5.3541E+03	9.3132E+03	F26	2.0585E+04	6.6364E+04	1.4186E+05	4.2238E+05
F27	3.0920E+03	3.2270E+03	3.3950E+03	3.5190E+03	F27	5.0884E+00	1.7226E+02	4.8657E+03	2.7023E+03
F28	3.2670E+03	3.1848E+03	3.2906E+03	3.3529E+03	F28	2.0170E+04	3.9912E+03	4.3718E+02	1.1385E+03
F29	3.1684E+03	3.8195E+03	4.3618E+03	6.8314E+03	F29	7.0525E+02	4.0492E+04	1.0960E+05	3.7314E+05
F30	2.6185E+05	1.1213E+04	1.0136E+06	1.6899E+04	F30	1.6454E+11	1.3022E+07	4.5319E+10	3.3652E+07

$$x = x + x; x = x/2; x = x \times x;$$
$$x = \sqrt{x}; x = \ln x; x = e^x; x = x/(x+2) \quad (11)$$

For $(T_{\text{mean}}\text{-}T_1)/T_0$, the smaller its value, the smaller the computational complexity of the validation algorithm. The computational cost information of all the algorithms is given in Table 13. From the results, the PPO is ranked third in terms of computational cost. The computational cost of the PPO is slightly worse than the three algorithms the GTO and AVOA. However, from the experimental results in the previous section, the optimization performance of the PPO is much better than this algorithm in CEC2017. Taken together, the computational cost of the PPO is not dominant, but given its optimization performance, this shortage is acceptable.

4.8 Competitive analysis

Table 14 illustrates the results of the PPO and SOTAs on the CEC2022 benchmark set, thereby substantiating that the PPO is competitiveness. The evaluation metrics include mean, variance, and minimum. The results demonstrate that PPO achieves the highest ranking on H1. Facing the SOTAs obtained after many improvements, we cannot expect that the PPO ranks first on all functions. Nevertheless, the results of the PPO consistently demonstrate a strong performance, with nearly all of them exhibiting a similar order of magnitude as the top-ranked result. While the PPO may not secure the top position in all indexes, its competitive edge remains evident when against the SOTAs.

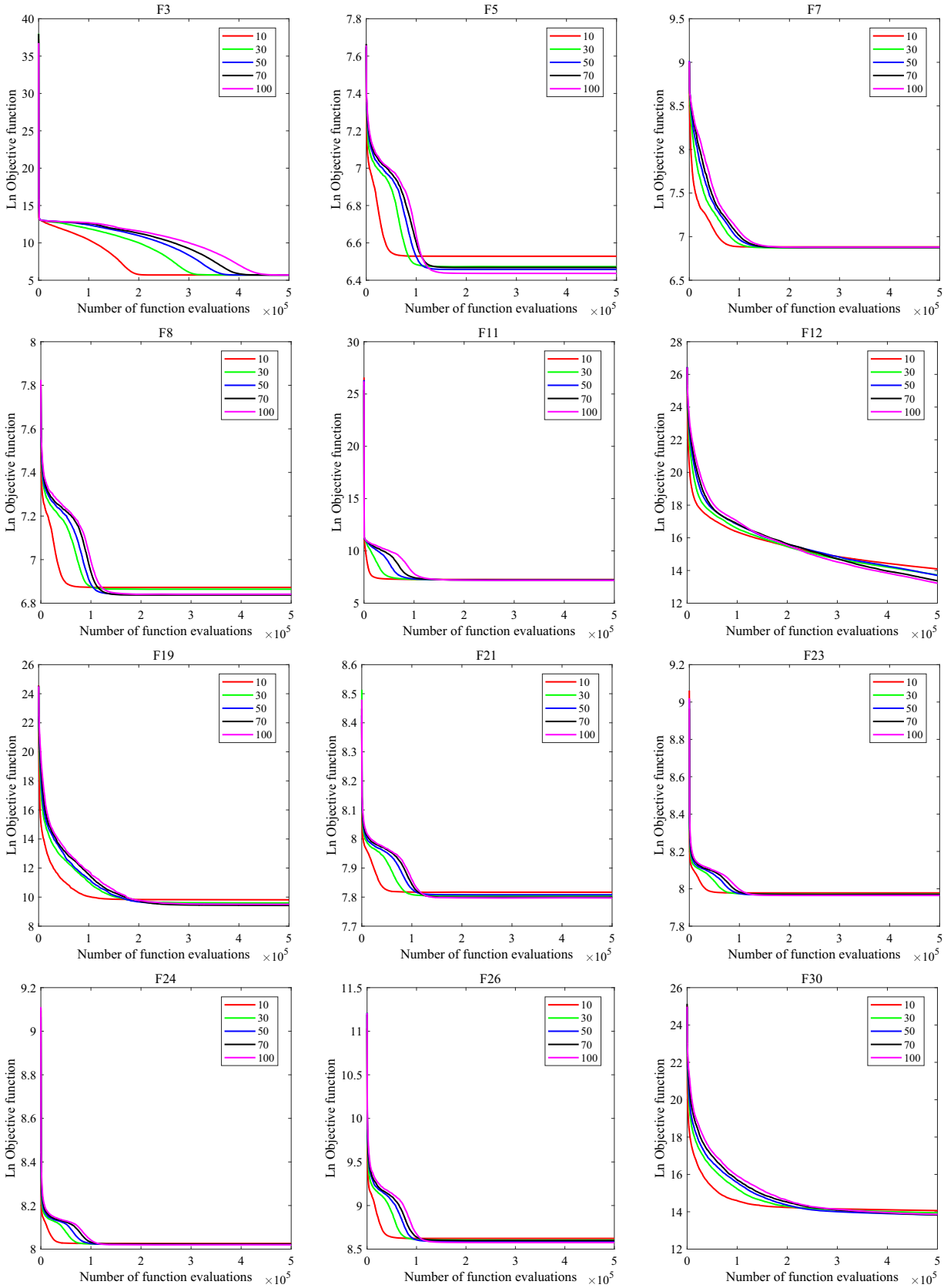


Fig. 12 Convergence curves of the PPO in different population size

Table 13 Computational costs for the PPO and its competitors

Algorithms	Properties	Results	Algorithms	Properties	Results
PPO	T_0	0.0589	GTO	T_0	0.0537
	T_1	0.9927		T_1	1.0030
	T_{mean}	2.0868		T_{mean}	1.6843
	$(T_{\text{mean}}\text{-}T_1)/T_0$	18.5690		$(T_{\text{mean}}\text{-}T_1)/T_0$	12.6755
GJO	T_0	0.0570	MGO	T_0	0.0532
	T_1	0.9767		T_1	0.9800
	T_{mean}	2.2707		T_{mean}	2.0220
	$(T_{\text{mean}}\text{-}T_1)/T_0$	22.7015		$(T_{\text{mean}}\text{-}T_1)/T_0$	19.5747
TSA	T_0	0.0541	AVOA	T_0	0.0535
	T_1	1.0201		T_1	1.0438
	T_{mean}	2.1010		T_{mean}	1.7674
	$(T_{\text{mean}}\text{-}T_1)/T_0$	19.9905		$(T_{\text{mean}}\text{-}T_1)/T_0$	13.5198
LEA	T_0	0.0539	NOA	T_0	0.0542
	T_1	0.9405		T_1	0.9541
	T_{mean}	2.5411		T_{mean}	2.0319
	$(T_{\text{mean}}\text{-}T_1)/T_0$	29.7150		$(T_{\text{mean}}\text{-}T_1)/T_0$	19.8998

5 Engineering problems

In order to verify the capability of the PPO to solve practical engineering problems, four engineering problems are selected to test the PPO: the cantilever beam design, I-beam vertical deflection design, corrugated bulkhead design, and gear train design. Set the population size to 100, the maximum number of function evaluations to 100,000, and run it 25 times independently. The obtained best value is compared with the current publicly available algorithms to verify the feasibility of the proposed the PPO to solve engineering problems.

5.1 Cantilever beam design problem

A cantilever beam [96] consists of a hollow build with five square cross sections, and its specific structure is shown in Fig. 13. The problem is a structural engineering design example for optimizing the weight of a cantilever beam of square section. Each cell is defined by one variable with constant widths, resulting in a total of five structural parameters. The thickness of the blocks is kept constant 2/3 in this problem. Its height (or width) is considered a decision variable (x_1, x_2, x_3, x_4, x_5), and the problem can be expressed as Eq. (12).

Minimize:

$$f(\boldsymbol{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5).$$

Subject to:

$$g(\boldsymbol{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0. \tag{12}$$

Variable range:

$$0.01 \le x_i \le 100,\ i = 1, \cdots, 5.$$

The best results of the PPO and competitors for cantilever beam design problem are shown in Table 15. From the results, the results of the PPO are better than the other algorithms. Meanwhile, the LEA, MGO and AVOA are more similar to the results of the PPO in terms of the best optimization results. In addition, the GTO has poor performance in this problem.

5.2 I-beam vertical deflection design problem

The I-beam vertical deflection design problem [97] is a typical engineering optimization problem as shown in Fig. 14. The optimization objective of this problem is to minimize the vertical deflection of the beam, taking into account the cross-sectional area and stress constraints given by the problem. The four design variables for this problem include: the width of flange (b), the height of section (h), the thickness of the web (t_w), and the thickness of the flange (t_f).The maximum vertical deflection of the beam is $PL^3/48EI$, where beam length (L) is 5200 cm and the modulus of elasticity (E) is 523.104 kN/cm^2. The objective function and constraints of the problem are as Eq. (13).

Consider:

$$f(\boldsymbol{x}) = \frac{5000}{x_3(x_2 - 2x_4)^3 \Big/ 12 + \left(x_1 x_4^3/6\right) + 2bx_4(x_2 - x_4/2)^2}.$$

Subject to:

Table 14 The results of the PPO and SOTAs on the CEC2022 benchmark functions

No.	Metrics	PPO	JADE	LSHADE_spacma	LSHADE_cnEpSin	A-LSHADE	LSHADE	CMA-ES
H1	Mean	3.0000E+02	2.5309E+04	3.0294E+03	3.0000E+02	3.0000E+02	8.9582E+02	3.0000E+02
	Min	3.0000E+02	1.0729E+03	3.0000E+02	3.0000E+02	3.0000E+02	3.0000E+02	3.0000E+02
	Var	0.0000E+00	4.8250E+07	2.4706E+07	1.0771E−28	9.6935E−28	7.3594E+06	1.1848E−27
H2	Mean	4.3937E+02	1.3028E+03	4.4465E+02	4.0398E+02	4.4811E+02	4.4853E+02	4.0040E+02
	Min	4.0000E+02	4.4916E+02	4.0000E+02	4.0013E+02	4.4490E+02	4.4490E+02	4.0000E+02
	Var	3.1303E+02	1.2926E+05	2.3961E+02	4.5382E+00	3.1391E+00	2.0277E+00	1.4304E+00
H3	Mean	6.0106E+02	6.5955E+02	6.0001E+02	6.0000E+02	6.0001E+02	6.0000E+02	6.6765E+02
	Min	6.0000E+02	6.0012E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.6589E+02
	Var	1.7176E+00	1.7301E+02	3.2414E−03	4.3082E−28	1.1380E−03	8.4602E−07	5.2245E−01
H4	Mean	8.6023E+02	9.6445E+02	8.1161E+02	8.2780E+02	8.0876E+02	8.1191E+02	8.8855E+02
	Min	8.3681E+02	8.4780E+02	8.0597E+02	8.1891E+02	8.0398E+02	8.0597E+02	8.8358E+02
	Var	1.7324E+02	6.9788E+02	1.2429E+01	1.2006E+01	4.2502E+00	8.6774E+00	1.0691E+01
H5	Mean	1.0177E+03	3.7064E+03	9.0084E+02	9.9874E+02	9.0014E+02	9.0125E+02	2.4749E+03
	Min	9.4109E+02	1.2088E+03	9.0000E+02	9.1117E+02	9.0000E+02	9.0000E+02	2.4354E+03
	Var	4.0276E+03	1.1004E+06	2.6190E+00	3.6581E+03	3.9862E−02	2.0023E+00	2.8177E+02
H6	Mean	1.1751E+04	3.4565E+08	1.8577E+03	1.8009E+03	1.8430E+03	1.8549E+03	1.8438E+03
	Min	1.8811E+03	6.7686E+03	1.8101E+03	1.8002E+03	1.8032E+03	1.8053E+03	1.8180E+03
	Var	7.0249E+07	2.1689E+16	6.4860E+02	2.1647E−01	9.2224E+02	1.2282E+03	3.5816E+02
H7	Mean	2.0387E+03	2.1319E+03	2.0184E+03	2.0165E+03	2.0168E+03	2.0178E+03	2.5018E+03
	Min	2.0232E+03	2.0330E+03	2.0046E+03	2.0023E+03	2.0013E+03	2.0033E+03	2.5003E+03
	Var	1.9249E+02	1.1951E+03	5.4968E+01	4.3465E+01	6.3480E+01	4.1448E+01	1.5503E+00
H8	Mean	2.2238E+03	2.2702E+03	2.2247E+03	2.2203E+03	2.2204E+03	2.2207E+03	2.2797E+03
	Min	2.2209E+03	2.2223E+03	2.2202E+03	2.2177E+03	2.2186E+03	2.2199E+03	2.2207E+03
	Var	1.9571E+01	3.9285E+02	4.3269E+02	3.8124E−01	2.7725E−01	2.3198E−01	6.2914E+03
H9	Mean	2.4808E+03	2.7229E+03	2.4808E+03	2.4656E+03	2.4808E+03	2.4808E+03	2.4653E+03
	Min	2.4808E+03	2.4865E+03	2.4808E+03	2.4654E+03	2.4808E+03	2.4808E+03	2.4653E+03
	Var	3.4466E−25	5.1984E+03	6.2039E−26	7.5102E−03	0.0000E+00	2.0680E−26	1.6544E−24
H10	Mean	2.5115E+03	2.6343E+03	2.5128E+03	2.5005E+03	2.5100E+03	2.4951E+03	5.8467E+03
	Min	2.5002E+03	2.5010E+03	2.4000E+03	2.5004E+03	2.4000E+03	2.4000E+03	5.1446E+03
	Var	3.6613E+03	8.9128E+03	2.0258E+03	2.6760E−03	2.5598E+03	2.2735E+03	8.2254E+04
H11	Mean	2.9167E+03	5.0970E+03	2.9267E+03	2.9300E+03	2.9100E+03	2.9100E+03	2.9567E+03
	Min	2.9000E+03	2.9014E+03	2.9000E+03	2.6000E+03	2.9000E+03	2.9000E+03	2.9000E+03
	Var	1.3889E+03	1.5295E+06	1.9556E+03	1.4100E+04	9.0000E+02	9.0000E+02	2.4556E+03
H12	Mean	2.9482E+03	3.2513E+03	2.9446E+03	2.8892E+03	2.9438E+03	2.9419E+03	4.6805E+03
	Min	2.9357E+03	2.9500E+03	2.9324E+03	2.8884E+03	2.9326E+03	2.9324E+03	2.9000E+03
	Var	3.2480E+01	1.4713E+04	5.0971E+01	7.7518E−02	8.9940E+01	4.4979E+01	9.2268E+05

$$g_1(\boldsymbol{x}) = 2x_1x_3 + x_3(x_2 - 2x_4) \le 300, \tag{13}$$

$$g_2(\boldsymbol{x}) = \frac{18x_2 \times 10^4}{x_3(x_2 - 2x_4)^3 + 2x_1x_3\left[4x_4^2 + 3x_2(x_2 - 2x_4)\right]} + \frac{15x_1 \times 10^3}{(x_2 - 2x_4)x_3^2 + 2x_3x_1^3} \le 56$$

Variable range:

$$10 \le x_1 \le 50,\ 10 \le x_2 \le 80,\ 0.9 \le x_3 \le 5,\ 0.9 \le x_4 \le 5.$$

Table 16 gives the best optimization results of the PPO and the competitors in the I-beam vertical deflection design problem. From the results, the PPO ranks first. The optimization results of all the competitors except GTO are similar to the optimization results of the PPO.

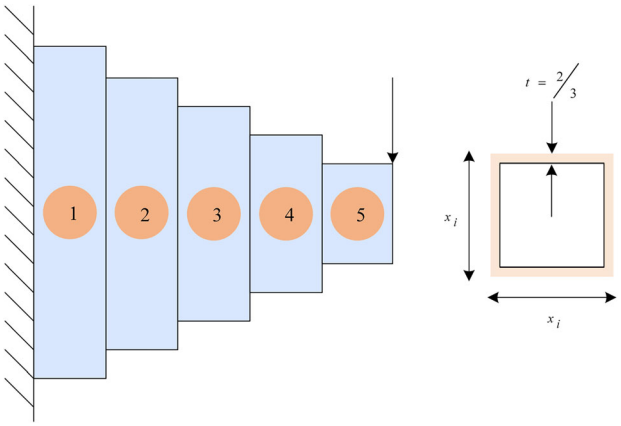


Fig. 13 Schematic diagram of the cantilever beam design

$$f(\boldsymbol{x}) = \frac{5.885x_4(x_1 + x_3)}{x_1 + \sqrt{|x_3^2 - x_2^2|}},$$

Subject to:

$$g_1(\boldsymbol{x}) = -x_4x_2\left(0.4x_1 + \frac{x_3}{6}\right) + 8.94\left(x_1 + \sqrt{|x_3^2 - x_2^2|}\right) \leq 0,$$

$$g_2(\boldsymbol{x}) = -x_4x_2^2\left(0.2x_1 + \frac{x_3}{12}\right) + 2.2\left(8.94\left(x_1 + \sqrt{|x_3^2 - x_2^2|}\right)\right)^{4/3}, \leq 0$$

$$g_3(\boldsymbol{x}) = -x_4 + 0.0156x_1 + 0.15 \leq 0, \quad (14)$$

Table 15 The best results of different algorithms for the cantilever beam design problem

	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
x_1	6.01453	6.01518	5.96503	5.99363	20.24540	6.01301	6.01337	4.96567
x_2	5.30831	5.31574	5.16259	5.30437	22.42386	5.30611	5.31075	8.96611
x_3	4.49541	4.48609	4.76749	4.51920	17.55510	4.51941	4.49409	4.85308
x_4	3.50239	3.50155	3.51202	3.51326	6.82134	3.48679	3.50098	4.23571
x_5	2.15302	2.15516	2.11720	2.14383	3.42599	2.14876	2.15447	2.42994
g_1	0.00000	0.00000	− 0.00139	0.00000	− 0.93893	0.00000	0.00000	− 0.12244
g_2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
f	1.339956	1.339961	1.343118	1.339996	4.397434	1.339983	1.339957	1.588112

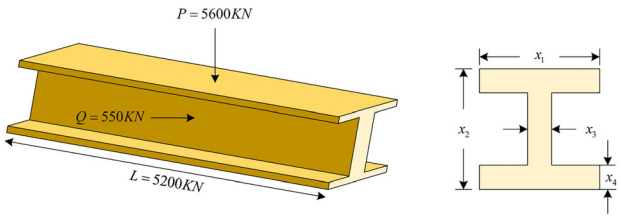


Fig. 14 Schematic diagram of the I-beam vertical deflection design

5.3 Corrugated bulkhead design problem

Corrugated bulkhead design problem [98] is a common engineering optimization problem as shown in Fig. 15. The optimization objective of this problem is to minimize the weight of the corrugated bulkhead of a chemical carrier. There are four decision variables in this problem, including the width (x_1), the depth (x_2), the length (x_3), and the thickness (x_4). The specific mathematical model of the Corrugated bulkhead design problem is shown in Eq. (14).

Consider:

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

Minimize:

$$g_4(\boldsymbol{x}) = -x_4 + 0.0156x_3 + 0.15 \leq 0,$$

$$g_5(\boldsymbol{x}) = -x_4 + 1.05 \leq 0,$$

$$g_6(\boldsymbol{x}) = -x_3 + x_2 \leq 0.$$

Variable range:

$$0 \leq x_1, x_2, x_3 \leq 100, \ 0 \leq x_4 \leq 5.$$

Table 17 gives the best optimization results of all the algorithms for the corrugated bulkhead design problem. From the results, the PPO slightly outperforms the MGO and AVOA and even significantly outperforms the algorithms such as the GTO and the TSA. To summarize, the PPO ranks first overall in the corrugated bulkhead design problem.

5.4 Gear train design problem

The gear train design problem [99] is a common unconstrained discrete engineering problem, as shown in Fig. 16.

In this problem, the gear ratio is defined as the ratio of the angular velocity of the output shaft to the angular velocity of the input shaft. The minimum gear ratio is taken as the final optimization objective. The four decision variables, designated n_A, n_B, n_C, and n_D, represent the

Table 16 The best results of different algorithms for the I-beam vertical deflection design problem

	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
x_1	80.00000	80.00000	80.00000	80.00000	79.62407	80.00000	80.00000	79.84561
x_2	50.00000	50.00000	50.00000	50.00000	36.51266	50.00000	50.00000	41.27569
x_3	0.90000	0.90000	0.90000	0.90000	1.08890	0.90000	0.90000	0.90324
x_4	2.32179	2.32179	2.32171	2.32179	2.87629	2.32179	2.32179	2.80433
g_1	0.00000	− 0.00057	− 0.00841	0.00000	− 9.51936	0.00000	0.00000	− 1.44502
g_2	− 1.57023	− 1.57022	− 1.57008	− 1.57023	− 0.59310	− 1.57023	− 1.57023	− 1.23197
g_3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
f	0.013074119	0.013074146	0.013074527	0.013074119	0.014449929	0.013074119	0.013074119	0.013352646

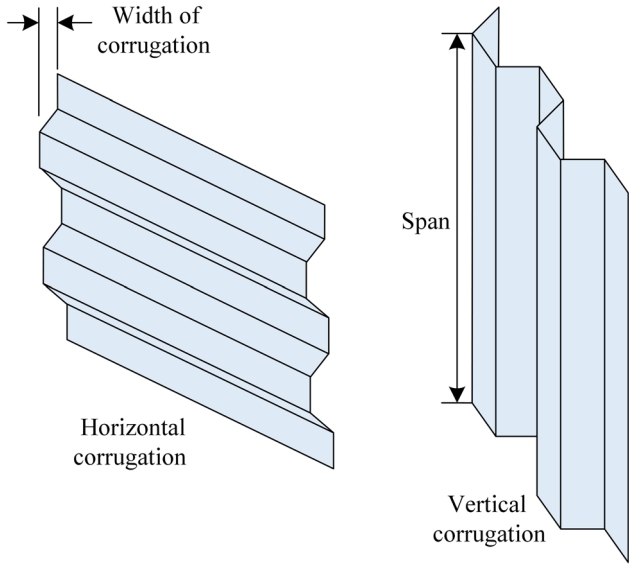


Fig. 15 Schematic diagram of the corrugated bulkhead design

Minimize:

$$f(\boldsymbol{x}) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5)$$

Subject to:

$$g(\boldsymbol{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0. \tag{15}$$

Variable range:

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100.$$

The best optimization results of all the algorithms for Gear train design problem are shown in Table 18. From the results, the PPO ranks first. Besides, the GJO, the LEA, the MGO, the AVOA and the NOA have the same optimization results as the PPO. However, the optimization results of the TSA and the GTO are poorer, two and three orders of magnitude worse than the best optimization results, respectively.

Table 17 The best results of different algorithms for the corrugated bulkhead design problem

	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
x_1	57.69231	57.61964	55.95022	57.70640	16.43537	57.69231	57.69231	52.11765
x_2	34.14762	34.14821	34.03432	34.14352	36.99876	34.14762	34.14762	33.78340
x_3	57.69231	57.62741	56.50741	57.71537	53.68656	57.69231	57.69231	53.74333
x_4	1.05000	1.05003	1.05493	1.05063	1.16882	1.05000	1.05000	1.05330
g_1	− 240.69462	− 240.69299	− 238.20565	− 241.19339	− 176.53133	− 240.69462	− 240.69462	− 220.95446
g_2	0.00000	− 16.20729	− 213.13723	− 1.38312	− 3809.77068	0.00000	− 0.00003	− 489.02785
g_3	0.00000	− 0.00116	− 0.03211	− 0.00041	− 0.76243	0.00000	0.00000	− 0.09026
g_4	0.00000	− 0.00104	− 0.02341	− 0.00027	− 0.18131	0.00000	0.00000	− 0.06490
g_5	0.00000	− 0.00003	− 0.00493	− 0.00063	− 0.11882	0.00000	0.00000	− 0.00330
g_6	− 23.54469	− 23.47920	− 22.47309	− 23.57184	− 16.68780	− 23.54469	− 23.54469	− 19.95993
g_7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
f	6.842958	6.845094	6.908534	6.846293	8.716315	6.842958	6.842958	6.987122

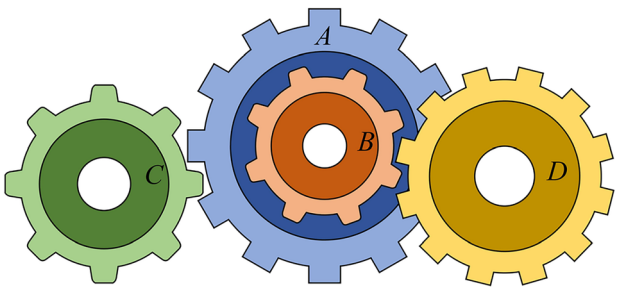


Fig. 16 Schematic diagram of the gear train design

number of teeth in each gear. The specific model of this problem is given by the following Eq. (15):

Consider:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5].$$

6 Application to 3D path planning

The path planning of the UAV power inspection represents a crucial aspect of the power inspection problem, exerting a direct influence on the probability of transmission line faults. The scientific and rational path of the UAV power inspection is closely associated with the efficiency of the UAV and the safety of the power grid. The preceding section evaluated the PPO's capabilities to address benchmark and engineering problems, substantiating its exemplary optimization functionality. In light of the aforementioned considerations, this section employs the PPO to address the 3D path planning of a UAV in a power inspection context. The population size of the PPO is set to 30, with a maximum number of function evaluations capped at 10,000. To obtain optimal results, the process is

Table 18 The best results of different algorithms for the gear train design problem

	PPO	GJO	TSA	LEA	GTO	MGO	AVOA	NOA
x_1	43.34613	42.76682	47.45929	43.29845	57.98712	43.17649	42.66003	49.41736
x_2	16.01725	15.81546	13.35721	19.30006	12.01797	15.95877	15.77737	19.47316
x_3	18.62745	19.11401	12.00000	16.05012	29.77879	18.96210	19.02989	15.80228
x_4	48.83598	49.46859	22.84399	49.41381	43.42751	48.61007	49.21926	43.39564
g_1	0	0	0	0	0	0	0	0
g_2	0	0	0	0	0	0	0	0
f	2.70086E−12	2.70086E−12	9.92158E−10	2.70086E−12	4.50330E−09	2.70086E−12	2.70086E−12	2.70086E−12

executed 25 times independently, and the resulting optimization outcomes are subjected to rigorous analysis and discussion.

6.1 Environment building and path representation

The UAVs often face high natural mountains when performing tasks such as the inspection of ultra-high voltage transmission lines. This flight environment can be described by an exponential function, and the mathematical model [100] can be expressed as follows

$$z(x,y)=\sum_{i=1}^{n} h_i \exp\left[-\left(\frac{x-x_c^{(i)}}{x_d^{(i)}}\right)^2-\left(\frac{y-y_c^{(i)}}{y_d^{(i)}}\right)^2\right] \tag{16}$$

where n denotes the total number of peaks and $\left(x_c^{(i)}, y_c^{(i)}\right)$ is the center coordinate of the ith peak. h_i is the topographic parameter of the ith peak, which is used to control the height of the peak. $x_d^{(i)}$ and $y_d^{(i)}$ are the decay amounts of the ith peak along the X-axis and Y-axis directions, respectively, for controlling the slope of the peak.

The path of a UAV is usually represented by ordered and discrete points, and the coordinate points close to each other are correlated by cubic spline interpolation. Suppose this ordered set of nodes is $\{\boldsymbol{X}_s, \boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n, \boldsymbol{X}_e\}$. For the starting point, $\boldsymbol{X}_s$ can be denoted as $\boldsymbol{X}_s=(x_s, y_s, z_s)$. As the ending point,$\boldsymbol{X}_e$ can be shown as $\boldsymbol{X}_e=(x_e, y_e, z_e)$. $\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n$ shows the resulting 1 to n nodes, and can be represented as $\boldsymbol{X}_i=(x_i, y_i, z_i)$, $i=1,2,\cdots n$. x_i and y_i are integers.

6.2 Mathematical model for path planning

In this problem, the optimization variables are the amount of variation of each node relative to the previous node in the X-axis and Y-axis direction, as well as the amount of variation relative to the terrain height. The coordinates of the ith node are then expressed as

$$\begin{cases} x_i=\text{round}\left(x_s+\sum_{j=1}^{i}\Delta x_j\right) \\ y_i=\text{round}\left(y_s+\sum_{j=1}^{i}\Delta y_j\right) \\ z_i=H_i+\Delta z_i \end{cases} \tag{17}$$

where Δx_j and Δy_j are the changes in the X-axis and Y-axis directions of the j node relative to the previous node, respectively, H_i refers to the height of terrain corresponding to the location of the ith node, and round$(\cdot)$ is a rounding function.

After the nodes are updated, path smoothing is performed using three times B-sample interpolation. If the number of interpolation is m, the path point after the path smoothing is $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots \boldsymbol{Y}_m\}$, where $\boldsymbol{Y}_1$ is the starting point coordinate, $\boldsymbol{Y}_m$ is the end point coordinate, and $\boldsymbol{Y}_i=(\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i)$, $i=1,2,\cdots m$.

The path planning of UAV is subject to a greater number of constraints, which render the identification of an optimal path particularly challenging. In addressing these constraints, researchers have typically employed the penalty function method. In the event that a candidate solution fails to satisfy the constraints, the objective value corresponding to that solution is subjected to a penalty. In the case of a minimization problem, a larger value is added to the penalty. Conversely, in the case of a maximization problem, a penalty is applied by subtracting a larger value.

The path planning of UAV typically involves the consideration of multiple objectives. However, the majority of current multi-objective problems are transformed into single-objective problems through the process of weighting. The objective function defined in this paper is

$$F=F_1+F_2+F_3+F_4+F_5+F_6 \tag{18}$$

where F_1 is a path length function, which is used to calculate the path length. F_2 shows a height fluctuation function, which is used to represent the degree of fluctuation of the path height. F_3 as a corner constraint function to prevent the corner of the path from exceeding the maximum corner. F_4 is a terrain constraint function to prevent the UAV from colliding with mountains and the ground. F_5 refers to a boundary constraint function, which is used to prevent the horizontal and vertical coordinates of the generated path points from exceeding the solution space. F_6 serves as an altitude constraint function to ensure that the path does not exceed the maximum flight altitude of the UAV.

F_1 is used to calculate the total length of the path. The specific calculation formula is

$$F_1 = \sum_{i=1}^{m-1} \sqrt{(\widetilde{x}_i - \widetilde{x}_{i+1})^2 + (\widetilde{y}_i - \widetilde{y}_{i+1})^2 + (\widetilde{z}_i - \widetilde{z}_{i+1})^2} \quad (19)$$

where $(\widetilde{x}_i, \widetilde{y}_i, \widetilde{z}_i)$ is the coordinate of the ith path point. F_2 describes the degree of fluctuation of the path height, so the standard deviation is found for the height of the path sequence. F_2 is expressed as

$$F_2 = \sqrt{\frac{1}{m}\sum_{i=1}^{m} (\widetilde{z}_i - z_{mean})^2} \quad (20)$$

where z_{mean} refers the mean value of height, which is calculate as

$$z_{\text{mean}} = \frac{1}{m}\sum_{i=1}^{m} \widetilde{z}_i \quad (21)$$

F_3 is used to penalize paths where the corner exceeds the maximum corner. Set the maximum angle of rotation of UAV to ψ. F_3 can be expressed as

$$F_3 = \begin{cases} 10^4, \exists \theta_i > \psi \\ 0, \forall \theta_i \le \psi \end{cases} \quad (22)$$

where θ_i denotes the ith corner. It is calculated as

$$\theta_i = \arctan\left(\frac{\boldsymbol{A}_i \cdot \boldsymbol{B}_i}{|\boldsymbol{A}_i||\boldsymbol{B}_i|}\right) \quad (23)$$

where $i = 1, 2, \cdots, m-2$. The calculation of $\boldsymbol{A}_i$ is shown in Eq. (24). The calculated $\boldsymbol{B}_i$ is described in Eq. (25).

$$\boldsymbol{A}_i = Y_j - Y_{j-1} \quad (24)$$

$$\boldsymbol{B}_i = Y_{j+1} - Y_j \quad (25)$$

where j satisfies $j = i + 1$. F_4 is employed to penalize paths that intersect with mountains and is denoted as

$$F_4 = \begin{cases} 10^4, \exists \widetilde{z}_i \le \widehat{z}_i + \Delta h_{min} \\ 0, \forall \widetilde{z}_i > \widehat{z}_i + \Delta h_{min} \end{cases} \quad (26)$$

where $\widehat{z}_i$ refers the height of the mountain at $(\widetilde{x}_i, \widetilde{y}_i)$ obtained by the bilinear interpolation algorithm, and Δh_{min} expresses the minimum height of the UAV from the mountain. Even though $\{\boldsymbol{X}_s, \boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n, \boldsymbol{X}_e\}$ doesn't exceed the map boundary, $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots \boldsymbol{Y}_m\}$ gained by three B spline interpolations may expand beyond the map boundary. Function F_5 is hence required for boundary crossing. F_5 is being stated as

$$F_5 = \begin{cases} 0, \forall \widetilde{x}_i, \widetilde{y}_i \in [m_l, m_u] \\ 10^4, \exists \widetilde{x}_i, \widetilde{y}_i \notin [m_l, m_u] \end{cases} \quad (27)$$

where m_l denotes the minimum value of the map boundary, and m_u is defined as the maximum value of the map boundary. F_6 refers an altitude constraint function, i.e. the flight altitude of the UAV cannot be higher than the set maximum altitude h_{max}. It can be calculated as

$$F_6 = \begin{cases} 10^4, \exists \widetilde{z}_i > h_{max} \\ 0, \forall \widetilde{z}_i \le h_{max} \end{cases} \quad (28)$$

6.3 Simulation experiments

Set the map as a 100×100 raster, then $m_l = 1$ and $m_u = 100$. The UAV flight starting point is $(10, 10, 2)$ and ending point is $(90, 90, 2)$. Make $\Delta h_{min} = 1$, $h_{max} = 12$, $\psi = \pi/2$, $n = 7$. For the ith node, Δx_i, $\Delta y_i \in [-30, 30]$, $\Delta z_i \in [\Delta h_{min}, 5]$.

In order to more comprehensively assess the speed and stability of the convergence of the PPO to the global optimum, two additional metrics are employed in this experiment: success rate (Sr) and mean evaluation (Me) [101, 102]. An algorithm is deemed successful if the objective function value obtained in an independent run is superior to a predefined threshold. The Sr is then calculated as the ratio of successful runs to the total number of independent runs. For successful runs, the minimum number of evaluations required for the objective function value to exceed the threshold is recorded. The average of the number of assessments yields the value of the Me indicator.

Two flight situations are considered: situation without no-fly zones and situation with no-fly zones. Set five cylindrical no-fly zones which are illustrated in Table 19, including the center coordinates (x_c, y_c), height (h_c), and radius (r_c) on the XY plane in the environmental model

Table 19 Information about the cylindrical no-fly zone

No.	1	2	3	4	5
x_c	56.7157	32.6590	13.1987	65.9033	64.0290
y_c	18.5965	39.5000	45.5621	70.4151	44.0858
h_c	6.4409	8.2620	3.8032	8.5927	9.5035
r_c	5.0676	4.7530	10.5505	4.9743	8.7564

with no-fly zone. Figure 17 illustrates the terrain environment for both cases.

In order to verify the performance of the PPO on this problem, the same algorithm as the benchmark test is selected for comparison with the PPO. The best results obtained by each algorithm for the no-fly zone condition are shown in Fig. 18. From Fig. 18a and b, it can be seen that the path obtained by the PPO, with less altitude fluctuations, is almost in the same plane except for the takeoff and landing parts. It can be seen that the overall the PPO-derived path has a lower height and stabilized path, which is more favorable for the UAV flight. In contrast, the NOA, the TSA, the GJO and the AVOA solicit paths with greater fluctuations in height. Meanwhile, the GTO, the LEA and the MGO performed similarly to the PPO and also obtained flight paths with less altitude fluctuations. From Fig. 18c, it can be seen that the paths obtained by the TSA, the NOA, and AVOA have large fluctuations in level, which are not conducive to smooth the UAV flight. The paths obtained by other algorithms, on the other hand, fluctuate less horizontally and are approximately a straight line. The values of each function obtained by each algorithm are shown in Table 20, and the best nodes are shown in Table 21. It can be concluded that all algorithms obtain feasible paths from Fig. 18d, it can be known that the NOA gets a feasible solution only after several iterations. It is noteworthy that the PPO finds the smallest values of both F_1 and F_2 functions.

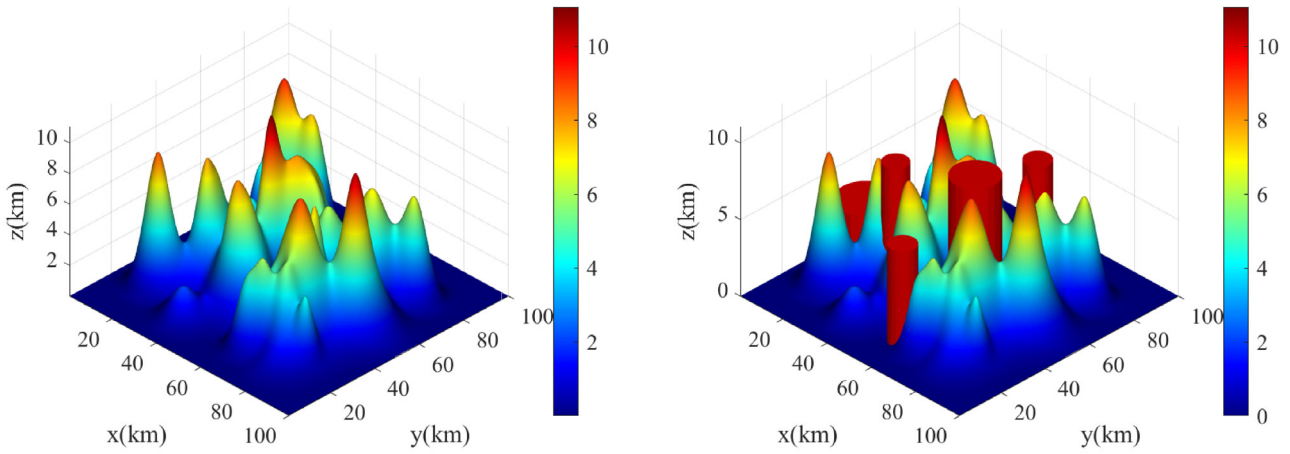
Table 22 presents the values of the *Sr* and *Me* metrics for the PPO and its competitors at a threshold assignment of 120. The results demonstrate that the PPO falls below the threshold in 30 runs. Additionally, the *Me* value indicates that the PPO can achieve satisfactory results with less number of function evaluations. It can be verified that the PPO exhibits a quality convergence performance in comparison to its competitors.

The presence of no-fly zones makes route planning more difficult. For F_4, the calculation is given by Eq. (29) as

$$F_4 = \begin{cases} 10^4, \exists \widetilde{z}_i \leq \max\left\{\widehat{z}_i, \widehat{h}_i\right\} + \Delta h_{min} \\ 0, \forall \widetilde{z}_i > \max\left\{\widehat{z}_i, \widehat{h}_i\right\} + \Delta h_{min} \end{cases} \tag{29}$$

where $\widehat{h}_i$ denotes the height of the no-fly zones corresponding to the ith path point. $\widehat{h}_i$ is equal to 0 when the ith path point not over the no-fly zone.

The best paths obtained by each algorithm under this condition are displayed in Fig. 19. The values of each function obtained by each algorithm are presented in Table 23 and the best nodes are displayed in Table 24. From Fig. 19a and b, it can be seen that the path obtained by the PPO is smoother in the middle part of the path and has a lower height compared to the paths obtained by other algorithms, while it is higher in the first half of the path compared to the other paths. In the first half, the UAV needs to reach a certain height to take off. The gradual increase of the path in the first half is probably to make the UAV height fluctuation as small as possible. From Table 24, it can be concluded that although the path obtained by PPO is higher in the first half compared to



(a) Situation without no-fly zones

(b) Situation with no-fly zones

Fig. 17 Terrain environment

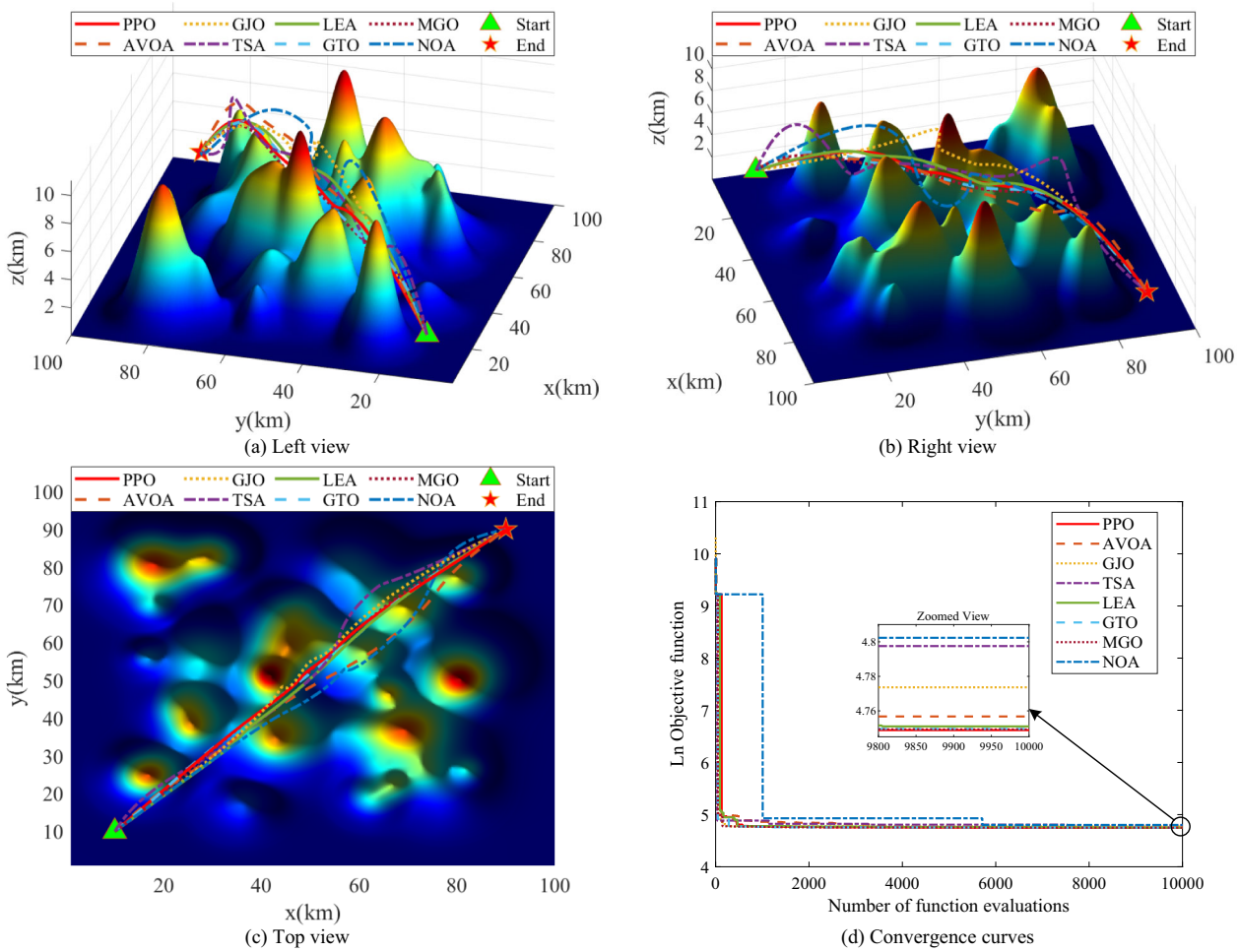


Fig. 18 The best paths obtained by each algorithm without the no-fly zones condition

Table 20 Optimization results of each algorithm without no-fly zones condition

F	PPO	AVOA	GJO	TSA	LEA	GTO	MGO	NOA
F_1	116.23105	116.35734	122.09085	133.44874	117.18561	116.31413	116.47134	121.39515
F_2	114.69335	114.80424	120.23514	131.28642	115.56016	114.68581	114.91147	119.28779
F_3	1.53770	1.55310	1.85572	2.16232	1.62545	1.62832	1.55988	2.10736
F_4	0	0	0	0	0	0	0	0
F_5	0	0	0	0	0	0	0	0
F_6	0	0	0	0	0	0	0	0
F_7	0	0	0	0	0	0	0	0

other paths, it does not affect the path length too much, and on the contrary, it may make the height fluctuation smaller. The TSA-derived path has less height fluctuations, but as seen in Fig. 19, the path is poor and cannot be used as a path for UAV inspection. It should be noted that the paths derived by the NOA and the GJO have large altitude fluctuations, which are not conducive to the UAV flight. As can be seen from Fig. 19c, the TSA, the GJO and the NOA-derived paths are more volatile in level, while all other paths are smoother. As can be seen from Fig. 19d, the NOA and the GTO require more iteration to find feasible paths. while the other algorithms are able to find feasible paths in smaller iterations. Taken together, although the convergence curve of the PPO is not the fastest decreasing but it has the highest convergence accuracy. The data in Table 24 show that the PPO finds the smallest values of both F_1 and F_2 functions, which again verifies the excellent performance of the PPO.

Table 25 illustrates the *Sr* and *Me* of the PPO and its competitors at a threshold of 125. The results demonstrate

Table 21 The best nodes obtained by each algorithm without no-fly zone condition

Algorithm		Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9
PPO	x	10.0000	32.0000	40.0000	46.0000	49.0000	53.0000	60.0000	71.0000	90.0000
	y	10.0000	34.0000	42.0000	48.0000	52.0000	56.0000	64.0000	74.0000	90.0000
	z	2.0000	6.5407	6.2356	6.5759	6.4890	6.7043	7.0938	7.1795	2.0000
AVOA	x	10.0000	21.0000	27.0000	39.0000	51.0000	63.0000	73.0000	78.0000	90.0000
	y	10.0000	22.0000	27.0000	39.0000	49.0000	58.0000	69.0000	77.0000	90.0000
	z	2.0000	5.0391	5.4414	6.6448	6.5161	7.1079	7.0862	7.5141	2.0000
GJO	x	10.0000	36.0000	42.0000	44.0000	46.0000	50.0000	53.0000	61.0000	90.0000
	y	10.0000	37.0000	46.0000	46.0000	48.0000	55.0000	58.0000	68.0000	90.0000
	z	2.0000	7.0671	9.1931	8.9533	9.0024	8.8594	8.0991	8.2310	2.0000
TSA	x	10.0000	19.0000	28.0000	36.0000	52.0000	57.0000	63.0000	68.0000	90.0000
	y	10.0000	22.0000	29.0000	36.0000	53.0000	65.0000	74.0000	77.0000	90.0000
	z	2.0000	6.7572	4.2551	6.1536	7.5521	7.7551	9.8439	7.0140	2.0000
LEA	x	10.0000	28.0000	34.0000	38.0000	45.0000	53.0000	57.0000	67.0000	90.0000
	y	10.0000	27.0000	33.0000	38.0000	45.0000	54.0000	59.0000	70.0000	90.0000
	z	2.0000	6.2509	6.7841	7.1038	7.5728	7.0914	7.1169	7.6179	2.0000
GTO	x	10.0000	20.0000	28.0000	37.0000	46.0000	55.0000	64.0000	73.0000	90.0000
	y	10.0000	21.0000	30.0000	39.0000	48.0000	57.0000	67.0000	76.0000	90.0000
	z	2.0000	4.9906	5.5404	5.8760	6.1901	6.4884	7.4285	6.7649	2.0000
MGO	x	10.0000	19.0000	27.0000	35.0000	45.0000	53.0000	61.0000	70.0000	90.0000
	y	10.0000	19.0000	28.0000	37.0000	47.0000	56.0000	65.0000	74.0000	90.0000
	z	2.0000	4.6012	5.2710	5.6299	6.2031	6.1505	6.8844	6.9499	2.0000
NOA	x	10.0000	39.0000	47.0000	54.0000	57.0000	60.0000	71.0000	77.0000	90.0000
	y	10.0000	37.0000	43.0000	49.0000	53.0000	54.0000	67.0000	81.0000	90.0000
	z	2.0000	9.7054	5.9796	5.1078	7.4799	8.7556	8.2393	5.0211	2.0000

Table 22 The *Sr* and *Me* metrics of the and its competitors without no-fly zone condition

Metrics	PPO	AVOA	GJO	TSA	LEA	GTO	MGO	NOA
Sr	100.00%	86.67%	6.67%	0.00%	90.00%	100.00%	100.00%	0.00%
Me	5789.283	5551.32	5135.315	*N.A.*	6533.954	5443.463	5119.396	*N.A.*

N.A. means that the value is not obtained

that the PPO exhibits accelerated convergence while maintaining a high level of convergence accuracy. It is also verified that PPO has good convergence performance considering the best optimization results of the PPO in path planning.

Collectively, all algorithms find a feasible path under the above two conditions. The TSA finds the worst path and the proposed PPO finds the best path. Moreover, the path found by PPO has the advantages of smoothness, low height and short path, which can provide a better path for the UAV power inspection. The source codes of the PPO for solving the 3D path planning problem are publicly available at https://ww2.mathworks.cn/matlabcentral/fileexchange/171634-philoponella-prominens-optimizer-for-3d-path-planning.

7 Conclusions

This paper proposes a novel bio-inspired metaheuristic algorithm called the *P. prominens* optimizer, inspired by the distinctive mating pattern of *P. prominens*. The algorithmic approach primarily emulates the distinctive post-mating behavior observed in the males of the *P. prominens* species. Following the completion of mating, if the male is unable to escape by ejecting, it is consumed by the female, who then gives birth to new spiders. Males that have successfully escaped regain their strength by feeding.

In order to examine the performance of the PPO to solve the benchmark problems, we use 41 benchmark functions from CEC2017 and CEC2022 to verify the various performances of the PPO. On F1, F3 $\sim$ F10, by comparing

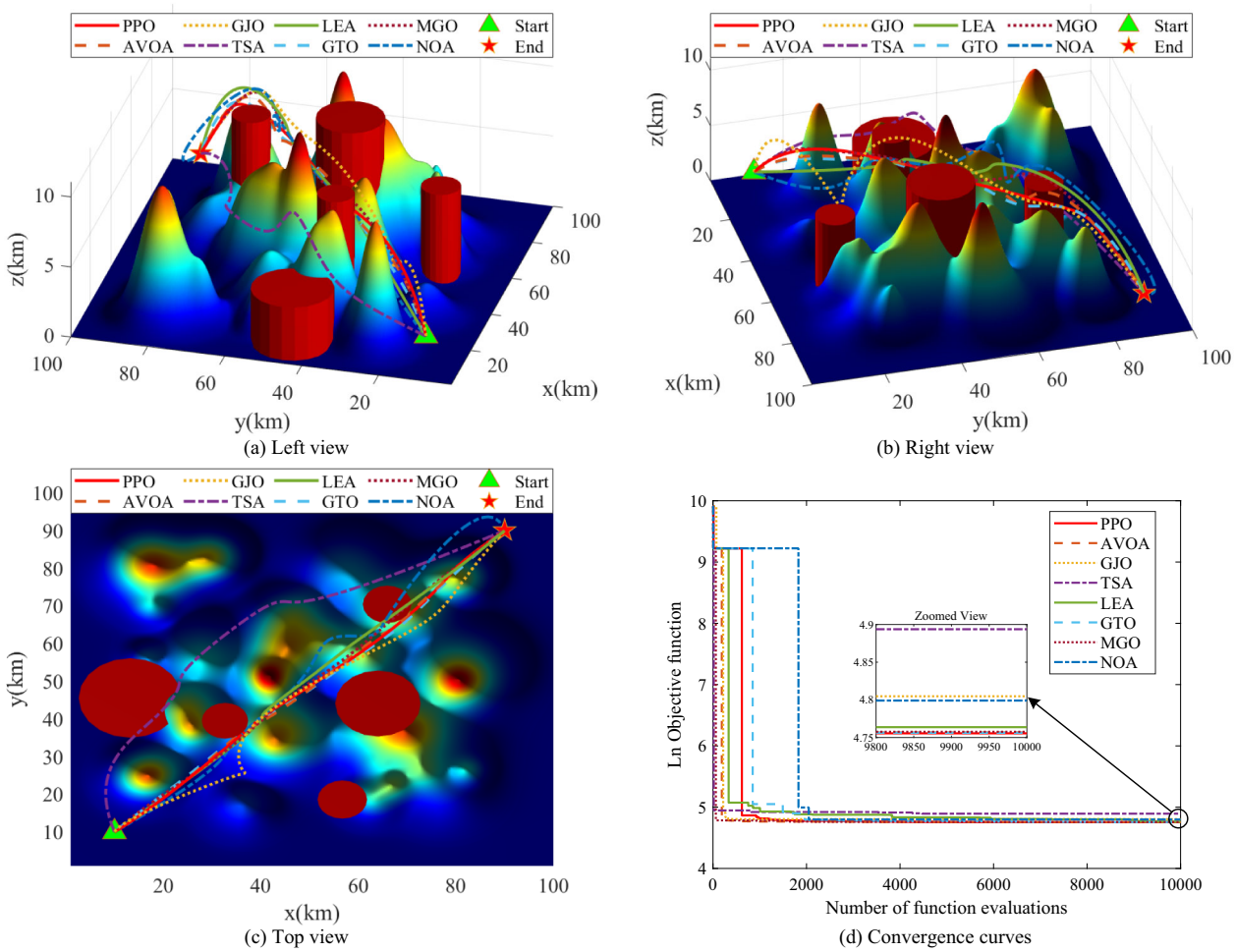


Fig. 19 The best paths obtained by each algorithm with the no-fly zones condition

Table 23 Optimization results of each algorithm with no-fly zones condition

F	PPO	AVOA	GJO	TSA	LEA	GTO	MGO	NOA
F_1	115.45144	116.36839	118.46680	121.21106	115.70127	115.53291	115.50224	121.79353
F_2	114.33872	115.04496	116.64099	119.28644	114.41365	114.24930	114.26453	119.73252
F_3	1.11272	1.32342	1.82581	1.92462	1.28762	1.28361	1.23772	2.06101
F_4	0	0	0	0	0	0	0	0
F_5	0	0	0	0	0	0	0	0
F_6	0	0	0	0	0	0	0	0
F_7	0	0	0	0	0	0	0	0

the results of the PPO with recently proposed algorithms (the AVOA, the GJO, the TSA, the LEA, the GTO, the MGO, and the NOA), we verified that the PPO has superior exploration and exploitation capabilities. From the results on F11 $\sim$ F30, PPO has a good capability of avoiding locally optimal, which benefits from PPO's capability of balancing its exploration and exploitation operations well. After that, we illustrate the shift from exploration to exploitation of the PPO by showing 2D images, search history, trajectories, and average fitness curves on some of the functions of the PPO. Similarly, we show the convergence curves of the PPO and competitive algorithms on partial functions, verifying that the proposed algorithm has good convergence performance. Moreover, the Wilcoxon signed rank sum test confirmed 93.60% significance compared to the competitive calculation; in the Friedman ranking test, the PPO ranks first; and the Kruskal–Wallis test verified that the PPO has the smallest value on about 2/3 of the functions. After that, we performed scalability analysis and sensitivity analysis on the PPO, verifying that

Table 24 The best nodes obtained by each algorithm with no-fly zone condition

Algorithm		Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9
PPO	x	10.0000	24.0000	27.0000	35.0000	60.0000	68.0000	77.0000	81.0000	90.0000
	y	10.0000	23.0000	26.0000	34.0000	57.0000	65.0000	76.0000	81.0000	90.0000
	z	2.0000	5.6358	5.9731	6.9455	7.5267	8.4263	7.5378	6.4306	2.0000
AVOA	x	10.0000	19.0000	28.0000	36.0000	46.0000	56.0000	65.0000	74.0000	90.0000
	y	10.0000	19.0000	28.0000	36.0000	45.0000	54.0000	62.0000	72.0000	90.0000
	z	2.0000	4.9562	6.0598	6.6190	6.9326	7.3723	7.5120	8.6103	2.0000
GJO	x	10.0000	17.0000	20.0000	33.0000	55.0000	60.0000	75.0000	77.0000	90.0000
	y	10.0000	15.0000	21.0000	36.0000	58.0000	73.0000	82.0000	82.0000	90.0000
	z	2.0000	5.3266	6.4031	10.0174	6.9050	5.7812	5.6372	5.0246	2.0000
TSA	x	10.0000	12.0000	28.0000	34.0000	37.0000	46.0000	50.0000	66.0000	90.0000
	y	10.0000	28.0000	44.0000	45.0000	54.0000	65.0000	65.0000	66.0000	90.0000
	z	2.0000	6.0903	8.9712	7.5865	7.0869	8.6314	9.0655	11.3081	2.0000
LEA	x	10.0000	18.0000	19.0000	24.0000	26.0000	30.0000	34.0000	60.0000	90.0000
	y	10.0000	16.0000	17.0000	21.0000	24.0000	28.0000	32.0000	56.0000	90.0000
	z	2.0000	5.5148	5.2993	5.1154	5.0980	5.9938	6.7824	9.6547	2.0000
GTO	x	10.0000	20.0000	29.0000	38.0000	48.0000	59.0000	71.0000	79.0000	90.0000
	y	10.0000	20.0000	29.0000	38.0000	48.0000	58.0000	68.0000	78.0000	90.0000
	z	2.0000	4.3507	5.5686	6.5332	6.7409	7.6083	7.8371	7.4731	2.0000
MGO	x	10.0000	20.0000	28.0000	37.0000	47.0000	57.0000	66.0000	74.0000	90.0000
	y	10.0000	19.0000	28.0000	37.0000	46.0000	55.0000	63.0000	72.0000	90.0000
	z	2.0000	5.1529	5.9874	6.5439	7.0357	7.2619	7.7026	8.7907	2.0000
NOA	x	10.0000	25.0000	29.0000	43.0000	48.0000	70.0000	75.0000	83.0000	90.0000
	y	10.0000	20.0000	26.0000	43.0000	50.0000	60.0000	62.0000	72.0000	90.0000
	z	2.0000	3.9590	5.5128	8.6675	8.5146	6.8175	5.5274	3.9305	2.0000

Table 25 The *Sr* and *Me* metrics of the and its competitors with no-fly zone condition

Metrics	PPO	AVOA	GJO	TSA	LEA	GTO	MGO	NOA
Sr	100.00%	93.33%	10.00%	0.00%	100.00%	100.00%	100.00%	6.67%
Me	5870.269	5356.232	5158.157	*N.A.*	6707.34	5558.129	5128.709	6488.161

N.A. means that the value is not obtained

its optimization results generally do not differ much when the dimensions and population sizes change, indicating the scalability and stability of the proposed algorithm. It is worth mentioning that the proposed algorithm ranks third in terms of computational cost, which is a sizable ranking given that the performance is better than the comparison algorithms. Finally, we compare the results of the PPO with six SOTAs (the JADE, the LSHADE, the A-LSHADE, the LSHADE-spacma, the LSHADE-cnEpSin, and the CMA-ES) on CEC2022, verifying that the PPO is still competitive even in the face of SOTAs algorithms.

For engineering applications, four complex engineering problems are selected , including the cantilever beam design, I-beam vertical deflection design, corrugated bulkhead design, and gear train design. The current publicly available optimization results are compared with the PPO to verify the feasibility of the PPO to solve practical engineering problems. Finally, the application of the PPO to 3D path planning of the UAV in power inspection further reflects the practical application value of the PPO.

From the good results that the PPO has gotten in this article, the future research of the PPO is very promising, and we provide some ideas for the future progress of the PPO:

(1) Attempt to develop multi-objective and binary versions of the PPO.
(2) Try to apply the PPO to other challenging engineering problems.

(3) Improving the PPO using different methods such changing the population initialization method, changing the updating mechanism and changing the guidance mechanism.
(4) Prove theoretically the convergence of the PPO.
(5) Attempt to fuse the PPO with other metaheuristics to propose fusion algorithms with better performance.

Acknowledgements The authors have no acknowledgements to declare.

Author contributions Yuansheng Gao: conceptualization, methodology, software, investigation, verification, writing-original draft, writing-reviewing & editing; Jinpeng Wang: software, validation, data curation, writing-original draft, writing-reviewing & editing; Changlin Li: software.

Funding The authors have not disclosed any funding.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Kyriakakis, N.A., Marinaki, M., Matsatsinis, N., Marinakis, Y.: A cumulative unmanned aerial vehicle routing problem approach for humanitarian coverage path planning. Eur. J. Oper. Res. **300**(3), 992–1004 (2022)
2. Peng, C., Qiu, S.: A decomposition-based constrained multi-objective evolutionary algorithm with a local infeasibility utilization mechanism for UAV path planning. Appl. Soft Comput. **118**, 108495 (2022)
3. Rostami, S.M.H., Sangaiah, A.K., Wang, J., Liu, X.: Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–19 (2019)
4. Wang, X., Zhang, H., Liu, S., Wang, J., Wang, Y., Shangguan, D.: Path planning of scenic spots based on improved A* algorithm. Sci. Rep. **12**(1), 1320 (2022)
5. Zhou, Y., Zhang, E., Guo, H., Fang, Y., Li, H.: Lifting path planning of mobile cranes based on an improved RRT algorithm. Adv. Eng. Inform. **50**, 101376 (2021)
6. Lv, J.X., Yan, L.J., Chu, S.C., Cai, Z.M., Pan, J.S., He, X.K., Xue, J.K.: A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3D path planning of multiple UAVs in power inspection. Neural Comput. Appl. **34**(14), 11911–11936 (2022)
7. Pan, J.S., Lv, J.X., Yan, L.J., Weng, S.W., Chu, S.C., Xue, J.K.: Golden eagle optimizer with double learning strategies for 3D path planning of UAV in power inspection. Math. Comput. Simul **193**, 509–532 (2022)
8. Yang, Q., Yang, Z., Hu, G., Du, W.: A new fusion chemical reaction optimization algorithm based on random molecules for multi-rotor uav path planning in transmission line inspection. J. Shanghai Jiaotong Univ. (Science) **23**, 671–677 (2018)
9. Wei, Z., Xuhu, Z., Shigang, L., et al.: An improved Harris hawks optimizer combining novel nonlinear convergence factor and mutation strategy for global optimization, 25 October 2022, PREPRINT (Version 1) available at Research Square. https://doi.org/10.21203/rs.3.rs-1830937/v1
10. Guo, Y., Liu, X., Liu, X., Yang, Y., Zhang, W.: FC-RRT*: an improved path planning algorithm for UAV in 3D complex environment. ISPRS Int. J. Geo Inf. **11**(2), 112 (2022)
11. Bai, Z., Zhou, H., Shi, J., Xing, L., Wang, J.: A hybrid multi-objective evolutionary algorithm with high solving efficiency for UAV defense programming. Swarm Evol. Comput. **87**, 101572 (2024)
12. Gao, Y., Li, C., Huang, L.: An improved deep extreme learning machine to predict the remaining useful life of lithium-ion battery. Front. Energy Res. **10**, 1032660 (2022)
13. Yang, X.S.: Engineering optimization: an introduction with metaheuristic applications. Wiley, Hoboken (2010)
14. Fogel, D.B.: Artificial intelligence through simulated evolution. Wiley, Hoboken (1998)
15. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
16. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**(2), 82–102 (1999)
17. Ingo, R.: Evolution strategy: optimization of technical systems by means of biological evolution. Fromman-Holzboog. Stuttgart **104**, 15 (1973)
18. Mitchell, M.: An introduction to genetic algorithms. MIT Press, Cambridge (1998)
19. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Stat. Comput. **4**, 87–112 (1994)
20. Civicioglu, P., Besdok, E.: Contrast stretching based pansharpening by using weighted differential evolution algorithm. Expert Syst. Appl. **208**, 118144 (2022)
21. Amali, D., Dinakaran, M.: Wildebeest herd optimization: a new global optimization algorithm inspired by wildebeest herding behaviour. J. Intell. Fuzzy Syst. **37**(6), 8063–8076 (2019)
22. Veysari, E.F.: A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm. Expert Syst. Appl. **193**, 116468 (2022)
23. Gao, Y., Zhang, J., Wang, Y., Wang, J., Qin, L.: Love evolution algorithm: a stimulus–value–role theory-inspired evolutionary algorithm for global optimization. J. Supercomput. (2024). https://doi.org/10.1007/s11227-024-05905-4
24. Gelatt, C.D.: Optimization by simulated annealing. Science **200**, 671 (1983)
25. Birbil, Şİ, Fang, S.C.: An electromagnetism-like mechanism for global optimization. J. Global Optim. **25**, 263–282 (2003)
26. Yadav, A.: AEFA: artificial electric field algorithm for global optimization. Swarm Evol. Comput. **48**, 93–108 (2019)
27. Faramarzi, A., Heidarinejad, M., Stephens, B., Mirjalili, S.: Equilibrium optimizer: a novel optimization algorithm. Knowl.-Based Syst. **191**, 105190 (2020)
28. Mirrashid, M., Naderpour, H.: Transit search: an optimization algorithm based on exoplanet exploration. Results Control Optim. **7**, 100127 (2022)
29. Cheng, M.Y., Sholeh, M.N.: Optical microscope algorithm: a new metaheuristic inspired by microscope magnification for solving engineering optimization problems. Knowl.-Based Syst. **279**, 110939 (2023)
30. Azizi, M., Aickelin, U., Khorshidi, H.A., Shishehgarkhaneh, M.B.: Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. Sci. Rep. **13**(1), 226 (2023)
31. Kundu, R., Chattopadhyay, S., Nag, S., Navarro, M.A., Oliva, D.: Prism refraction search: a novel physics-based metaheuristic algorithm. J. Supercomput. (2024). https://doi.org/10.1007/s11227-023-05790-3

32. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. **96**, 120–133 (2016)
33. Su, H., Zhao, D., Heidari, A.A., Liu, L., Zhang, X., Mafarja, M., Chen, H.: RIME: a physics-based optimization. Neurocomputing **532**, 183–214 (2023)
34. Ahmadianfar, I., Bozorg-Haddad, O., Chu, X.: Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf. Sci. **540**, 131–159 (2020)
35. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., Gandomi, A.H.: The arithmetic optimization algorithm. Comput. Methods Appl. Mech. Eng. **376**, 113609 (2021)
36. Ahmadianfar, I., Heidari, A.A., Noshadian, S., Chen, H., Gandomi, A.H.: INFO: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst. Appl. **195**, 116516 (2022)
37. Gao, Y.: PID-based search algorithm: a novel metaheuristic algorithm based on PID algorithm. Expert Syst. Appl. **232**, 120886 (2023)
38. Sowmya, R., Premkumar, M., Jangir, P.: Newton-Raphson-based optimizer: a new population-based metaheuristic algorithm for continuous optimization problems. Eng. Appl. Artif. Intell. **128**, 107532 (2024)
39. Wang, Q., Zhou, G., Song, R., Xie, Y., Luo, M., Yue, T.: Continuous space ant colony algorithm for automatic selection of orthophoto mosaic seamline network. ISPRS J. Photogramm. Remote Sens. **186**, 201–217 (2022)
40. Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), vol. 1, pp. 81–86. IEEE (2001)
41. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. SIMULATION **76**(2), 60–68 (2001)
42. Rao, R., Patel, V.: An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. Int. J. Ind. Eng. Comput. **3**(4), 535–560 (2012)
43. Dhiman, G., Kumar, V.: Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv. Eng. Softw. **114**, 48–70 (2017)
44. Dhiman, G., Kumar, V.: Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl.-Based Syst. **165**, 169–196 (2019)
45. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: algorithm and applications. Future Gener. Comput. Syst. **97**, 849–872 (2019)
46. Kaur, S., Awasthi, L.K., Sangal, A.L., Dhiman, G.: Tunicate Swarm Algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng. Appl. Artif. Intell. **90**, 103541 (2020)
47. Aslan, S., Demirci, S.: Immune plasma algorithm: a novel metaheuristic for optimization problems. IEEE Access **8**, 220227–220245 (2020)
48. Naruei, I., Keynia, F.: Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. Eng. Comput. **38**(Suppl 4), 3025–3056 (2022)
49. Talatahari, S., Bayzidi, H., Saraee, M.: Social network search for global optimization. IEEE Access **9**, 92815–92863 (2021)
50. Chopra, N., Ansari, M.M.: Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst. Appl. **198**, 116924 (2022)
51. Hashim, F.A., Hussien, A.G.: Snake optimizer: a novel meta-heuristic optimization algorithm. Knowl.-Based Syst. **242**, 108320 (2022)
52. Jia, H., Rao, H., Wen, C., Mirjalili, S.: Crayfish optimization algorithm. Artif. Intell. Rev. **56**(Suppl 2), 1919–1979 (2023)
53. Bouaouda, A., Hashim, F.A., Sayouti, Y., Hussien, A.G.: Pied kingfisher optimizer: a new bio-inspired algorithm for solving numerical optimization and industrial engineering problems. Neural Comput. Appl. (2024). https://doi.org/10.1007/s00521-024-09879-5
54. Abdollahzadeh, B., Khodadadi, N., Barshandeh, S., Trojovský, P., Gharehchopogh, F.S., El-kenawy, E.S.M., et al.: Puma optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning. Clust. Comput. (2024). https://doi.org/10.1007/s10586-023-04221-5
55. Tanyildizi, E., Demir, G.: Golden sine algorithm: a novel math-inspired algorithm. Adv. Electr. Comput. Eng. **17**(2), 71–78 (2017)
56. Wang, J., Wang, W.C., Hu, X.X., Qiu, L., Zang, H.F.: Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. Artif. Intell. Rev. **57**(4), 98 (2024)
57. Adegboye, O.R., Deniz Ülker, E.: Gaussian mutation specular reflection learning with local escape operator based artificial electric field algorithm and its engineering application. Appl. Sci. **13**(7), 4157 (2023)
58. Adegboye, O.R., Ülker, E.D.: Hybrid artificial electric field employing cuckoo search algorithm with refraction learning for engineering optimization problems. Sci. Rep. **13**(1), 4098 (2023)
59. Adegboye, O.R., Feda, A.K., Ishaya, M.M., Agyekum, E.B., Kim, K.C., Mbasso, W.F., Kamel, S.: Antenna S-parameter optimization based on golden sine mechanism based honey badger algorithm with tent chaos. Heliyon **9**(11), e21596 (2023)
60. Adegboye, O.R., Feda, A.K., Ojekemi, O.R., Agyekum, E.B., Khan, B., Kamel, S.: DGS-SCSO: enhancing sand cat swarm optimization with dynamic pinhole imaging and golden sine algorithm for improved numerical optimization performance. Sci. Rep. **14**(1), 1491 (2024)
61. Khodadadi, N., Soleimanian Gharehchopogh, F., Mirjalili, S.: MOAVOA: a new multi-objective artificial vultures optimization algorithm. Neural Comput. Appl. **34**(23), 20791–20829 (2022)
62. Gharehchopogh, F.S., Nadimi-Shahraki, M.H., Barshandeh, S., Abdollahzadeh, B., Zamani, H.: Cqffa: a chaotic quasi-oppositional farmland fertility algorithm for solving engineering optimization problems. J. Bionic Eng. **20**(1), 158–183 (2023)
63. Gharehchopogh, F.S., Khargoush, A.A.: A chaotic-based interactive autodidactic school algorithm for data clustering problems and its application on COVID-19 disease detection. Symmetry **15**(4), 894 (2023)
64. Ayar, M., Isazadeh, A., Gharehchopogh, F.S., Seyedi, M.: Chaotic-based divide-and-conquer feature selection method and its application in cardiac arrhythmia classification. J. Supercomput. (2022). https://doi.org/10.1007/s11227-021-04108-5
65. Gharehchopogh, F.S., Ibrikci, T.: An improved African vultures optimization algorithm using different fitness functions for multi-level thresholding image segmentation. Multimed. Tools Appl. **83**(6), 16929–16975 (2024)
66. Gharehchopogh, F.S., Ghafouri, S., Namazi, M., Arasteh, B.: Advances in manta ray foraging optimization: a comprehensive survey. J. Bionic Eng. **21**(2), 953–990 (2024)
67. Gharehchopogh, F.S., Abdollahzadeh, B., Arasteh, B.: An improved farmland fertility algorithm with hyper-heuristic approach for solving travelling salesman problem. CMES-Comput. Model. Eng. Sci. (2023). https://doi.org/10.32604/cmes.2023.024172
68. Shayanfar, H., Gharehchopogh, F.S.: Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl. Soft Comput. **71**, 728–746 (2018)

69. Aslan, S., Demirci, S., Oktay, T., Yesilbas, E.: Percentile-based adaptive immune plasma algorithm and its application to engineering optimization. Biomimetics **8**(6), 486 (2023)
70. Aslan, S., Demirci, S.: An improved immune plasma algorithm with a regional pandemic restriction. SIViP **16**(8), 2093–2101 (2022)
71. Zhong, M., Wen, J., Ma, J., Cui, H., Zhang, Q., Parizi, M.K.: A hierarchical multi-leadership sine cosine algorithm to dissolving global optimization and data classification: the COVID-19 case study. Comput. Biol. Med. **164**, 107212 (2023)
72. Karimzadeh Parizi, M., Keynia, F., Khatibi Bardsiri, A.: Woodpecker Mating Algorithm (WMA): a nature-inspired algorithm for solving optimization problems. Int. J. Nonlinear Anal. Appl. **11**(1), 137–157 (2020)
73. Karimzadeh Parizi, M., Keynia, F.: OWMA: an improved self-regulatory woodpecker mating algorithm using opposition-based learning and allocation of local memory for solving optimization problems. J. Intell. Fuzzy Syst. **40**(1), 919–946 (2021)
74. Parizi, M.K., Keynia, F., Bardsiri, A.K.: HSCWMA: a new hybrid SCA-WMA algorithm for solving optimization problems. Int. J. Inf. Technol. Decis. Mak. **20**(02), 775–808 (2021)
75. Karimzadeh Parizi, M., Keynia, F., Khatibi Bardsiri, A.: Woodpecker mating algorithm for optimal economic load dispatch in a power system with conventional generators. Int. J. Ind. Electron. Control Optim. **4**(2), 221–234 (2021)
76. Zhang, J., Li, H., Parizi, M.K.: HWMWOA: a hybrid WMA–WOA algorithm with adaptive Cauchy mutation for global optimization and data classification. Int. J. Inf. Technol. Decis. Mak. **22**(04), 1195–1252 (2023)
77. Gong, J., Karimzadeh Parizi, M.: GWMA: the parallel implementation of woodpecker mating algorithm on the GPU. J. Chin. Inst. Eng. **45**(6), 556–568 (2022)
78. Abed-alguni, B.H., Paul, D.: Island-based Cuckoo Search with elite opposition-based learning and multiple mutation methods for solving optimization problems. Soft. Comput. **26**(7), 3293–3312 (2022)
79. Abed-alguni, B.H., Alawad, N.A., Barhoush, M., Hammad, R.: Exploratory cuckoo search for solving single-objective optimization problems. Soft. Comput. **25**(15), 10167–10180 (2021)
80. Abed-Alguni, B.H., Paul, D., Hammad, R.: Improved Salp swarm algorithm for solving single-objective continuous optimization problems. Appl. Intell. **52**(15), 17217–17236 (2022)
81. Park, T.S., Namkung, J., Choe, J.C.: Life history of a colonial spicier *Philoponella prominens* (Araneae: Uloboridae) in Korea. Korean J. Biol. Sci. **3**(2), 167–172 (1999)
82. Zhang, S., Liu, Y., Ma, Y., Wang, H., Zhao, Y., Kuntner, M., Li, D.: Male spiders avoid sexual cannibalism with a catapult mechanism. Curr. Biol. **32**(8), R354–R355 (2022)
83. Ma, Y., Hua, Z., Mao, A., Li, D., Zhang, S.: Male opportunistic mating increases with intensity of female sexual cannibalism in 3 web-building spiders. Curr. Zool. **68**(1), 113–119 (2022)
84. Kaveh, A., Eslamlou, A.D.: Water strider algorithm: a new metaheuristic and applications. In: Structures, vol. 25, pp. 520–541. Elsevier, Amsterdam (2020)
85. Fister, I., Fister, I., Jr., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm Evol. Comput. **13**, 34–46 (2013)
86. Awad, N.H., Ali, M.Z., Suganthan, P.N.: Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE congress on evolutionary computation (CEC), pp. 372–379. IEEE (2017)
87. Biedrzycki, R., Arabas, J., Warchulski, E.: A version of NL-SHADE-RSP algorithm with midpoint for CEC 2022 single objective bound constrained problems. In: 2022 IEEE congress on evolutionary computation (CEC), pp. 1–8. IEEE (2022)
88. Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput. Ind. Eng. **158**, 107408 (2021)
89. Abdollahzadeh, B., Soleimanian Gharehchopogh, F., Mirjalili, S.: Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int. J. Intell. Syst. **36**(10), 5887–5958 (2021)
90. Abdollahzadeh, B., Gharehchopogh, F.S., Khodadadi, N., Mirjalili, S.: Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv. Eng. Softw. **174**, 103282 (2022)
91. Abdel-Basset, M., Mohamed, R., Jameel, M., Abouhawwash, M.: Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowl.-Based Syst. **262**, 110248 (2023)
92. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. **13**(5), 945–958 (2009)
93. Mohamed, A.W., Hadi, A.A., Fattouh, A.M., Jambi, K.M.: 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 (2017)
94. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
95. Awad, N.H., Ali, M.Z., Suganthan, P.N., Reynolds, R.G.: An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In: 2016 IEEE congress on evolutionary computation (CEC), pp. 2958–2965. IEEE (2016)
96. Chickermane, H.E., Gea, H.C.: Structural optimization using a new local approximation method. Int. J. Numer. Methods Eng. **39**(5), 829–846 (1996)
97. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**, 17–35 (2013)
98. Lagaros, N.D., Plevris, V., Kallioras, N.A.: The mosaic of metaheuristic algorithms in structural optimization. Arch. Comput. Methods Eng. **29**, 5457–5492 (2022)
99. Gandomi, A.H.: Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans. **53**(4), 1168–1183 (2014)
100. Xiang, S., Wang, L., Xing, L., Du, Y.: An effective memetic algorithm for UAV routing and orientation under uncertain navigation environments. Memetic Comput. **13**(2), 169–183 (2021)
101. Aslan, S., Demirci, S.: An immune plasma algorithm with Q-learning based pandemic management for path planning of unmanned aerial vehicles. Egypt. Inform. J. **26**, 100468 (2024)
102. Aslan, S.: A hospitalization mechanism based immune plasma algorithm for path planning of unmanned aerial vehicles. Int. J. Mach. Learn. Cybern. (2024). https://doi.org/10.1007/s13042-023-02087-y

Yuansheng Gao is currently a Ph.D. student at the College of Computer Science and Technology, Zhejiang University. He received his B.S. degree in July 2024 from the College of Science, Liaoning Technical University. His current research interests are in generative artificial intelligence and metaheuristics. His current research during his Ph.D. focuses on generative artificial intelligence.

Jinpeng Wang is currently an undergraduate student at the College of Science, Liaoning Technical University. His major is Mathematics and Applied Mathematics. He is currently interested in metaheuristics and machine learning.

Changlin Li received his B.S. degree in July 2024 from the College of Science, Liaoning Technical University. Currently, he is working in iSoftStone Information Technology (Group) Co., Ltd. as a software development engineer. His research interests include data mining, machine learning and intelligent optimization.