Neural Computing and Applications
https://doi.org/10.1007/s00521-023-08983-2

S.I.: HYBRID APPROACHES TO NATURE-INSPIRED OPTIMIZATION ALGORITHMS AND THEIR APPLICATIONS

UAV path planning in mountain areas based on a hybrid parallel compact arithmetic optimization algorithm

Ruo-Bin Wang[1,2] · Wei-Feng Wang[1] · Fang-Dong Geng[1] · Jeng-Shyang Pan[4] · Shu-Chuan Chu[5] · Lin Xu[3]

Received: 7 February 2023 / Accepted: 15 August 2023

Abstract
Unmanned Aerial Vehicle (UAV) path planning is one of the core components of its entire autonomous control system. The main challenge lies in efficiently obtaining an optimal flight route in complex environments, especially in mountain areas. To address this, we propose a novel version of arithmetic optimization algorithm (AOA), named parallel and compact AOA (PCAOA). In PCAOA, the compact technique can save the memory of UAV and shorten the calculation time, and the parallel technique can quicken the convergence speed and improve the solution accuracy. In addition, the flight path generated by PCAOA is smoothed with cubic B-spline curves, making the path suitable for a UAV. The performance of PCAOA is demonstrated on 23 benchmark functions. Experimental results show that PCAOA achieves competitive results. Finally, the simulation studies are conducted to verify that PCAOA can successfully acquire a feasible and effective route in different mountain areas.

Keywords UAV path planning · Parallel technique · Compact technique · Arithmetic optimization algorithm

1 Introduction

In recent years, the rapid development and widespread application of unmanned aerial vehicle (UAV) technology have been evident across various industries [1–3]. For UAVs to maintain long-term popularity, it is essential to employ robust algorithms for continuous flight path

Ruo-Bin Wang and Wei-Feng Wang have contributed equally to this work.

✉ Ruo-Bin Wang
wrb@ncut.edu.cn

Wei-Feng Wang
wangweifeng0916@qq.com

Fang-Dong Geng
gfd21@qq.com

Jeng-Shyang Pan
jspan@cc.kuas.edu.tw

Shu-Chuan Chu
scchu0803@gmail.com

Lin Xu
xuyly032@mymail.unisa.edu.au

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China

[2] Beijing Urban Governance Research Center, North China University of Technology, Beijing 100144, China

[3] STEM, University of South Australia, Adelaide 5095, Australia

[4] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

[5] College of Science and Engineering, Flinders University, 1284 South Road, Tonsley, SA 5042, Australia

Published online: 28 November 2023

optimization, especially in complex and adventurous mountainous environments. These terrains, characterized by crisscrossing mountains and rolling hills, pose significant challenges to UAV flight paths. Therefore, path planning is a vital aspect of UAV flight system design [4]. Usually, flight path planning in a mission area is a typical and complex optimization problem that requires efficient algorithms to address. The popular path planning algorithms have been roughly divided into three categories: traditional algorithms, heuristic algorithms, and meta-heuristic algorithms(MHA) [5, 6]. Traditional algorithms, such as artificial potential fields [7], voronoi diagram method [8], and rapidly-exploring random tree [9], are relatively straightforward to implement but are inefficient when faced with complex scenes and multiple obstacles [7, 10]. Heuristic algorithms, i.e., A-star algorithm [11], bi-level programming-based algorithm [12], and greedy algorithm [13], can address multi-obstacle path planning issues, which outperforms traditional methods. Typically, these algorithms have limitations in terms of global optimization capabilities, as they cannot guarantee finding the global optimal path. Additionally, due to the exponential growth of the search space, the running time of these methods tends to significantly increase when in larger problem scales, resulting in low efficiency for path planning. Neither type of algorithm above can overcome the weakness of failing when solving NP-hard problems with a large number of variables and nonlinear objective functions. To overcome the disadvantages, MHAs are proposed with the advantages of simple structure, rapid convergence, and strong robustness, and therefore can solve the problems caused by the above two kinds of algorithms. In recent years, it has been proven to provide efficient, accurate, and fast solutions to path-planning problems [5, 14].

MHAs are inspired by nature laws and phenomena [15], in which they imitate the predation behavior of animals [16], physical phenomena [17], or mathematic theory [18]. In recent years, more MHAs have employed on UAV path planning issues. For example, Ma et al. [19] put forward a chaotic random opposition-based learning and cauchy mutation improved moth-flame optimization algorithm(MFO) which enhances the global search capability of the algorithm. Shao et al. [20] proposed an improved PSO which adopted a chaos-based logistic map to improve the initial particle distribution and used the typical constant acceleration coefficients and maximum velocity to adaptive linear-varying ones. Cekmez et al. [21] developed a multi-colony ant colony algorithm(ACA) to avoid fall into local optimal. Lv et al. [22] presented a hybrid algorithm called HGEOGWO by combining a simplified grey wolf optimizer(GWO) and a personal example learning golden eagle optimizer (GEO). Yao et al. [23] designed a modified hybrid slap swarm algorithm(SSA) and aquila optimizer(AO) called IHSSAO. They introduced the leader mechanism of SSA into AO to strengthen the global search capability. Tong et al. [24] proposed a multi-objective pigeon-inspired optimization (PIO) and combined with mutation strategies of DE. In addition, there are several variants of DE, such as [25–27], and this combined optimization method of MHA and DE variants provides another way of thinking about implementing algorithmic improvements. In many cases, it is not common to directly employ MHAs for the optimization of UAV flight paths. This is primarily because most MHAs tend to have slow convergence rates and are susceptible to getting trapped in local optima, which will negatively impact the performance of UAV path planning.

The improvements of MHAs employed in UAV path planning are categorized as parameters improvement, multi-objective optimization, and hybrid algorithms [28]. Usually, these improvements can facilitate the achievement of path planning with the cost of higher memory consumption and longer computing time, which is expensive for mobile equipment like UAVs. To this point, how to improve an algorithm to achieve lower memory consumption and shorter computing time will be one of the essential issues. However, like another side of a coin, lower memory consumption and shorter computing time mean a compact version of an algorithm, which will spoil the capability of optimum searching. Therefore, how to improve an algorithm to achieve a good performance on optimum searching under the condition of compact running will be another essential issue.

Arithmetic optimization algorithm (AOA) is one of the latest proposed algorithm invented by Mirjalili et al. [29] in 2021. This algorithm utilized the standard arithmetic operators (i.e., Addition, Subtraction, Multiplication, and Division) as a mathematical optimization tool to search the optimal solution under specific criteria. By rationally allocating and combining these four fundamental operators during exploration and exploitation phases, AOA offers simplicity, flexibility, and ease of implementation. Consequently, AOA has found application in various optimization problems [30–32]. However, AOA does suffer from a drawback in its exploration performance, making it prone to getting trapped in local optima and exhibiting poor convergence. To overcome these limitations, researchers have proposed various ideas to improve it, i.e., increasing the exploratory ability [31, 33], parameter adjustment [33, 34], hybrid algorithm [30, 35].

In this paper, we propose a parallel and compact AOA (PCAOA) in view of the performance, time efficiency, and memory-cost of the path planning algorithm. The compact technique [36–38] utilizes distribution characteristics of the initial population to form a probability model. The operation of the initial population was substituted with the

probabilistic model. The model requires fewer variables, which saves more memory and shorten the calculation time. Many researchers have directly adopted this technique due to its efficient memory utilization, i.e., compact GA [36], compact PSO [39], compact CSO [37], etc. However, with the decrease in population, the algorithms lose their original diversities, which will lead to insufficient search and low-accuracy solutions. To avoid the defect, we attempt to employ a parallel technique to optimize the performance of compact AOA. Applying parallel technique is a prevalent way for MHA [40–42] because it can accelerate the converging speed and improve the solution precision. The parallel technology proposed in this paper is applied to solve the problem that AOA is easy to fall into local optimum and poor convergence. This parallel technology cannot only improve the search speed and convergence, but also effectively balance the computing load and reduce the overhead of storing data, thereby further improving the efficiency of the algorithm. Although many researchers have applied parallel and compact technique to various fields [43, 44], there are few applications in the field of path planning, especially in UAV path planning in mountainous areas.

The main innovations of this paper are generalized as follows:

- A compact AOA (CAOA) is proposed, which can reduce the memory consumption of a UAV and shorten the calculation time;
- A novel parallel technique is proposed and applied to improve compact AOA (PCAOA), which can accelerate the convergence speed and improve the precision of the solution;

Meanwhile, to evaluate the performance of PCAOA, we compare it with other state-of-the-art MHAs and test it on 23 benchmark functions. Furthermore, we also apply PCAOA to the UAV path planning problem in complex mountainous areas. By using PCAOA, we are able to efficiently plan the path of the UAV so that it can efficiently navigate in mountainous areas.

The remainder of this paper is organized as follows. Section 2 introduces the problem statement of UAV path planning. Section 3 discusses the methodology including the original AOA, CAOA, and PCAOA in detail. Section 4 conducts the experiments and results of AOA compared with other algorithms based on CEC2013 benchmark functions. Section 5 implements the simulation research. Section 6 concludes this paper.

2 Problem statement

This section discusses the general steps of UAV path planning, including environment modeling, path searching, and path smoothing.

2.1 Environment modeling

Environment modeling is a vital part of path planning. The purpose is to establish an environment model that can be processed by the path planning algorithm. This involves transforming the actual physical space into an abstract representation that facilitates mutual mapping. In this paper, we focus on developing a mountainous environment model to aid UAVs in extracting essential environmental information during flight. By giving the original terrain, mountain size, and mountain concentration, the basic model [45] and mountains range model was established.

$$\begin{aligned} M_1(x,y) &= \alpha \cdot \sin x + \sin(\beta + y) + \gamma \cdot \sin\left(\mu \cdot \sqrt{(x^2+y^2)}\right) \\ &\quad + \varphi \cdot \cos\left(\omega \cdot \sqrt{(x^2+y^2)}\right) + \vartheta \cdot \cos y, \end{aligned} \tag{1}$$

Equation (1) expresses the basic model. Where M_1 represents the height corresponding to the coordinates (x, y) of the model projected on the horizontal plane. $\alpha, \beta, \gamma, \mu, \varphi, \omega, \vartheta$ represent the constant-coefficient which controls the fluctuant of the basic model.

$$M_2(x,y) = \sum_{m=1}^{n} e^{-\left[\left(\frac{x-x_m}{x_{m\_Att}}\right)^2 + \left(\frac{y-y_m}{y_{m\_Att}}\right)^2\right]} \cdot H_m, \tag{2}$$

Equation (2) indicates the model of the mountains. m represents the mth mountain, and n means the number of peaks. (x_m, y_m) represents the center coordinate of the mth peak. (x_{m_Att}, y_{m_Att}) represents the attenuation of the mth peak and controls the slope of the mountain. H_m represents terrain parameters and controls mth mountain height. The final mountain threat model is formed by integrating the basic model into the mountains model. The mountain threat model is shown in Fig. 1.

2.2 Path searching

In the planning space, the starting point has been defined as $S(x_s, y_s, z_s)$ and the desired destination has been defined as $D(x_d, y_d, z_d)$. The searching path of UAVs can be determined by many control points which lie between starting point and desired destination. The control points searched can be represented by $S, C(c_1, c_2, ..., c_n), D$. The single control point can be defined as $c_i = (x_i, y_i, z_i)$.

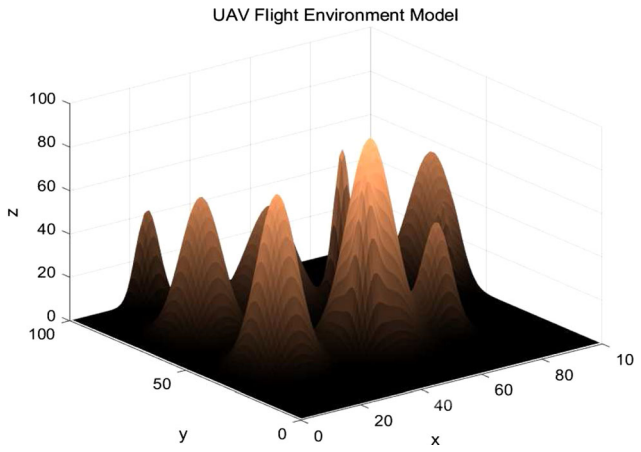


Fig. 1 UAV flight environment model

The purpose of UAV path planning is to minimize flight path costs and meet the requirements of constraint conditions and safe flight. Consequently, path planning requires to establish a path cost function as an index to evaluate path quality. In a majority of studies, the cost function is calculated by employing fuel consumption, maximum climb angle, flying altitude, peak threat. The last three costs are included in the established environment model in the above section. Fuel consumption is associated with flying speed and the length of the flight path. While the fuel consumption can be replaced by the path length on the premise that the UAV always flies at a certain speed. Thus, the optimal cost is considered on the path with the smallest euclidean distance between the start and end points, and it is defined as the following Eq. (3):

$$\text{Cost} = \sum_{i=s}^{D} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}, \tag{3}$$

where, the S represents the initial station, and the D is the destination.

The constraint conditions are shown as follows:

$$M_i > M_2(x_i, y_i) \begin{cases} 0 < x_i \leq x_{\max} \\ 0 < y_i \leq y_{\max} \\ M_1 \leq z_i \leq z_{\max} \end{cases}, \tag{4}$$

The meaning of the above three formulas indicates that the flying height of the UAV should always be higher than the terrain height and can only work in the designated planned path.

2.3 Path smoothing

The improved AOA in this paper plans the path with control points connected by straight lines, which cannot meet the requirements of the actual path planning. To ensure a smooth and flyable path and reduce the running time of the algorithm, we use cubic B-spline curve [46] to smooth the path. Figure 2 shows the strategy of cubic B-spline curve.

In this paper, two equations are employed to construct the mountainous terrain model. Equation (1) is used to generate the base terrain, while Eq. (2) is used for the mountain peak terrain. The specific details of these equations are explained in the manuscript.

For collision detection, the B-spline interpolation detection method is utilized. This method continuously adjusts the shape of the curve by manipulating three control points while ensuring the continuity of curvature.

The overall approach can be summarized as follows:

Firstly, the entire free space is divided into a grid to construct a global set of path points and node vectors. Subsequently, the B-spline basis functions are computed based on these path points and node vectors.

Secondly, the mountainous terrain is constructed. Then, the new control points are computed using a weighted control point matrix. The intersection between the new control points and the mountainous terrain is examined to determine if an update of the control points is necessary.

Finally, the final trajectory is fitted based on the new control points, serving as the outcome of the path planning. This method, utilizing B-spline interpolation and control point adjustments, enables the generation of smooth paths considering the mountainous terrain while performing collision detection.

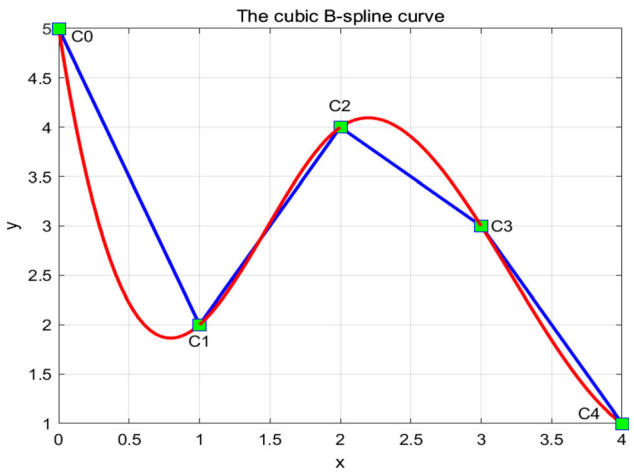


Fig. 2 The smoothing effect figure of cubic B-spline curve. The blue line in the figure is the flight path of the UAV, and the red curve formed after the smoothing effect of cubic B-spline. C_0, C_1, C_2, C_3, C_4 are control points (color figure online)

3 Methodology

3.1 The basic concept of AOA

The AOA [29] has two search phases: exploration and exploitation. The former is ability to explore the search space on a global scale, and its responsibility is to find areas with promising solutions. The latter refers to quickly finding the optimal solution and converging within the range of promising solutions. Before AOA starts working, it ought to choose the search stage through math optimization accelerated function (5).

$$\text{MOA}\ (t) = \ \text{Min}\ + t \times \left(\frac{\text{Max}\ -\ \text{Min}}{T} \right), \tag{5}$$

where MOA(t) is the function value at tth iteration, t and T represent the current and the maximum iteration, respectively. Max and Min represent, respectively, the maximum and minimum values of the accelerated function.

In the outset of AOA, a set of candidate solutions are randomly generated. Then, it switches between exploration and exploitation, as well as iteratively updates until the global optimal solution is found. In the phase of exploration, the division(D) and the multiplication(M) have high dispersion which makes the algorithm search more fully. The positions updated in the exploratory phase are defined by Eq. (6). The subtraction (S) or addition (A) has low dispersion, which helps the algorithm approach the optimal solution in the phase of exploitation. The methods to update the position are proposed by Eq. (7).

$$x_{i,j}(t+1) = \begin{cases} \text{best}\ (x_j) \div (\ \text{MOP}\ + \varepsilon) \times \left[(\ \text{UB}_j - \ \text{LB}_j) \times \mu + \ \text{LB}_j \right], r_2 < 0.5 \\ \text{best}\ (x_j) \times \ \text{MOP}\ \times \left[(\ \text{UB}_j - \ \text{LB}_j) \times \mu + \ \text{LB}_j \right], \ \text{otherwise} \end{cases}, \tag{6}$$

$$x_{i,j}(t+1) = \begin{cases} best\ (x_j) - \ MOP\ \times \left[(\ UB_j - \ LB_j) \times \mu + \ LB_j \right], & r_3 < 0.5 \\ best\ (x_j) + \ MOP\ \times \left[(\ UB_j - \ LB_j) \times \mu + \ LB_j \right], & otherwise \end{cases} \tag{7}$$

where, best (x_j) is the best-obtained solution in the jth position so far, ε is a small integer number, UB_j and LB_j represent the upper value and lower bound value, respectively, in the jth position, μ is a control parameter to adjust search process and is set equal to 0.499, r_2, r_3 are random numbers between 0 and 1. Math optimizer probability is a coefficient that is calculated by Eq. (8).

$$\text{MOP}\ (t) = 1 - \left(\frac{t}{T} \right)^{1/\alpha}, \tag{8}$$

where α is a sensitive parameter and represents the accuracy of exploitation in the whole iterative process.

3.2 Compact AOA

This section introduces the features of CAOA. The compact technique provides an alternative and promises a small amount of memory and computing requirements, which belong to the category of estimation of distribution algorithms (EDA)–with the explicit representation of the population is replaced with a probability distribution. Specifically, compact techniques are population-less, which utilize probabilistic models to represent the distribution characteristics of the original population. Consequently, the algorithm operates using the probabilistic model and conducts continuous search in the decision space by updating the probabilistic model.

The probabilistic model is encoded into a data structure named after the perturbation vector (PV). PV consists of two parts: the mean value μ and the standard deviation σ of a gaussian probability distribution function (PDF) truncated within the interval [–1,1] for each variable. The amplitude of the PDF is normalized to keep its area equal to 1. Based on above-mentioned, the PV is actually a n × 2 matrix:

$$\text{PV}_{\ c} = [\mu_c, \sigma_c], \tag{9}$$

where c is the number of current iterations, μ is initialized to 0, and σ is initialized to 10.

The sampling mechanism for the variable $x[i]$ indexed by i is not a simple process, who is related to the generic candidate solution x from PV. For each design variable indexed by i, a truncated gaussian PDF with mean value μ_i and standard deviation σ_i is associated. The following Eq. (10) defines the PDF:

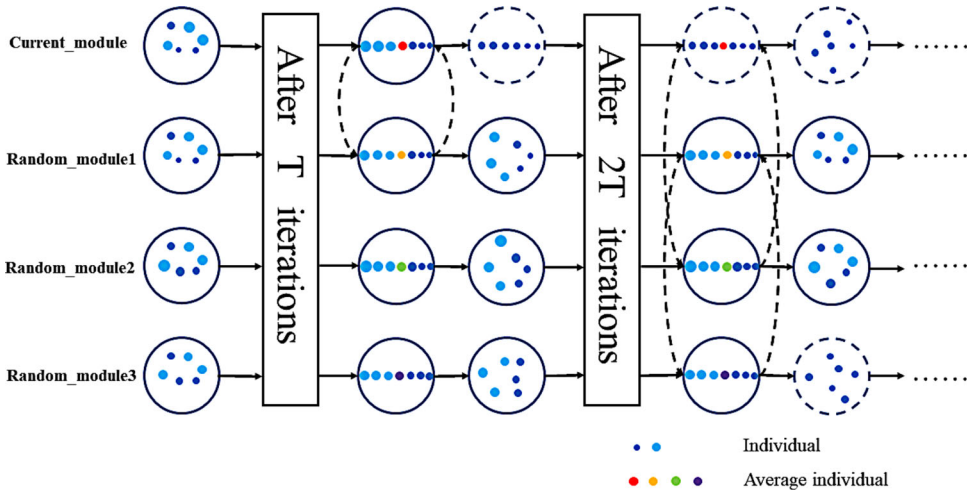


Fig. 3 The process of parallel technique

$$\mathrm{PDF}(\mathrm{truncNorm}\ (x)) = \frac{e^{-\frac{(x-\mu[i])^2}{2\sigma[i]^2}}\sqrt{\frac{2}{\pi}}}{\sigma[i]\left(\ \mathrm{erf}\ \left(\frac{\mu[i]+1}{\sqrt{2}\sigma[i]}\right) - \ \mathrm{erf}\ \left(\frac{\mu[i]-1}{\sqrt{2}\sigma[i]}\right)\right)}, \tag{10}$$

where erf () is the error function. In the process of constructing Chebyshev polynomials, PDF is used to calculate the corresponding cumulative distribution function(CDF), which is defined as Eq. (11):

$$\mathrm{CDF}\ = \frac{\mathrm{erf}\ \left(\frac{\mu+1}{\sqrt{2}\delta}\right) + \ \mathrm{erf}\ \left(\frac{x-\mu}{\sqrt{2}\delta}\right)}{\mathrm{erf}\ \left(\frac{\mu+1}{\sqrt{2}\delta}\right) - \ \mathrm{erf}\ \left(\frac{\mu-1}{\sqrt{2}\delta}\right)}, \tag{11}$$

The relationship between PDF and CDF is $\mathrm{CDF}\ = \int_0^1 \mathrm{PDF}\ (x) rmdx$. So, the sampling mode is changed that the generic design variable $x[i]$ from PV is performed by generating a random number rand(0, 1) from a uniform distribution and then computing the inverse function of CDF in rand (0, 1). During the operation of algorithm, it is of great necessity to update the PV. Specifically, compare which of the two solutions sampled from the PV is better. Let us mark a good fitness value as a winner, the worse as a loser. Regarding to the mean value μ, the update rule is:

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_\mathrm{p}}(\mathrm{winner}_i - \ \mathrm{loser}_i), \tag{12}$$

Regarding to the standard deviation σ, the update rule is:

$$\sigma_i^{t+1} = \sqrt{(\sigma_i^t)^2 + (\mu_i^t)^2 - \left(\mu_i^{t+1}\right)^2 + \frac{1}{N_\mathrm{p}}(winner_i^2 - loser_i^2)}, \tag{13}$$

where N_p represents the actual population.

Based on the above description of the compact method, the CAOA can be implemented, including the following steps. Step 1: generate a random number x with uniform distribution in the range 0–1. Step 2: use PV to generate solution by the inverse function of CDF. The inverse function of CDF is defined as Eq. (14):

$$y = \sqrt{2}\delta\ \mathrm{erf}^{-1}\left(-\ \mathrm{erf}\ \left(\frac{\mu+1}{\sqrt{2}\delta}\right) - x\cdot\ \mathrm{erf}\ \left(\frac{\mu-1}{\sqrt{2}\delta}\right) + x\cdot\ \mathrm{erf}\ \left(\frac{\mu+1}{\sqrt{2}\delta}\right)\right) + \mu, \tag{14}$$

where $\mathrm{erf}^{(-1)}()$ is the inverse function of erf (). Since use the inverse CDF to produce solutions in the range of –1 to 1. Step 3, we need to map the value of y to the actual decision space through the Eq. (15).

$$x_{\mathrm{actual}} = y \times \frac{(\ \mathrm{ub}\ -\ \mathrm{lb}\ )}{2} + \frac{(\ \mathrm{ub}\ +\ \mathrm{lb}\ )}{2}. \tag{15}$$

where ub and lb are the upper and lower bounds of the actual decision space, respectively.

To sum up, the pseudocode of CAOA is shown in algorithm 1.

Table 1 Time and space complexity

Algorithm	AOA	CAOA	PCAOA
Time complexity	O($T * N * D$)	O($T * D$)	O($T * G * M$)
Space complexity	O(N)	O(2)	O(G)

Algorithm 1 Compact AOA

Require: Parameters:d, ub, lb, and, N_p.
Ensure: Global optimum $best$ and its fitness value f_{min}.

for $i = 1 \rightarrow d$ **do**
  Initialize $\mu = 0$, $\sigma = \lambda$, $\lambda = 10$;
end for
Initialize $best = ub$, $f_{min} = \inf$;
while $i < Max\ iteration$ **do**
  Get x_1 from PV via equations (14), (15);
  Update x_1 to get x_2 via equation (6) or (7);
  $[winner, loser]$=compete (x_1, x_2);
  for $i = 1 \rightarrow d$ **do**
    Update PV via equations (12), (13);
    if $fit_{winner} < f_{min}$ **then**
      $best = winner$; $f_{min} = fit_{winner}$;
    end if
  end for
end while

3.3 Compact AOA with parallel technique

During the execution of the algorithm, only two particles participate in each iteration update, which is prone to cause the algorithm to fall into a local optimum. Parallel techniques have become popular to improve the efficiency of population-based MHA in the last decades. By dividing the population into several processing modules, a parallel technique can obtain high-quality results at the same execution time. Integrating parallel techniques into the algorithm not only benefits from the fast search capabilities brought by multiple processing modules but also introduces various communication methods among these modules.

We propose a novel parallel technique that can be applied to communicate in different processing modules. The rules of this technique are as follows:

Firstly, the strategy selects a module randomly, distinct from the current one. After T iterations, it computes and compares the fitness values of individuals in the two selected modules. These individuals are sorted in descending order. The strategy proceeds to calculate the average of the individuals in the two modules, and subsequently filters out the individuals that have fitness values below the calculated average from both modules. The selected individuals are combined to form a new module, integrating the best-performing individuals from both modules. Figure 3 shows the parallel technique.

Secondly, after $2T$ iterations, a perturbation term is added to strategy, which further strengthens the ability of algorithm to jump out of the local optimum. The meaning of this perturbation term is that randomly chooses two modules to compare and update according to above methods. Finally, the new module replaces the old module to perform algorithmic operations. The final and optimal results are compared with previously saved global optimization.

For the sake of clarity, Algorithm 2 demonstrates the working principles of PCAOA.

Algorithm 2 Compact AOA with parallel technique

Require: Parameters: d, ub, lb, N and g.
Ensure: Global optimum $Globalbest$ and its fitness value $Globalfmin$.

Set the number of groups g, each group is $G_i (i \leq g)$;
Initialize $G[i].PV$, $G[i].best$ and $G[i].f_{min}$ of each group;
Initialize $Globalbest = G[1].best$, $Globalfmin = G[1].f_{min}$;
while $t < Max\ iteration$ **do** (note:-$Max\ iteration$ is max iteration.)
  for $i = 1 \rightarrow g$ **do**
    Get x_1, x_2 via equations (14), (15) and (6) or (7);
    $[winner, loser]$=compete (x_1, x_2);
    for $i = 1 \rightarrow d$ **do**
      Update $G[i].PV$ via equations (12), (13);
    end for
    if $fit_{winner} < G[i].f_{min}$ **then**
      $G[i].best = winner$; $G[i].f_{min} = fit_{winner}$;
    end if
    Update $Globalbest$ and $Globalfmin$;
    if $rem(Max\ iteration, T) == 0$ **then**
      perform parallel technique;
    end if
    if $rem(Max\ iteration, 2T) == 0$ **then**
      perform reinforced parallel technique;
    end if
  end for
  Change the worst individual to the best individual by the sorting statement;
  $t = t + 1$;
end while

Table 2 The parameter settings of the related algorithms

Algorithm	Parameters
AOA [29]	Actual $Np = 40$, $max = 1$, $min = 0.2$, $\alpha = 5$, $\mu = 0.499$
CAOA	Actual $Np = 40$, $max = 1$, $min = 0.2$, $\alpha = 5$, $\mu = 0.499$, $\lambda = 10$
PCAOA	Actual $Np = 40$, $max = 1$, $min = 0.2$, $\alpha = 5$, $\mu = 0.499$, $\lambda = 10$, $modules = 4$
APAOA [34]	Actual $Np = 40$, $\mu = 0.499$, $\alpha_{max} = 1$, $\alpha_{min} = 0.1$, $groups = 4$
MVO [17]	Actual $Np = 40$, $WEP_{Max} = 1$, $WEP_{Min} = 0.2$, $p = 6$
SCA [18]	Actual $Np = 40$, $a = 2$
PCSCA [48]	Actual $Np = 40$, $a = 2$,$\lambda = 10$, $groups = 4$
SSA [23]	Actual $Np = 40$, $c_1 = [0, 2]$ $(linearly\ reduce)$
PSO [49]	Actual $Np = 40$, $w = 1.2, c1 = c2 = 2$
ACA [21]	Actual $Np = 40$, $\alpha = 10, \beta = 1, \rho = 0.1, Q = 1$

Table 3 The performance comparison of the three algorithms 30-times running independently on the test functions (F1–F13) (30D)

Algorithm	AOA			CAOA			PCAOA		
F	Ave	Std	Time(s)	Ave	Std	Time(s)	Ave	Std	Time(s)
F1	2.58E-06	1.05E-06	0.635	2.66E-03	3.11E-03	**0.394**	**4.38E-08**	2.403-07	0.536
F2	1.12E-03	1.70E-03	0.682	**2.43E-12**	1.30E-11	**0.408**	1.81E-08	9.90E-08	0.649
F3	6.54E-04	5.22E-04	0.79	5.10E-02	5.55E-02	**0.402**	**5.55E-06**	2.85E-05	0.714
F4	1.62E-02	1.24E-02	0.646	1.67E+01	3.79E+01	**0.391**	**4.54E-06**	2.09E-05	0.613
F5	2.78E+01	3.28E-01	0.661	2.89E+01	6.50E-02	**0.393**	**1.55E-02**	3.35E-02	0.624
F6	2.57E+00	1.81E-01	0.638	5.69E+00	2.70E-01	**0.395**	**6.33E-04**	1.54E-03	0.586
F7	**5.34E-05**	5.19E-05	0.716	2.00E-03	1.41E-03	**0.433**	2.41E-04	1.90E-04	0.662
F8	− 5.65E+03	4.14E+02	0.678	**− 3.69E+03**	6.54E+02	**0.405**	− 9.00E+03	2.19E+01	0.659
F9	1.04E-06	8.01E-07	0.652	**0.00E+00**	0.00E+00	**0.391**	**0.00E+00**	0.00E+00	0.637
F10	3.16E-04	1.28E-04	0.65	1.69E-06	7.20E-06	**0.409**	**4.45E-07**	1.69E-06	0.621
F11	2.31E-03	1.02E-02	0.683	1.36E+02	9.63E+01	**0.394**	**1.33E-06**	7.14E-06	0.634
F12	6.92E-01	2.84E-02	0.873	1.15E+00	1.46E-01	**0.393**	**7.45E-05**	1.40E-05	0.696
F13	2.95E+00	3.73E-02	0.843	2.97E+00	3.88E-02	**0.422**	**2.82E-05**	4.25E-05	0.712

The bold means the optimal value

Ave and *Std*, respectively, represent the average optimal value and standard deviation of the algorithm, and *Times* represents the average running time of the algorithm

3.4 Time and space complexity

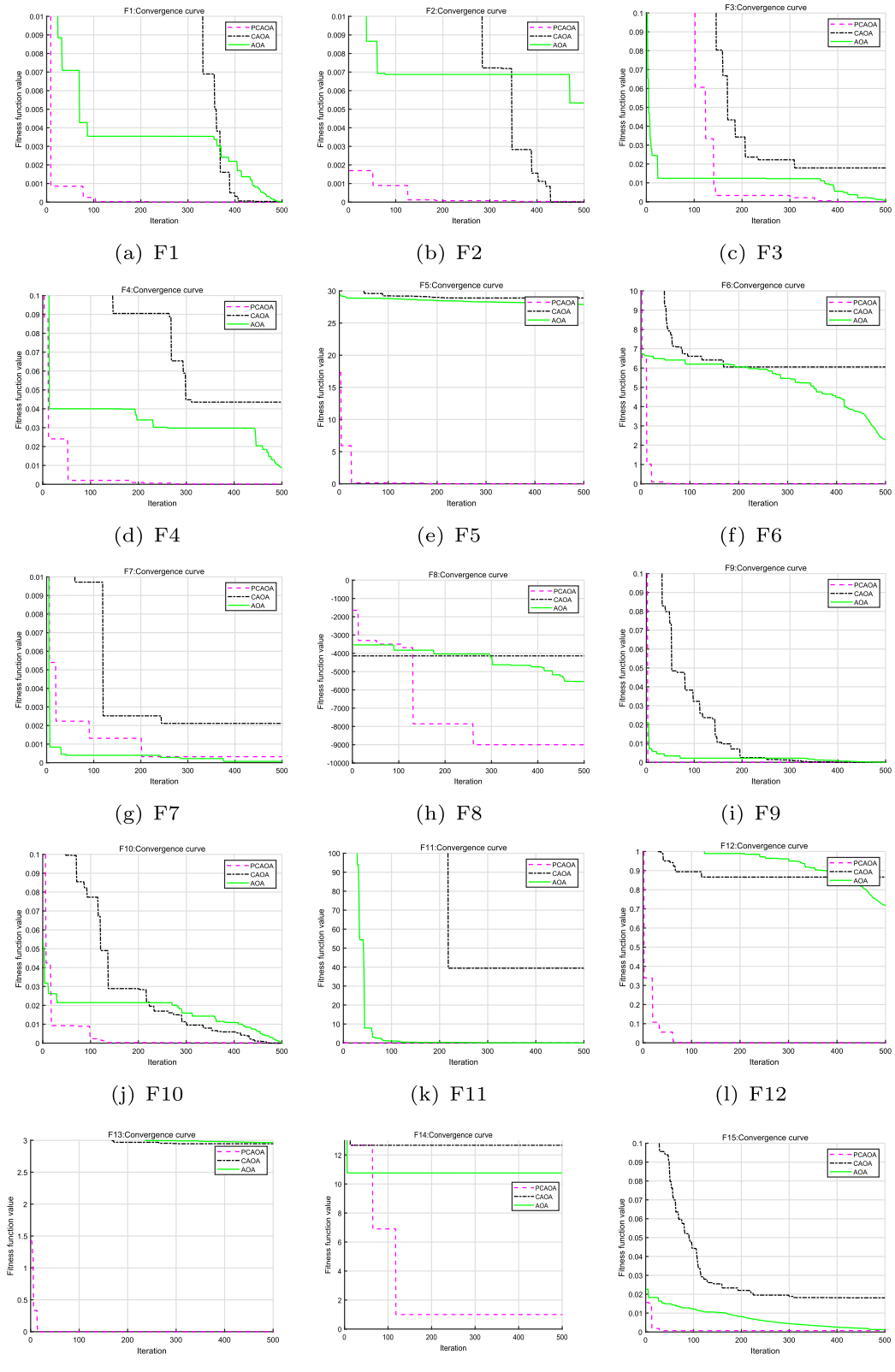
To further improve its practicality and reliability, necessary time and spatial complexity analysis should be conducted before practical application. Because the complexity of algorithms determines their performance and efficiency, it is of great significance for solving practical problems. In order to facilitate comparison, we renames some parameters, where T represents the maximum number of iterations, D represents the dimension, M represents the number of modules, and N represents the overall number. The time and spatial complexity of the algorithm are shown in Table 1.

In a nutshell, compared with AOA, the CAOA obtains a certain improvement not only in time complexity but also in space complexity. In addition, the time complexity and space complexity of PCAOA depend on the number of processing modules. Generally speaking, the number of modules is 4, while the number of populations is generally 40, indicating a significant difference between the two. According to the findings presented in Table 3, it is evident that both CAOA and PCAOA exhibit faster algorithm runtime compared to AOA when subjected to the same experimental parameter settings.

4 Test experiments

This section verifies the effectiveness of parallel and compact AOA (PCAOA). The experiment was tested on MATLAB 2019b with 11th Gen Intel(R) Core(TM) i5-11300H @ 3.10 GHz and 16 GB memory. This paper utilizes 23 benchmark functions [47] to test the performance of PCAOA, in which F1-F7 are unimodal functions, F8–F13 are multimodal functions, and F14-F23 are fixed-dimension multimodal functions. Unimodal functions have a smooth curve shape and only one local extreme point. They are commonly used to test the global search performance of optimization algorithms. Multimodal functions have complex curve shapes and multiple local extreme points. They are typically used to test the local search and convergence performance of optimization algorithms. Fixed-dimension multimodal functions are a special class of multi-peaked functions that contain multiple local optima in specific dimensions. These functions exhibit an exponential growth of optimal solution space with increasing dimensions, posing greater challenges for optimization algorithms. Next, this paper compares the proposed PCAOA with popular MHA. The parameter settings of related algorithms are shown in Table 2.

To achieve a fair comparison, the related algorithms have been implemented employing the same number of population and iterations. The maximum number of iterations is set to 500, the number of population is set to 40. The algorithms are compared using the average, standard deviation, and Friedman ranking (Rank). All results are



(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

(g) F7 (h) F8 (i) F9

(j) F10 (k) F11 (l) F12

(m) F13 (n) F14 (o) F15

Fig. 4 Comparison of convergence curves in F1-F15 between AOA algorithm and its variants

Table 4 The performance comparison of the three algorithms 30-times running independently on the test functions (F14-F23) (fixed dimensions)

Algorithm	AOA			CAOA			PCAOA		
F	Ave	Std	Time(s)	Ave	Std	Time(s)	Ave	Std	Time(s)
F14	9.44E+00	4.56E+00	0.738	1.27E+00	1.06E-10	**0.402**	**9.98E-01**	4.71E-04	0.606
F15	**5.45E-03**	8.59E-03	0.618	3.03E-02	4.04E-02	**0.397**	1.32E-02	4.47E-04	0.524
F16	**− 1.03E+00**	2.28E-11	0.608	− 9.29E-01	7.96E-02	**0.391**	− 1.00E+00	9.97E-02	0.537
F17	**3.98E-01**	1.28E-05	0.623	9.52E-01	6.00E-01	**0.388**	4.43E-01	4.88E-02	0.514
F18	2.19E+01	2.76E+01	0.597	3.84E+02	2.18E+02	**0.394**	**5.47E+00**	3.15E+00	0.544
F19	**− 3.86E+00**	2.79E+04	0.622	− 3.62E+00	5.91E-01	**0.411**	− 2.19E+00	4.09E-01	0.518
F20	**− 3.29E+00**	5.54E-02	0.626	− 2.53E+00	2.73E-01	**0.407**	− 1.27E+00	2.32E-01	0.51
F21	− 7.97E+00	2.78E+00	0.63	− 2.76E+00	1.22E+00	**0.417**	**− 1.01E+01**	5.91E-02	0.566
F22	-8.51E+00	3.03E+00	0.639	− 2.33E+00	1.16E+00	**0.425**	**− 1.04E+01**	9.70E-02	0.531
F23	-6.65E+00	3.55E+00	0.668	− 2.25E+00	1.13E+00	**0.422**	**− 1.05E+01**	2.54E-02	0.548

The bold means the optimal value

obtained by running each algorithm independently 30 times.

4.1 Comparison between AOA and its variants

This subsection compares the above three algorithms, and the parameter settings are shown in Table 2. In reference [29], AOA has been proven to perform best in 30 dimensions (30D). Therefore, this experiment is still based on 30D.

According to Table 3, PCAOA had achieved better results in 10 of the 13 test functions, CAOA performs best in running time. Specifically, PCAOA was not as good as AOA on F7, and it also did not perform as well as CAOA on F2 and F8. However, the performance improvement of PCAOA over AOA was significant. In the column of time, CAOA had taken the top rank, PCAOA ranked the second. This is because the compact technique allows the algorithm to use only two particles to participate in the calculation, and the parallel technique introduces multiple groups into the algorithm. The calculation time of PCAOA is longer than that of CAOA. In Fig. 4a–m, PCAOA has obtained absolute advantages in terms of convergence speed and optimization accuracy. It converged on every test function. To be specific, AOA can only converge on F2, F7, F8, and F11, and the convergence speed is slower than PCAOA. PCAOA and AOA can find optimal values on F1, F3, F9, and F10, but AOA cannot converge. However, PCAOA performs worse than AOA on F7 due to being trapped in a local optimum. CAOA also showed competitive performance, indicating its effectiveness in certain scenarios.

Table 4 shows the performance of the three algorithms in the fixed dimensional multimodal test functions. The performance of both PCAOA and AOA wins five times each on the test function (F14-F23). But upon closer inspection shows that when AOA wins, the gap between the two algorithms is very tiny, and when PCAOA wins, it has a better performance than AOA. The running time of PCAOA on each test function is less than that of AOA, and CAOA still takes the shortest running time. According to Fig. 5a–h, the convergence speed of PCAOA is still the fastest, but the optimization accuracy of PCAOA does not perform so eye-catching. Such as, on F16, F17, F19 and F20, AOA can find a better optimal solution. Besides, in other test functions, PCAOA yet occupies a favorable position, especially in the last three test functions, both convergence and optimization capability are the best.

4.2 Comparison between PCAOA and popular algorithms

In the above experiments, the superiority of PCAOA in 30D has been proved. In this section, this research is to further test the ability of a new algorithm to handle high-dimensions problems in complex environments. This paper picks out four popular algorithms, namely SSA, Sine Cosine Alogrithm(SCA), PCSCA, and MVO, tested with PCAOA on 100 dimensions (100D). In references [34], researchers use parallel techniques to improve AOA and apply the improved algorithm to robot path planning. We not only propose a new parallel technique, but also apply the improved algorithm to UAV path planning in mountainous environments, which highlights the superiority of the parallel technique.

According to the experimental results in Table 5, we can see that PCAOA outperforms other algorithms in 13 test functions. According to the ranking results, PCAOA ranks first, PCSCA ranks second, and APAOA ranks third. From the mean and standard deviation of the experimental data,

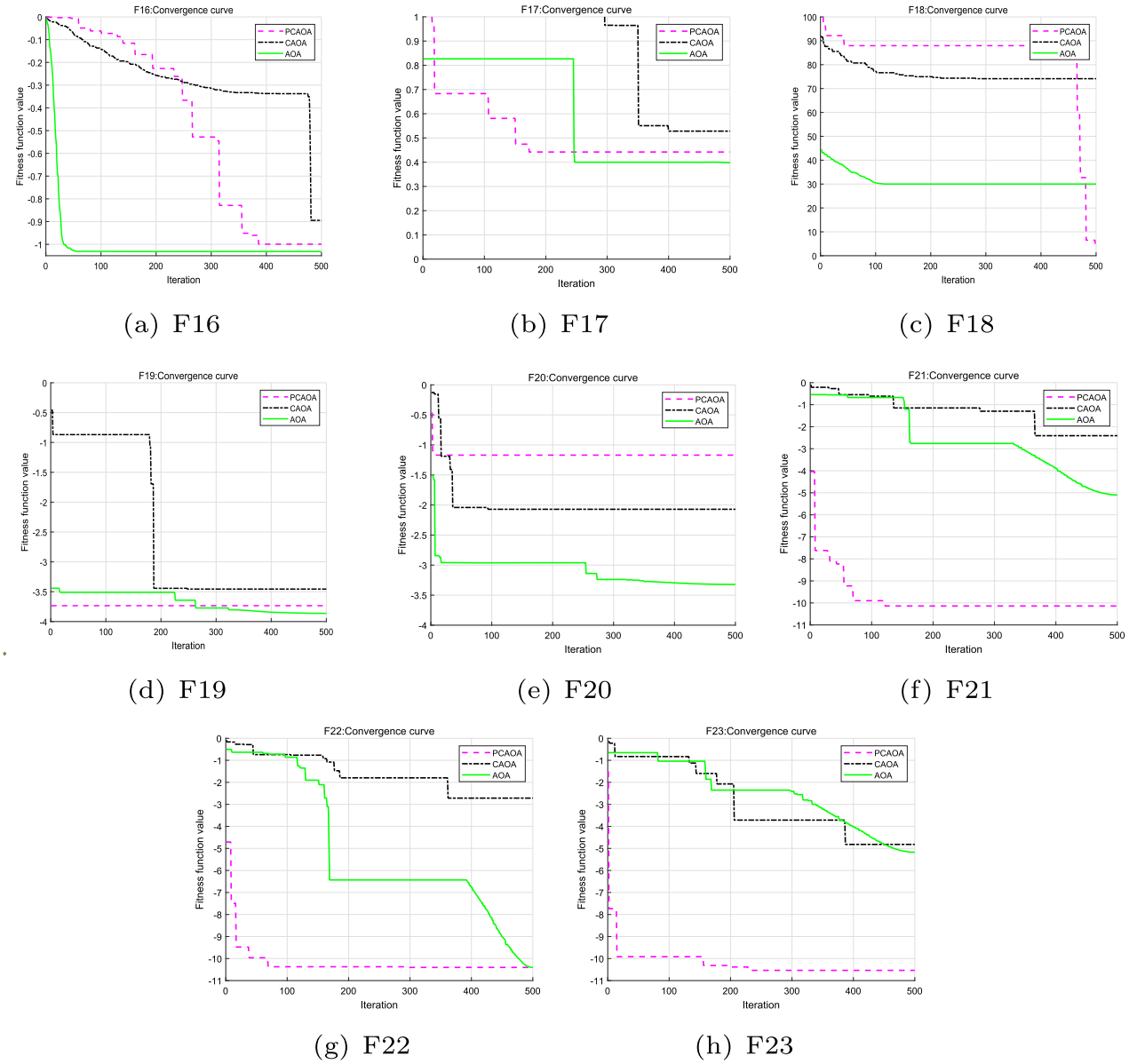


Fig. 5 Comparison of convergence curves in F16-F23 between AOA algorithm and its variants

compared with these three popular algorithms, PCAOA has obviously improved. As a path planning algorithm, APAOA also applies parallel technology. However, its performance is relatively poor compared to PCAOA. This shows that different parallel techniques can lead to large differences in performance. In addition, PCSCA is also improved with parallel and compact techniques, but the performance of PCSCA is still inferior to PCAOA, which further demonstrates the effectiveness of our proposed parallel technique. Through these experimental results, we can conclude that PCAOA performs well on the test function, and the introduction of our parallel technology plays a positive role in the improvement of the algorithm. These findings further support the application potential of our proposed PCAOA algorithm in path planning problems.

The above experiments show that the performance of PCAOA is powerful. Hence, we finally select PCAOA as the main algorithm for UAV path planning.

5 Simulation experiments

In the above section, we describes and constructs the mountain environment model of UAV path planning. This section will implement the simulation research. We set the size of the virtual space as 100*100*100, and the coordinates of the start and destination points are (1, 1, 1) and

Table 5 The performance comparison of the six algorithms 30-times running independently on the test functions (F1–F13) (100D)

Algorithm	PCAOA			APAOA			PCSCA		
F	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	**3.06e-10**	1.52e-09	1	2.92e-01	8.48e-02	3	3.57e-03	8.65e-03	2
F2	**0.00e+00**	0.00e+00	1	1.84e-01	2.51e-02	2	3.96e-01	2.75e-01	3
F3	**0.00e+00**	0.00e+00	1	1.29e+00	4.84e-01	3	3.90e-10	1.45e-09	2
F4	**1.99e-07**	1.09e-06	1	1.86e-01	1.58e-02	3	1.64e-02	1.45e-02	2
F5	3.54E-01	7.06-01	2	1.01e+02	1.24e+00	3	**0.00e+00**	0.00e+00	1
F6	**1.08e-03**	2.12e-03	1	1.87e+01	3.00e+00	3	1.79e+00	6.01e-01	2
F7	3.00E-04	3.38e-04	2	**1.85e-05**	1.87e-05	1	2.62e-02	4.10e-02	3
F8	− **2.98e+04**	6.23e+02	1	− 8.57e+03	2.26e+03	3	− 2.39e+04	2.75e+03	2
F9	**0.00e+00**	0.00e+00	1	1.08e-01	6.85e-02	2	− 9.89e+02	2.37e+00	6
F10	**1.50e-07**	8.20e-07	1	7.27e-02	1.71e-02	2	1.62e-01	2.31e-01	3
F11	**2.00e-09**	1.09e-08	1	2.16e+03	6.40e+02	6	2.91e-03	4.91e-03	2
F12	**5.63e-06**	9.05e-06	1	1.02e+00	6.70e-02	3	3.94e-05	1.35e-04	2
F13	1.63e-04	3.58e-04	2	1.04e+01	1.60e-01	3	**9.81e-05**	4.22e-04	1
Mean Rank	1.23			2.85			2.38		
Rank	1			3			2		
Algorithm	SCA			SSA			MVO		
F	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	1.02e+04	5.83e+03	6	6.75e+02	2.04e+02	5	1.18e+02	1.58e+01	4
F2	6.17e+00	5.23e+00	4	3.74e+01	8.02e+00	5	4.03e+26	2.20e+27	6
F3	2.52e+05	4.50e+04	4	3.97e+04	1.54e+04	5	6.04e+04	7.57e+03	6
F4	8.94e+01	2.65e+00	6	2.47e+01	2.84e+00	4	5.76e+01	5.03e+00	5
F5	1.05e+08	5.71e+07	5	5.17e+04	2.55e+04	4	8.41e+03	8.06e+03	3
F6	1.17e+04	7.05e+03	6	6.85e+02	2.57e+02	5	1.15e+02	2.13e+01	4
F7	1.39e+02	9.70e+01	6	1.82e+00	4.37e-01	5	5.25e-01	1.24e-01	4
F8	− 6.84e+03	4.08e+02	4	− 2.21e+04	1.80e+03	6	− 2.38e+04	1.39e+03	5
F9	2.31e+02	9.46e+01	4	1.85e+02	3.91e+01	3	6.88e+02	7.33e+01	5
F10	1.92e+01	3.71e+00	6	8.55e+00	1.01e+00	5	7.23e+00	5.81e+00	4
F11	1.13e+02	6.76e+01	5	6.55e+00	1.93e+00	4	2.05e+00	1.36e-01	3
F12	3.08e+09	1.57e+09	6	2.23e+01	8.76e+00	5	1.58e+01	5.49e+00	4
F13	5.71e+09	2.27e+09	6	8.28e+02	1.67e+03	5	1.53e+02	2.00e+01	4
Mean Rank	5.23			4.69			4.38		
Rank	6			5			4		

The bold means the optimal value

(100, 100, 80). The simulation research is divided into two parts. The first part is to compare the proposed algorithm with the classical PSO and ACA, and the second part is to study the ability of the proposed algorithm to find the optimal collision free path when dealing with different complex environments. We will examine how PCAOA handles various challenging scenarios and assess its capability to navigate safely and efficiently in these environments.

5.1 Simulation study 1: comparisons between PCAOA, PSO and ACA

The same environmental model is employed in this experiment. The parameters of PSO and ACA are set in Table 2. This paper compares the collision avoidance capability, convergence capability, running time, and path distance of the algorithms.

According to Fig. 6, it can be found that each algorithm can plan a collision free and flying path in complex mountain areas. However, different algorithms find path

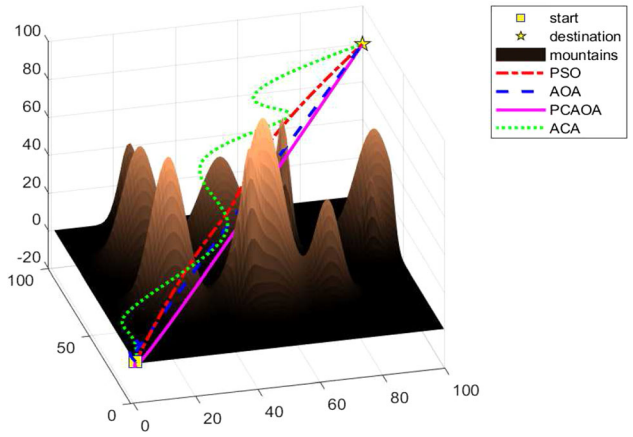


Fig. 6 Path planning results of different algorithms

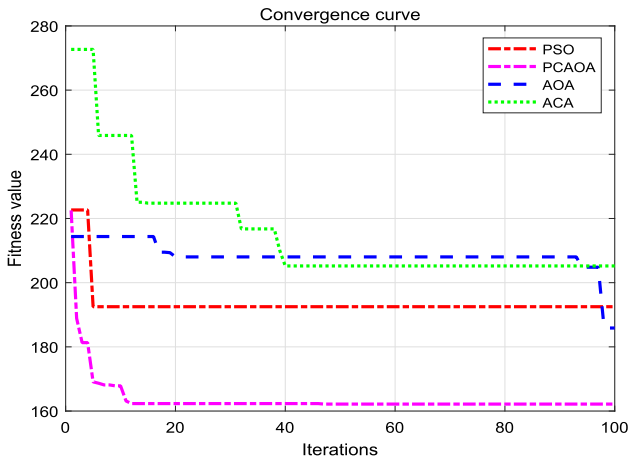


Fig. 7 Convergence curves of different algorithms

Table 6 Path length and running time planned by different algorithms

Algorithms	Average distance	Running times
PCAOA	162.14	15.3
AOA	176.44	43.92
PSO	167.63	37.25
ACA	200.2	146.66

length which is the distinction. Standard AOA does not perform as well as PSO in terms of shortest distance and running time. Nevertheless, after adding parallel technique and compact technique, the capability of AOA optimization is strengthened. Therefore, PCAOA can find the shortest path with the shortest time. The convergence ability of the algorithm is proved in Fig. 7. Within 100 iterations, PSO has the fastest convergence speed, but its optimization ability performs not well. The convergence speed of AOA is worse, but it finds the optimal solution second only to PCAOA. PCAOA performs well both in terms of convergence speed and optimization ability. In Table 6, the average distance of PSO running 10 times is not much different from that of PCAOA. However, it is found from Fig. 7 that PSO is easy to fall into local optimization and its performance is unstable. The performance of PCAOA remains powerful.

5.2 Simulation study 2: impact of different mountain areas on planning capacity

This paper designs an experiment to prove the ability of the algorithm to deal with planning in different environments. According to Fig. 8, four different environmental models are constructed, in which PCAOA can always find a collision-free and optimal path, AOA also can find a collision-free path in most cases. However, when dealing with complex terrain, AOA cannot find a suitable path, as shown in Fig. 8d.

The experiment confirms that PCAOA exhibits robustness and adaptability in handling path planning challenges posed by diverse and intricate environments. Its capability to navigate safely and efficiently even in complex terrains sets it apart from AOA, making it a promising and effective option for UAV path planning in mountainous and challenging regions.

In summary, the PCAOA proposed in this paper showcases superior capabilities in addressing the UAV path planning challenges in mountainous areas. It outperforms traditional algorithms like AOA in terms of collision avoidance, convergence speed, and optimization ability. However, there are still some areas that require improvements, such as the issue of convergence incapability. Further enhancements are needed to achieve even better results in UAV path planning for complex and challenging environments.

6 Conclusions

In this paper, we propose an improved AOA based on hybrid compact and parallel technique to address the challenges of UAV path planning in complex mountainous areas. The compact AOA effectively reduces the memory consumption of UAVs and shortens the computational time of the algorithm. Additionally, we design a novel parallel technique to accelerate the convergence speed and enhance the optimization capability. The combination of these two techniques greatly enhances the performance of AOA.

Through experimental validation, we first verify the performance of the proposed algorithm and compare it with other algorithms. The experimental results demonstrate the significant advantages of the proposed PCAOA in solving

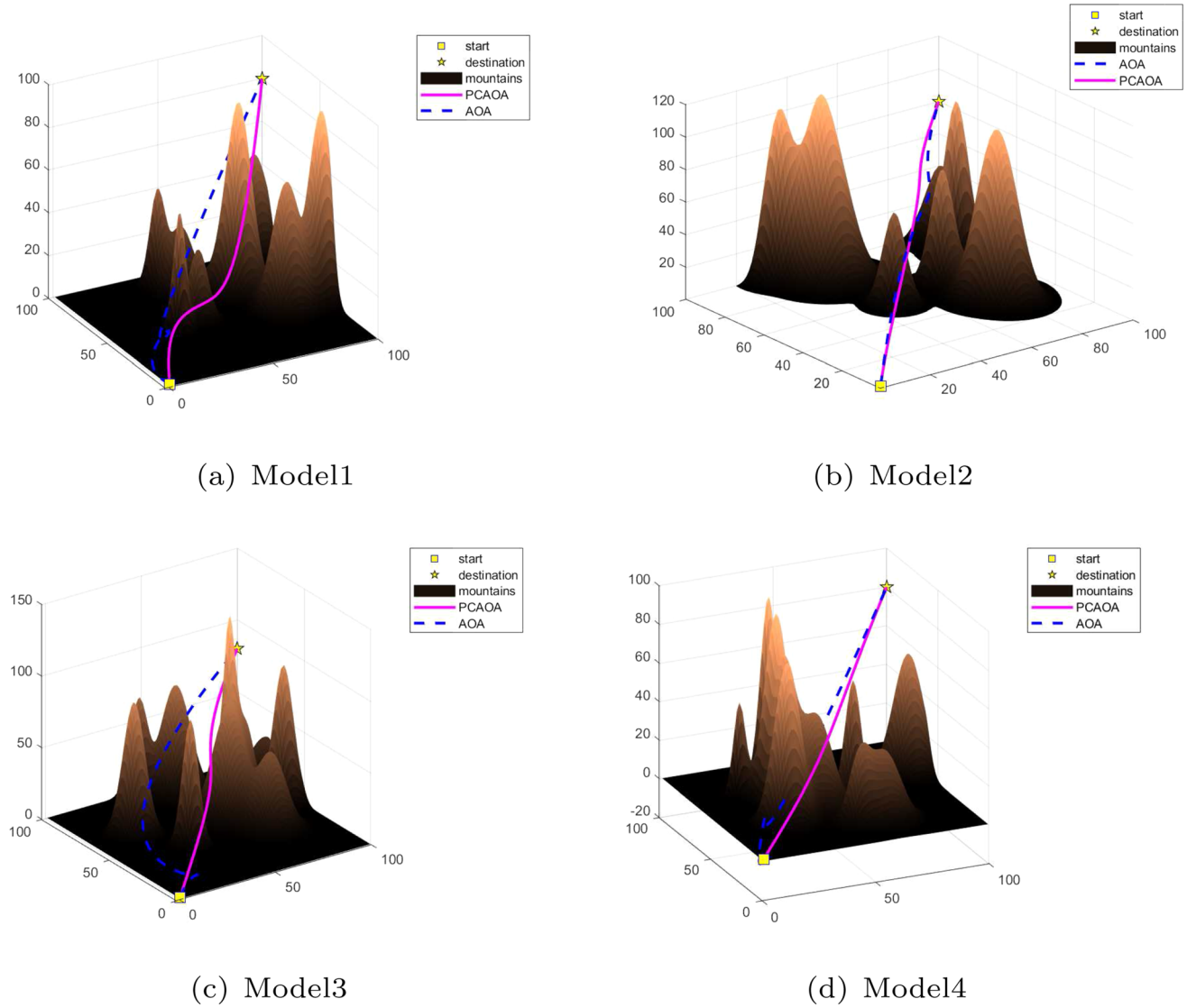


Fig. 8 Comparison of PCAOA and AOA path planning capabilities

UAV path planning problems. According to our experimental data, the PCAOA can quickly find the optimal collision-free path. Compared to the ACA, AOA, and PSO, it reduces the path length by 38.06, 14.3, and 5.49, respectively. Moreover, the PCAOA exhibits excellent computational efficiency. Compared to the ACA, AOA, and PSO, it is 9.56, 2.87, and 2.43 times faster. These experimental results further validate the effectiveness and superiority of our proposed algorithm.

While our research has achieved competitive results in single UAV path planning, we acknowledge the necessity of further investigation in the field of multi-UAV path planning, which will be the future work in this domain. We plan to further investigate and develop algorithms to address the challenges of multi-UAV cooperative flight and path planning, aiming to improve the overall performance and efficiency of the system.

Funding The authors did not receive support from any organization for the submitted work.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Radoglou-Grammatikis P, Sarigiannidis P, Lagkas T, Moscholios I (2020) A compilation of UAV applications for precision agriculture. Comput Netw 172:107148. https://doi.org/10.1016/j.comnet.2020.107148
2. Torresan C, Berton A, Carotenuto F, Di Gennaro SF, Gioli B, Matese A, Miglietta F, Vagnoli C, Zaldei A, Wallace L (2017) Forestry applications of UAVs in Europe: a review. Int J Remote Sens 38(8–10):2427–2447. https://doi.org/10.1080/01431161.2016.1252477
3. Liu H, Chen Q, Pan N, Sun Y, An Y, Pan D (2021) UAV stocktaking task-planning for industrial warehouses based on the improved hybrid differential evolution algorithm. IEEE Trans Indus Inform 18(1):582–591. https://doi.org/10.1109/TII.2021.3054172
4. Zhang Z, Li J, Wang J (2018) Sequential convex programming for nonlinear optimal control problems in UAV path planning. Aerosp Sci Technol 76:280–290. https://doi.org/10.1016/j.ast.2018.01.040
5. Ait Saadi A, Soukane A, Meraihi Y, Benmessaoud Gabis A, Mirjalili S, Ramdane-Cherif A (2022) Uav path planning using optimization approaches: a survey. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-022-09742-7
6. Puente-Castro A, Rivero D, Pazos A, Fernandez-Blanco E (2021) A review of artificial intelligence applied to path planning in UAV swarms. Neural Comput Appl. https://doi.org/10.1007/s00521-021-06569-4
7. Nikranjbar A, Haidari M, Atai AA (2018) Adaptive sliding mode tracking control of mobile robot in dynamic environment using artificial potential fields. J Comput Robot 11(1):1–14. https://doi.org/10.1109/CCDC.2019.8832832
8. Bhattacharya P, Gavrilova ML (2008) Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path. IEEE Robot Autom Magaz 15(2):58–66. https://doi.org/10.1109/MRA.2008.921540
9. Meng L, Qinpeng S, Mengmei Z (2019) Uav 3-dimension flight path planning based on improved rapidly-exploring random tree. In: 2019 Chinese Control and Decision Conference (CCDC), pp. 921–925. IEEE. https://doi.org/10.1109/CCDC.2019.8832832
10. Aggarwal S, Kumar N (2020) Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. Comput Commun 149:270–299. https://doi.org/10.1016/j.comcom.2019.10.014
11. Farid G, Cocuzza S, Younas T, Razzaqi AA, Wattoo WA, Cannella F, Mo H (2022) Modified a-star (a*) approach to plan the motion of a quadrotor UAV in three-dimensional obstacle-cluttered environment. Appl Sci 12(12):5791. https://doi.org/10.3390/app12125791
12. Kang M, Liu Y, Ren Y, Zhao Y, Zheng Z (2017) An empirical study on robustness of uav path planning algorithms considering position uncertainty. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6. IEEE. https://doi.org/10.1109/ISKE.2017.8258825
13. Ahmed S, Mohamed A, Harras K, Kholief M, Mesbah S (2016) Energy efficient path planning techniques for uav-based systems with space discretization. In: 2016 IEEE Wireless Communications and Networking Conference, pp. 1–6 IEEE. https://doi.org/10.1109/WCNC.2016.7565126
14. Saeed RA, Omri M, Abdel-Khalek S, Ali ES, Alotaibi MF (2022) Optimal path planning for drones based on swarm intelligence algorithm. Neural Comput Appl 34(12):10133–10155. https://doi.org/10.1007/s00521-022-06998-9
15. Guo W, Chen M, Wang L, Mao Y, Wu Q (2017) A survey of biogeography-based optimization. Neural Comput Appl 28(8):1909–1926. https://doi.org/10.1007/s00521-016-2179-x
16. Meng Z, Zhong Y, Mao G, Liang Y (2022) Pso-sono: a novel PSO variant for single-objective numerical optimization. Inf Sci 586:176–191. https://doi.org/10.1016/j.ins.2021.11.076
17. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7
18. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
19. Ma M, Wu J, Shi Y, Yue L, Yang C, Chen X (2022) Chaotic random opposition-based learning and Cauchy mutation improved moth-flame optimization algorithm for intelligent route planning of multiple uavs. IEEE Access 10:49385–49397. https://doi.org/10.1109/ACCESS.2022.3172710
20. Shao S, Peng Y, He C, Du Y (2020) Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. ISA Trans 97:415–430. https://doi.org/10.1016/j.isatra.2019.08.018
21. Cekmez U, Ozsiginan M, Sahingoz OK (2016) Multi colony ant optimization for uav path planning with obstacle avoidance. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 47–52. IEEE. https://doi.org/10.1109/ICUAS.2016.7502621
22. Lv J-X, Yan L-J, Chu S-C, Cai Z-M, Pan J-S, He X-K, Xue J-K (2022) A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3d path planning of multiple uavs in power inspection. Neural Computing and Applications. https://doi.org/10.1007/s00521-022-07080-0
23. Yao J, Sha Y, Chen Y, Zhang G, Hu X, Bai G, Liu J (2022) Ihssao: an improved hybrid salp swarm algorithm and aquila optimizer for uav path planning in complex terrain. Appl Sci 12(11):5634. https://doi.org/10.3390/app12115634
24. Tong B, Chen L, Duan H (2021) A path planning method for UAVs based on multi-objective pigeon-inspired optimisation and differential evolution. Int J Bio-Inspired Comput 17(2):105–112. https://doi.org/10.1504/IJBIC.2021.114079
25. Meng Z, Pan J-S, Tseng K-K (2019) Pade: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowl Based Syst 168:80–99. https://doi.org/10.1016/j.knosys.2019.01.006
26. Meng Z, Yang C (2022) Two-stage differential evolution with novel parameter control. Inf Sci 596:321–342. https://doi.org/10.1016/j.ins.2022.03.043
27. Meng Z (2023) Dimension improvements based adaptation of control parameters in differential evolution: a fitness-value-independent approach. Exp Syst Appl 223:119848. https://doi.org/10.1016/j.eswa.2023.119848
28. Hussain K, Mohd Salleh MN, Cheng S, Shi Y (2019) Metaheuristic research: a comprehensive survey. Artif Intell Rev 52(4):2191–2233. https://doi.org/10.1007/s10462-017-9605-z
29. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609
30. Abualigah L, Almotairi KH (2022) Dynamic evolutionary data and text document clustering approach using improved aquila optimizer based arithmetic optimization algorithm and differential evolution. Neural Comput Appl 34(23):20939–20971. https://doi.org/10.1007/s00521-022-07571-0
31. Agushaka JO, Ezugwu AE (2021) Advanced arithmetic optimization algorithm for solving mechanical engineering design

problems. PLoS One 16(8):0255703. https://doi.org/10.1371/journal.pone.0255703
32. Guo H, Sun Z, Sun H, Ebrahimian H (2021) Optimal model of the combined cooling, heating, and power system by improved arithmetic optimization algorithm. Energy Sour Part A Recov Utili Environ Effects. https://doi.org/10.1080/15567036.2021.1966138
33. Kaveh A, Hamedani KB (2022) Improved arithmetic optimization algorithm and its application to discrete structural optimization. Structures, vol 35. Elsevier, UK, pp 748–764
34. Wang R-B, Wang W-F, Xu L, Pan J-S, Chu S-C (2021) An adaptive parallel arithmetic optimization algorithm for robot path planning. J Adv Transp. https://doi.org/10.1155/2021/3606895
35. Abualigah L, Ewees AA, Al-qaness MA, Elaziz MA, Yousri D, Ibrahim RA, Altalhi M (2022) Boosting arithmetic optimization algorithm by sine cosine algorithm and levy flight distribution for solving engineering optimization problems. Neural Comput Appl 34(11):8823–8852. https://doi.org/10.1007/s00521-022-06906-1
36. Harik GR, Lobo FG, Goldberg DE (1999) The compact genetic algorithm. IEEE Trans Evolu Comput 3(4):287–297. https://doi.org/10.1109/4235.797971
37. Pan J-S, Song P-C, Chu S-C, Peng Y-J (2020) Improved compact cuckoo search algorithm applied to location of drone logistics hub. Mathematics 8(3):333. https://doi.org/10.3390/math8030333
38. Xue X, Lu J (2020) A compact brain storm algorithm for matching ontologies. IEEE Access 8:43898–43907. https://doi.org/10.1109/ACCESS.2020.2977763
39. Neri F, Mininno E, Iacca G (2013) Compact particle swarm optimization. Inf Sci 239:96–121. https://doi.org/10.1016/j.ins.2013.03.026
40. Mehanović D, Kečo D, Kevrić J, Jukić S, Miljković A, Mašetić Z (2021) Feature selection using cloud-based parallel genetic algorithm for intrusion detection data classification. Neural Comput Appl 33(18):11861–11873. https://doi.org/10.1007/s00521-021-05871-5
41. Ding S, Du W, Zhao X, Wang L, Jia W (2019) A new asynchronous reinforcement learning algorithm based on improved parallel PSO. Appl Intell 49(12):4211–4222. https://doi.org/10.1007/s10489-019-01487-4
42. Chai Q-w, Chu S-C, Pan J-S, Hu P, Zheng W-M (2020) A parallel woa with two communication strategies applied in dv-hop localization method. EURASIP J Wirel Commun Netw 1:1–10. https://doi.org/10.1186/s13638-020-01663-y
43. Wang X, Pan J-S, Chu S-C (2020) A parallel multi-verse optimizer for application in multilevel image segmentation. IEEE Access 8:32018–32030. https://doi.org/10.1109/ACCESS.2020.2973411
44. Song P-C, Pan J-S, Chu S-C (2020) A parallel compact cuckoo search algorithm for three-dimensional path planning. Appl Soft Comput 94:106443. https://doi.org/10.1016/j.asoc.2020.106443
45. Pehlivanoglu YV (2012) A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous uav. Aerosp Sci Technol 16(1):47–55. https://doi.org/10.1016/j.ast.2011.02.006
46. Wan N, Xu D, Ye H (2018) Improved cubic b-spline curve method for path optimization of manipulator obstacle avoidance. In: 2018 Chinese Automation Congress (CAC), pp. 1471–1476. IEEE. https://doi.org/10.1109/CAC.2018.8623056
47. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102. https://doi.org/10.1109/4235.771163
48. Zhang S, Fan F, Li W, Chu S-C, Pan J-S (2021) A parallel compact sine cosine algorithm for tdoa localization of wireless sensor network. Telecommun Syst 78(2):213–223. https://doi.org/10.1007/s11235-021-00804-y
49. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE. https://doi.org/10.1109/ICNN.1995.488968