Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3606895, 22 pages
https://doi.org/10.1155/2021/3606895

Research Article

An Adaptive Parallel Arithmetic Optimization Algorithm for Robot Path Planning

Ruo-Bin Wang[1], **Wei-Feng Wang**[1], **Lin Xu**[2], **Jeng-Shyang Pan**[3], **and Shu-Chuan Chu**[3,4]

[1]*School of Information Science and Technology, North China University of Technology, Beijing 100043, China*
[2]*STEM, University of South Australia, Adelaide 5095, Australia*
[3]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[4]*College of Science and Engineering, Flinders University, 1284 South Road, Tonsley, South Australia 5042, Australia*

Correspondence should be addressed to Ruo-Bin Wang; robin945@163.com and Lin Xu; xuyly032@mymail.unisa.edu.au

Received 16 July 2021; Revised 15 August 2021; Accepted 3 September 2021; Published 24 September 2021

Academic Editor: Chi-Hua Chen

Path planning is one of the hotspots in the research of automotive engineering. Aiming at the issue of robot path planning with the goal of finding a collision-free optimal motion path in an environment with barriers, this study proposes an adaptive parallel arithmetic optimization algorithm (APAOA) with a novel parallel communication strategy. Comparisons with other popular algorithms on 18 benchmark functions are committed. Experimental results show that the proposed algorithm performs better in terms of solution accuracy and convergence speed, and the proposed strategy can prevent the algorithm from falling into a local optimal solution. Finally, we apply APAOA to solve the problem of robot path planning.

1. Introduction

In recent years, automotive engineering has been an emerging area. Among it, the automotive robots have been widely employed in industry and social life and played an important role, especially during the pandemic of COVID-19. The existing literatures have explored issues on robot path planning. For example, Sarabu et al. [1] proposed the method of using dual robotic arms to collaboratively pick apples in an unstructured orchard environment. Sehestedt et al. [2] utilized a path planning algorithm based on probabilistic routes to sample Hidden Markov models through robots learning human motion patterns. Tiseni et al. [3]conducted an evaluation about measuring the energy dose delivered by a robot-based moving source of Ultraviolet type-C irradiation (UV-C) radiation at different locations in an indoor environment with genetic algorithm (GA).

In summary, robot path planning has become one of the key problems in the domain of robot automatic control. At present, there are three kinds of popular algorithms employed in path planning:

(1) Based on searching: Dijkstra algorithm [4] and A* algorithm [5]

(2) Based on probability: rapidly exploring Random Trees (RRT) [6] and rapidly exploring Random Trees* (RRT*) [7]

(3) Based on metaheuristic algorithm: particle swarm optimization (PSO) [8]

In this study, a metaheuristic algorithm is employed to optimize the robot path planning. Compared with other algorithms, a metaheuristic algorithm can achieve a stable convergence and avoid trapping into a local optimal solution, especially when facing a complex environment.

In the past few decades, metaheuristic algorithms have been regarded as an effective way to address optimization issues in various fields and have been widely used to improve the performance of real-world problems. Some popular metaheuristic algorithms are ant colony optimization algorithm (ACO) [9], ant lion optimizer (ALO) [10], moth-flame optimization algorithm (MFO) [11], multiverse optimizer (MVO) [12], sine cosine algorithm (SCA) [13], whale

optimization algorithm (WOA) [14], dragonfly algorithm (DA) [15], cat swarm optimization (CSO) [16], and so on. These algorithms have been successfully applied to different fields of medical industry [17], image processing [18], transportation [19], and the like. However, according to the No Free Lunch Theorem (NFL) [20], there is no metaheuristic algorithm suitable for handling all types of optimization problems. Therefore, researchers have continuously proposed new metaheuristic algorithms in recent years and have continuously improved them to deal with increasing complexities in the real world.

Currently, there are existing researches using metaheuristic algorithm for path planning. Zhang et al. [21] proposed a new improved artificial fish swarm algorithm (IAFSA) to process the mobile robot path planning problem in a real environment. Qin et al. [22] proposed an improved PSO with mutation operator to get the optimal path. Li et al. [23] proposed an improved GA, which integrated a fuzzy logic control algorithm to self-adaptively adjust the probabilities of crossover and mutation in GA. Wang et al. [24] presented a new weighted adjacency matrix to determine the walking direction, and the best ant and the worst ant are introduced for the adjustment of pheromone to facilitate the searching process. The proposed algorithm guarantees that robots are able to find an optimal path. Zhang et al. [25] proposed hybrid method with simulated annealing algorithm and ant colony algorithm to apply to robot path planning. Huang and Tsai [26] combined GA with PSO in evolving new solutions by applying crossover and mutation operators on solutions constructed by particles, which avoids the premature convergence and time complexity. Aiming at the problems of slow response speed, long planning path, unsafe factors, and a large number of turns in the traditional path planning algorithm, Dao et al. [27] proposed a novel multiobjective method for optimal mobile robot path planning based on WOA.

Usually, the original algorithm is improved and then applied to path planning. The types of algorithm improvement are roughly divided into three categories: improvements on parameters, hybrid algorithm, and multiobjective algorithms.

The arithmetic optimization algorithm (AOA) [28] is proposed in 2021, with the characteristics of simplicity, fewer control parameters, and stronger output performance. It has been proved that AOA performs well on welded beam design problem, tension/compression spring design problem, pressure vessel design problem, and the like. However, there is still seldom research on the improvement of AOA and its application on robot path planning. The algorithm is relatively new; therefore, there are relatively few improvements on it [29, 30], which deserves further improvement in certain areas. Parallel strategy is an effective algorithm optimization method, which can communicate and exchange information among groups. Some parallel algorithms are parallel SCA [31], parallel GA [32], parallel MVO [18], parallel WOA [33], parallel SCO [34], and parallel GWO [35]. However, there is still a lack of improvements on parallel strategies for AOA. Although AOA has superiority in some aspects over some other algorithms, there are still defects of converging slowly in complex environments or high-dimensional problems and is prone to fall into local optimal. Aiming at the algorithm defects, we propose an adaptive parallel AOA. Adaptive parameters can balance the capabilities of exploration and exploitation. Parallel strategy refers to strengthening the communication among groups and reducing the defects of the original AOA, such as premature convergence, search stagnation, and easy to fall into the local optimal search space. The main contributions of this article are summarized as follows:

(1) We propose a novel parameter adaptive equation to control the AOA sensitive parameter α, which can balance the capabilities of exploration and exploitation

(2) We propose a novel parallel communication strategy and apply it to AOA, which can strengthen the communication and information exchange among groups and avoid falling into the local optimal solution

(3) The improved AOA is applied to an optimization problem of 2D robot path planning

The rest of the article is organized as follows. Section 2 describes the principle of the original AOA and robot path planning. Section 3 introduces the improved AOA about self-adaption and parallel strategy. The performance of the proposed algorithm is tested, and the results of different algorithms are shown and analysed in Section 4. Section 5 introduces the application of the proposed algorithm in robot path planning. Finally, conclusions are delivered in Section 6.

2. Related Works

2.1. Original AOA. The AOA is inspired by the application of arithmetic operators in solving arithmetic problems [28], using simple arithmetic operators, such as addition, subtraction, multiplication, and division as mathematical optimization, to search for the optimal solution that meets the standards from a set of candidate solutions. Like MVO [12], AOA is also divided into the phrases of exploration and exploitation. Exploration refers to finding a range of promising optimal solutions in a broad search space, and exploitation refers to quickly finding the optimal solution within the range of promising solutions and converging.

Before running, the search phase of AOA should be confirmed. Therefore, a coefficient calculated by equation (1) is employed for choosing the phrase of exploration or exploitation, which is Math Optimizer Accelerated (MOA) function. Another coefficient defined by equation (2) is Math Optimizer Probability (MOP), which is employed for controlling the range of candidate solutions in the phase of exploring or exploiting.

$$\mathrm{MOA}(t) = \mathrm{Min} + t \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{T}\right), \tag{1}$$

$$\mathrm{MOP}(t) = 1 - \left(\frac{t}{T}\right)^{1/\alpha}, \tag{2}$$

where MOP (t) and MOA (t) are values at tth iteration, and t and T represent the current iteration and the maximum iteration, respectively. Max and Min represent, respectively, the maximum and minimum values of the accelerated function, and α is a sensitive parameter and represents the accuracy of exploitation in the whole iterative process.

In the stage of exploration, the division (D) and the multiplication (M) have high dispersion that probably leads to divergence. However, the communication between operators is increased to support the search in the exploration phase after several iterations. The position updates in the exploring phase are defined as

$$x_{i,j}(t+1) = \begin{cases} \text{best}(x_j) \div (\text{MOP} + \varepsilon) \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & r_2 < 0.5, \\ \text{best}(x_j) \times \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & \text{otherwise}, \end{cases} \tag{3}$$

where $x_{i,j}(t)$ represents the jth position of the ith solution at current iteration, 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, and r_2 is a random number between 0 and 1.

In the stage of exploiting, the subtraction (S) or addition (A) has low dispersion, which helps them approach the optimal solution. The position updating equations are proposed for the exploitation parts as

$$x_{i,j}(t+1) = \begin{cases} \text{best}(x_j) - \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & r_3 < 0.5 \\ \text{best}(x_j) + \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & \text{otherwise}, \end{cases} \tag{4}$$

where r_3 is a random number between 0 and 1 and other parameters are set as in equation (3).

2.2. Robot Path Planning. Autonomous navigation is one of the most important issues of intelligence. Robot path planning is designed for the target of avoiding barriers in the procedure of navigation. The navigation consists of four essential requirements known as perception, localization, cognition, and planning. Motion control in path planning is the most important part [36]. Normally, there are many feasible paths for a robot to reach the target from the starting position. However, in actual situations, the best feasible path is selected based on the shortest distance, path smoothness, minimum energy consumption, and the like [37] The path planning strategy of mobile robots can be categorised as classical methods and heuristic methods. However, classical methods do not always find the optimal path and are often locked in some local optimum. In addition, in the presence of multiple barriers or dynamic environments, some of them may not provide suitable solutions. To avoid the limitations of classical methods, heuristic methods is employed [38].

In this article, the improved AOA is applied to the robot path planning based on the position of static barriers. The mathematical model of robot path planning will be introduced in Section 5.1.

3. Improved AOA

3.1. Adaptive AOA. There are two hyperparameters in AOA: α and μ, where μ controls the absolute step size of the algorithm position update. Under a given objective function, it is a fixed value and will not change with the number of iterations. However, α is used to calculate MOP, and MOP is used to control the magnitude of the change of AOA in the iterative process. Experiments have shown that different α values will affect MOP and thus the performance of the algorithm [28]. Through experimental comparison, the results show that when $\alpha < 1$, MOP is a convex function, which is conducive to the full exploration of the algorithm in the early stage, rather than quickly converging to a local area. Based on these findings, we propose an adaptive change of α. A small value of α may cause the algorithm falling into a local optimum, and a large value of α may lead to insufficient search and even cannot find the optimal solution, which will affect the algorithm's efficiency. Therefore, it is useful to introduce an adaptively changed α to improve the balance between the exploration phase and exploitation phase's search capabilities of the algorithm. In this article, we change the size of α according to the fitness value and proposes a formula such as equations (5) and (6).

$$\alpha'(t) = \begin{cases} \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \times \dfrac{f - f_{\min}}{f_{\text{avg}} - f_{\min}}, & f \le f_{\text{avg}}, \\ \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \times \dfrac{f_{\max} - f}{f_{\max} - f_{\text{avg}}}, & f > f_{\text{avg}}, \end{cases} \tag{5}$$

$$\alpha(t) = 1 - \alpha'(t) + \varepsilon, \tag{6}$$

where $\alpha(t)$ is the value of α at the tth iteration, $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum value and the maximum value of α according to the experience value, respectively; f, $f_{\min}$, $f_{\max}$, f_{avg} are the fitness value, the minimum fitness value, the maximum fitness value, and the average fitness value, at

the current iteration, respectively; and ε is a very small positive integer. The value of α' is related to the value of α, which is inversely proportional, and the values of $f - f_{\min}/f_{\text{avg}} - f_{\min}$ and $f_{\max} - f/f_{\max} - f_{\text{avg}}$ are proportional to α'. It is better for the values of $f - f_{\min}/f_{\text{avg}} - f_{\min}$ and $f_{\max} - f/f_{\max} - f_{\text{avg}}$ be closer to 1. Therefore, we replace $f_{\max} - f_{\min}$ with $f_{\text{avg}} - f_{\min}$ and $f_{\max} - f_{\text{avg}}$ in equations (5) and (6).

3.2. Parallel Strategy. In order to improve computing speed and solve large and complex problems, parallel strategy is widely used, such as Data Mining [39] and Deep Learning [40]. In the field of intelligent algorithms, GA is one of the earliest algorithms employed parallel strategy [32]. Experiments have proved that adding a parallel strategy can implement multipoint parallel search in space, which increases the communication between populations. Due to the use of multiple CPUs, the search speed is accelerated. Therefore, the algorithm with the parallel strategy performs better than the original algorithm [31–35]. With the increment of data, a single communication strategy is usually not adequate. Therefore, Roddick [41] expanded the single communication strategy into three communication strategies and applied them to PSO. Experiments proved that the three communication strategies successfully increased the efficiency of PSO. Then, many researchers improve the parallel method based on Chang's theory. For example, Yang et al. [31] and Zhu et al. [19] proposed a parallel strategy of multiple groups and multiple strategies. Each population will update according to its own communication strategy. When a certain number of iterations are reached, the populations communicate with each other and exchange information. Their final update methods are the same, which use the optimal value of one group instead of the worst value of another group. Nasrabadi et al. [35] adopted the parallel strategy of multiple groups of the same strategy. At the beginning, each group used the same strategy to evolve independently. The populations reach a certain number of iterations and began to exchange information.

In this article, we add randomness to the communication strategy and use different strategies in local search and global search, which can help groups fully communicate, as shown in Figure 1. Specifically, for the local: two groups are arbitrarily selected, and one group of particles with the best fitness is substituted for the other group of particles with the worst fitness every T iteration. For the global, the global best particle replaces the worst particle in the group every $2T$ iterations. The purpose of using these two strategies is to enhance the randomness of the algorithm, strengthen the communication between populations, and avoid premature convergence of the algorithm, thereby improving the robustness of the algorithm.

Following to the above introduction, Figure 2 shows the model of adaptive parallel AOA (APAOA) search process. It can be seen that the optimal solution of the algorithm is first updated by the multiplication and division operator in the exploration stage, then updated by the addition and subtraction operator in the exploitation stage and finally find the global optimal solution. The pseudocode of the APAOA algorithm is shown in Algorithm 1.

4. Experimental Analysis of the Algorithm

In this section, the experimental research will be committed. Section 4.1 introduces 18 benchmark functions provided by [42], including unimodal functions, multimodal function, and fixed-dimensional multimodal functions. Section 4.2 tests APAOA and original AOA in 30 dimensions (30D). Section 4.3 compares APAOA with other popular intelligent algorithms on 100 dimensions (100D).

The algorithms are compared using mean, standard deviation, and Friedman ranking (Rank) test. These are popular evaluation indexes in algorithm comparisons.

4.1. Benchmark Functions. In this article, 18 benchmark functions are employed as one of the main methods to test the performance of intelligent algorithms. Benchmark functions are shown in Tables 1–3.

4.2. Comparison with the Original AOA and APAOA with 30D. To verify the performance of the APAOA algorithm, it is compared with the original AOA on unimodal functions, multimodal functions, and fixed-dimensional multimodal function. AOA is proved to perform the best on 30D in the study by Abualigah et al. [28]. The dimensions in this experiment are set to 30, the number of populations is set to 40, the maximum number of iterations is 500, and populations in APAOA with parallel strategy are divided into four groups, and each algorithm is run independently for 30 times. In Tables 4–6, the Ave represents the mean value, the Std indicates the standard deviation, and the Best represents the optimal value. The bold part of the tables indicate victory.

According to the results in Table 4, compared with AOA, APAOA has advantages in 5 of the 7 unimodal functions. Among them, APAOA is not as good as AOA in terms of mean value or optimal value in test functions F5 and F6, but the gap between the two algorithms is close, and APAOA is better than AOA in standard deviation in test function F5. This shows that the dispersion degree of solution accuracy of APAOA on F5 tends to be stable.

Figure 3 shows the convergence speed and accuracy of the two algorithms on the test function F1–F7. APAOA is better than AOA on the whole, except for the poor optimization ability of F5 and F6. In the unimodal test function of AOA, only F4 and F7 can converge. However, APAOA can converge quickly and find better values than AOA on the other six test functions except test function F5.

On the six multimodal test functions, APAOA is better than AOA on F8, F9, F10, and F12, and the solution accuracy is 10 times higher than the original, such as F9 and F10. APAOA is only worse than AOA on F11. On the test function F13, the two algorithms are almost the same in mean, standard deviation, and optimal value; the value of AOA is more stable than that of APAOA, and APAOA can find a better solution than AOA. The test results are shown in Table 5.

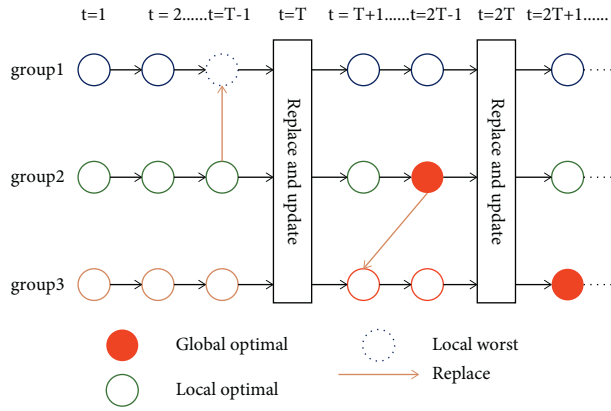


Figure 1: Parallel communication strategy process.

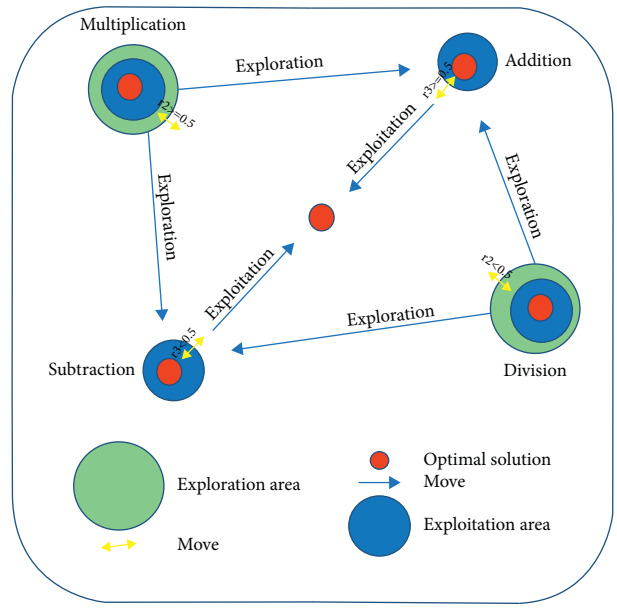


Figure 2: APAOA search process model.

According to Figure 4 and Table 5, on F13, neither algorithm can converge. On the other five test functions, APAOA can converge quickly and find the better solution. However, APAOA is not as good as AOA in all aspects on F11. This shows that APAOA still has the problem of insufficient accuracy of solution.

The benchmark functions F14–F18 are fixed-dimensional multimodal functions and low dimensional. It can be found that the performance of the two algorithms is close according to the results in Table 6. APAOA has superior performance on F14 and F18, whereas the performance on F15 and F17 is not as good as AOA. On F16, the performance of the two algorithms is almost the same. From Figure 5, both algorithms can converge, but the convergence speed of APAOA is still faster than that of AOA on the whole, except for F15.

To sum up, the APAOA we proposed, which adaptively changes the parameter α, can make algorithm jump out of the local optimal solution such as F5 and F13, and the advantage of parallel strategy is that the algorithm can quickly converge and find the better solution in the face of complex and high-dimensional environment, such as F1–F4. However, it also performs not good in some functions such as F11 and F15.

4.3. Comparisons with Popular Algorithm. In Section 4.2, APAOA has been compared with AOA in unimodal function, multimodal function, and fixed-dimensional multimodal function. Because the dimension of test function F14–F18 is fixed, the dimension of F1–F13 can be expanded. In this section, in order to further evaluate the ability of APAOA to solve high-dimensional problems, the APAOA was compared with popular optimization algorithms on F1–F13. Table 7 shows popular algorithms and their parameter settings. To achieve a fair comparison, all algorithms are tested on 100D, and the common parameters settings are the same as those in section 4.2.

```
Initialize the AOA parameters μ, groups;
Initialize the candidate solutions randomly (candidate solutions: i = 1, ..., N);
while (t < T)
  Calculate the fitness function for the given solutions;
  Find the best solution so far;
  for g = 1: groups
    if rand <0.5
      Perform parallel communication strategy 1 every T iteration;
    else
      Perform parallel communication strategy 2 every 2T iteration;
    end if
    Update the α value using equations (5) and (6);
    Update the MOA value using equation (1);
    Update the MOP value using equation (2);
    for i = 1: N
      for j = 1: dimensions
        Generate random values between [0, 1];
        if r1 < MOA//enter in exploration phase
          Update the solutions by equation (3);
        else//enter in exploiting phase
          Update the solutions by equation (4);
        end if
      end for
    end for
  end for
  t = t + 1;
end while
Return the best solution.
```

Algorithm 1: Pseudocode of APAOA.

Table 1: Unimodal test functions.

Name	Function	Dim	Range	f min
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
F2	$f(x) = \sum_{i=0}^{n} \lvert x_i \rvert + \prod_{i=0}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
F3	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	30	[−100, 100]	0
F4	$f(x) = \max_i \{\lvert x_i \rvert, \quad 1 \le i \le n\}$	30	[−100, 100]	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	30	[−30, 30]	0
F6	$f(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[−100, 100]	0
F7	$f(x) = \sum_{i=0}^{n} i x_i^4 + \text{random}[0, 1)$	30	[−128, 128]	0

The experimental results in Table 8 show that the APAOA has achieved good results on the test functions F1–F13, in which the dimension is set to 100, and it is in the first place in the Friedman ranking test. On F1–F13, APAOA achieved the best performance except that F6 and F11 did not perform as expected. The performance of APAOA on F6 is second only to SSA, but there is a large gap between them. APAOA only performed not good in F11. In the whole, APAOA outperforms other popular algorithm on most functions.

Next, the convergence and search ability of different algorithms are evaluated, and the results are shown in Figure 6. According to Table 8, it is found that there is a big gap between the search ability of other popular algorithms and APAOA. If all convergence curves are drawn on one graph, the convergence curve of APAOA will be seen like a straight line. Therefore, the convergence curve of APAOA is drawn separately (right of Figure 6) in this article. Among the 13 test functions, APAOA has poor convergence ability only on F8, F11 and F13, but its ability to search the optimal solution is stronger than other algorithms. On F2 and F4, APAOA converges extremely well and can find better solutions than other algorithms. On F2, the search capabilities of all algorithms differ greatly, causing the algorithm's convergence curve to become a straight line.

5. Application in Robot Path Planning

5.1. Robot Path Planning Mathematical Model. The robot path planning problem generally include three aspects: environment modelling, path searching, and path smoothing.

Table 2: Multimodal test functions.

Name	Function	Dim	Range	f min
F8	$f(x) = \sum_{i=1}^{n} [-x_i \sin(\sqrt{\lvert x_i \rvert})]$	30	$[-500, 500]$	$-418.9829 \times n$
F9	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
F10	$f(x) = -20\exp(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2} - \exp 1/n\sum_{i=1}^{n}\cos(2\pi x_i) + 20 + e)$	30	$[-32, 32]$	0
F11	$f(x) = 1 + 1/4000\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}}$	30	$[-600, 600]$	0
F12	$f(x) = x/n\{10\sin(\pi y_1)\} + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) + \sum_{i=1}^{n} u(x_i, 10, 100, 4)]$, where $y_i = 1 + x_i + 1/4$, $u(x_i, a, k, m)\begin{cases} K(x_i - a)^m, & \text{if } x_i > a, \\ 0, & -a \le x_i \ge a, \\ K(-x_i - a)^m, & -a \le x_i \end{cases}$	30	$[-50, 50]$	0
F13	$f(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 1 + \sin^2(2\pi x_n)) + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

Table 3: Fixed-dimension multimodal test function.

Name	Function	Dim	Range	f min
F14	$f(x) = [1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^{2} (x_i - a_{ij})]^{-1}$	2	$[-65, 65]$	1
F15	$f(x) = \sum_{i=1}^{11} [a_i - x_1 (b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4]^2$	4	$[-5, 5]$	0.00030
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
F17	$f(x) = (x_2 - 5.1/4\pi^2 x_1^2 + 5/\pi x_1 - 6)^2 + 10(1 - 1/8\pi)\cos\ x_1 + 10$	2	$[-5, 5]$	0.398
F18	$f(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_i + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2, 2]$	3

Table 4: Unimodal function test on 30D.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F1	$2.95e-06$	$1.10e-06$	$1.53e-06$	$\mathbf{3.72e-07}$	$2.52e-07$	$4.74e-09$
F2	$1.29e-03$	$1.38e-03$	$2.83e-05$	$\mathbf{5.84e-04}$	$2.81e-03$	$8.05e-06$
F3	$6.07e-04$	$5.38e-04$	$1.24e-05$	$\mathbf{2.05e-04}$	$4.25e-04$	$1.26e-07$
F4	$1.42e-02$	$1.04e-02$	$0.30e-02$	$\mathbf{1.33e-02}$	$3.31e-02$	$1.03e-05$
F5	$\mathbf{2.77e+01}$	$2.47e-01$	$2.81e+01$	$2.87+01$	$1.35e-01$	$2.85+01$
F6	$\mathbf{2.56e+00}$	$2.10e-01$	$2.34e+00$	$3.81e+00$	$1.22e+00$	$2.64e-01$
F7	$5.89e-05$	$5.87e-05$	$9.58e-05$	$\mathbf{7.38e-06}$	$8.00e-06$	$3.32e-06$

Table 5: Multimodal function test on 30D.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F8	$-5.49e+03$	$3.26e+02$	$-5.35+03$	$\mathbf{-5.25e+03}$	$1.72e+02$	$-8.22e+03$
F9	$8.15e-07$	$7.52e-07$	$1.46e-06$	$\mathbf{1.91e-09}$	$9.18e-08$	$2.26e-10$
F10	$2.92e-04$	$1.65e-04$	$4.23e-04$	$\mathbf{4.29e-05}$	$7.51e-05$	$1.27e-05$
F11	$\mathbf{1.25e-03}$	$4.69e-03$	$2.09e-05$	$4.81e+01$	$1.24e+01$	$1.60e+02$
F12	$6.89e-01$	$2.86e-02$	$0.71e+00$	$\mathbf{5.10e-01}$	$3.40e-01$	$0.51e+00$
F13	$2.96e+00$	$2.63e-02$	$2.97e+00$	$2.96e+00$	$2.67e-02$	$2.87e+00$

Table 6: Fixed-dimensional multimodal function test.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F14	$1.05e+01$	$3.50e+00$	$5.93e+00$	$\mathbf{1.00e+01}$	$5.82e+00$	$9.98e-01$
F15	$\mathbf{7.02e-03}$	$1.03e-02$	$1.31e-02$	$1.22e-02$	$1.27e-02$	$1.10e-03$
F16	$-1.03e+00$	$2.25e-11$	$-1.03e+00$	$-1.03e+00$	$1.33e-02$	$-1.03e+00$
F17	$\mathbf{3.98e-01}$	$2.05e-06$	$4.00e-01$	$4.76e-01$	$1.67e-01$	$4.00e-01$
F18	$9.30e+00$	$1.16e+01$	$3.00e+00$	$\mathbf{3.00e+00}$	$2.03e-03$	$3.00e+00$

5.1.1. Environment Modelling. In an actual working environment, a robot has to face the complexity and quick change. To regularize the experimental environment, we use a grid-like method for modelling. Figure 7 shows the robot workplace model. The robot path planning problem is transformed into the use of algorithms to make the robot walk from the source to target, avoiding collisions with barriers during the procedure and finding an optimal collision-free path.

5.1.2. Path Searching. On the basis of environmental modelling, the issue of finding an optimal path is transferred into the issue of an objective function obtaining the optimal value. Assuming that in an ideal state, the robot is a straight collision-free path from the source to the target, and the objective function is shown as

$$D = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2}, \tag{7}$$

where D represents the distance between the source and the target, (x_s, y_s) represents the source, and (x_t, y_t) represents the target. However, in the actual working environment, the robot will encounter barriers to prevent it from moving forward. Firstly, for the robot workplace model, a Cartesian coordinate system is constructed, and the horizontal axis and vertical axis of the two-dimensional plane are equidistantly divided to form a set of intersections Node $= \{(x_i, y_i), \quad i = 1, 2, 3, \ldots, n\}$, the sum of the distances between the particles randomly generated by the algorithm is L and is given as:

$$L = \sum_{j=1}^{k} \sqrt{\left(x_{j+1} - x_j\right)^2 + \left(y_{j+1} - y_j\right)^2}, \quad (k < n). \tag{8}$$

Secondly, we use equation (9) to search for points in the barriers on the working path of the robot. If any point on the path is within the barrier, the penalty function equation (10) is added to the objective function.

$$d = \frac{\sqrt{(x_i - x_b)^2 + (y_i - y_b)^2}}{r_b} - 1, \quad \begin{cases} d > 0, & \text{not in barrier,} \\ d < 0, & \text{in barrier,} \end{cases} \tag{9}$$

where (x_b, y_b) is the coordinate of the centre point of the obstacle b, r_b is the radius of the obstacle, and $b = 1, 2, 3, \ldots$.

$$\text{Penal} = \text{Penal} + \text{mean}\left(\min\left(d, 0\right)\right), \tag{10}$$

where Penal is the penalty function.

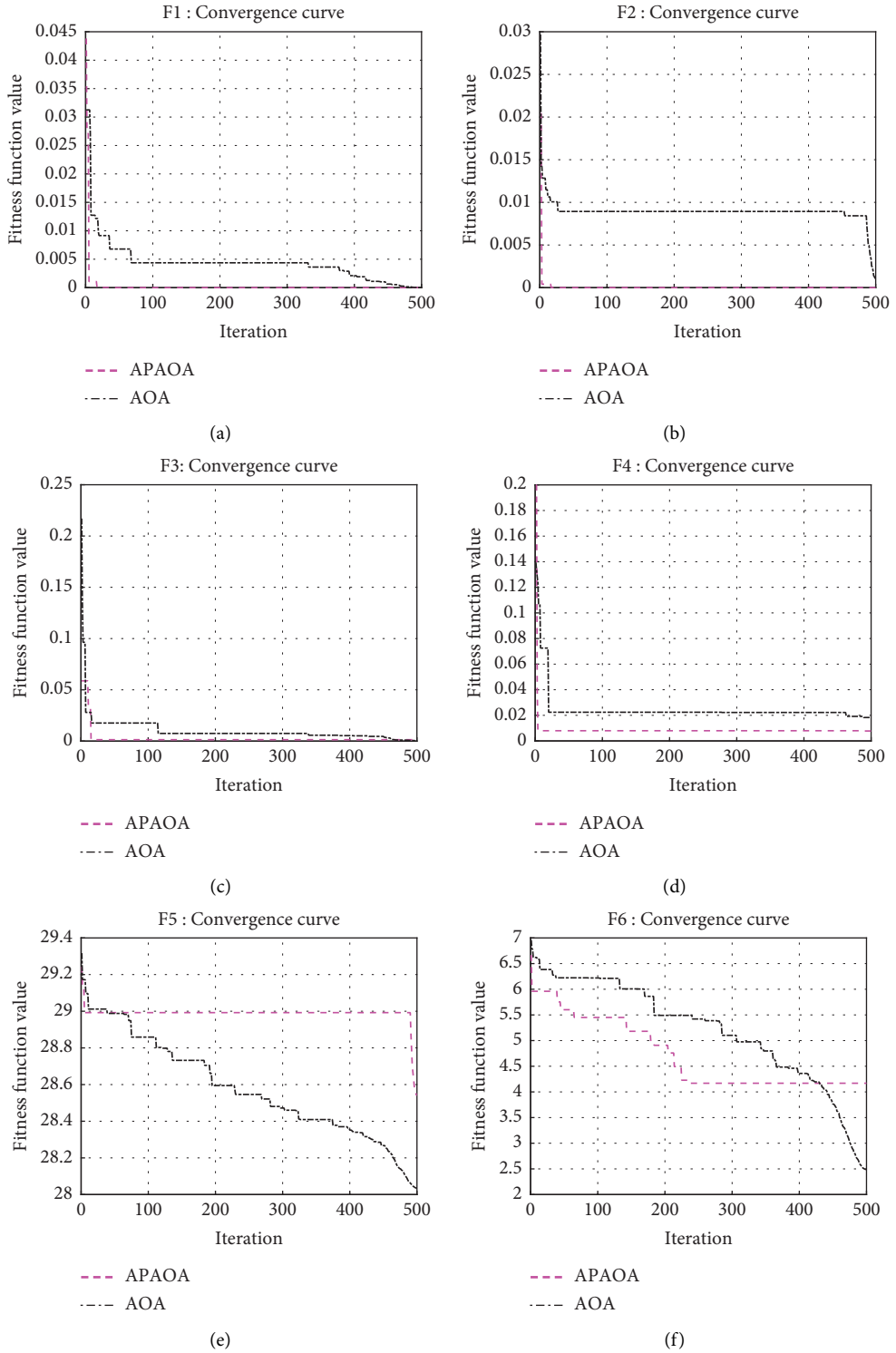
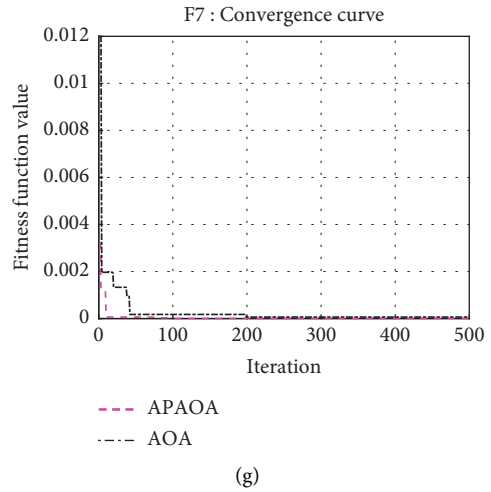


Figure 3: Continued.



(g)

Figure 3: Comparison of convergence curves between APAOA and AOA in F1–F7.

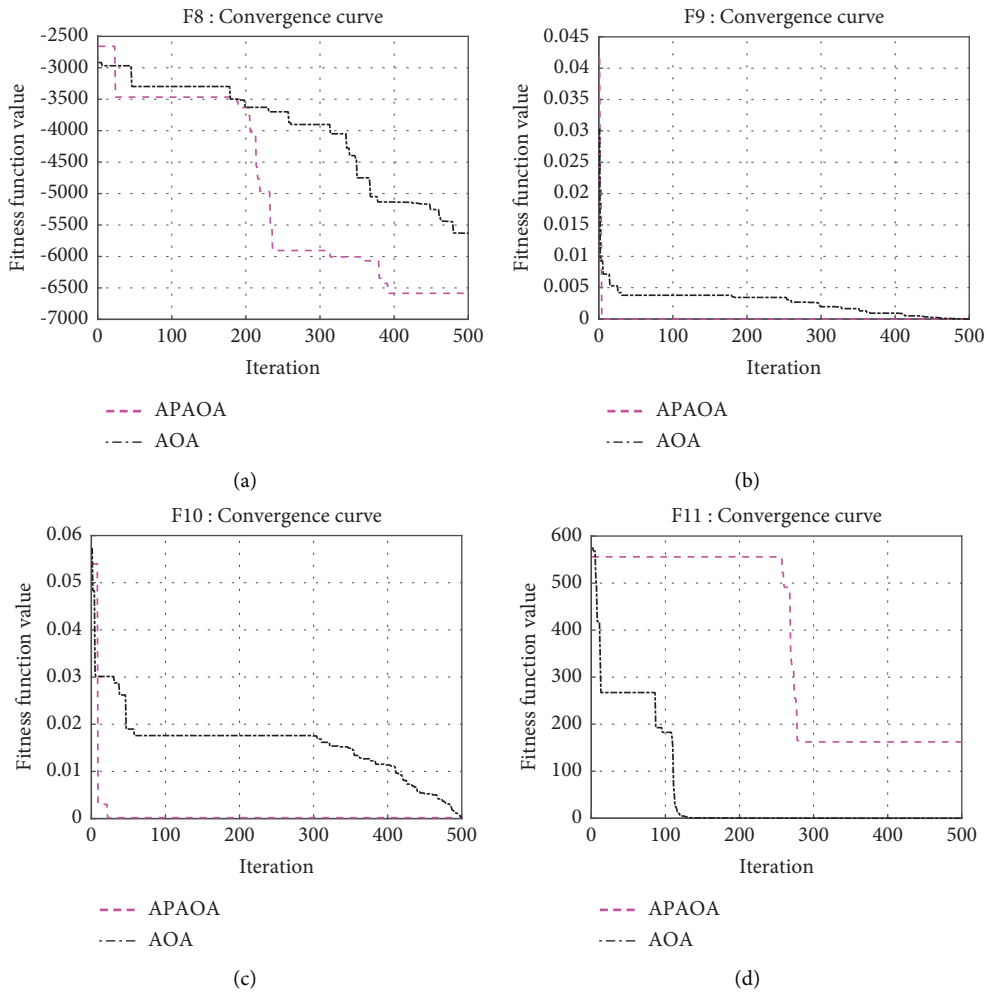


Figure 4: Continued.

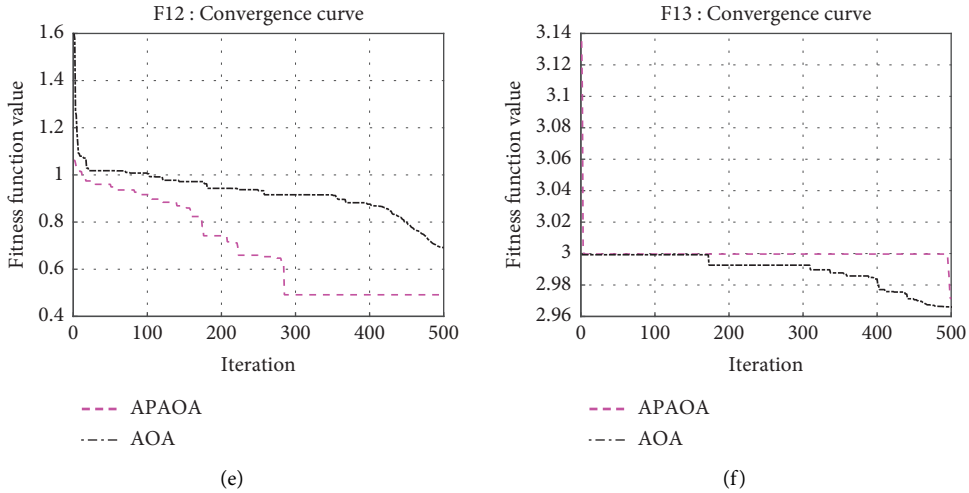


Figure 4: Comparison of convergence curves between APAOA and AOA in F8–F13.

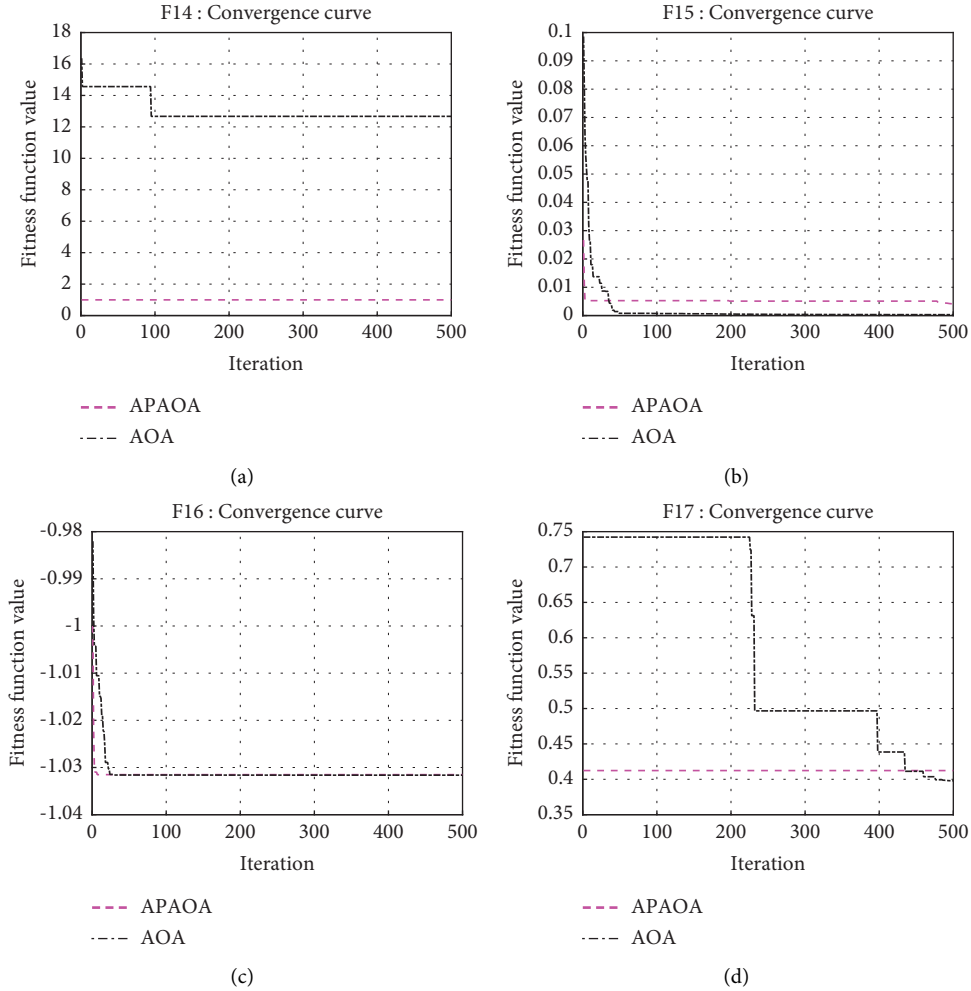
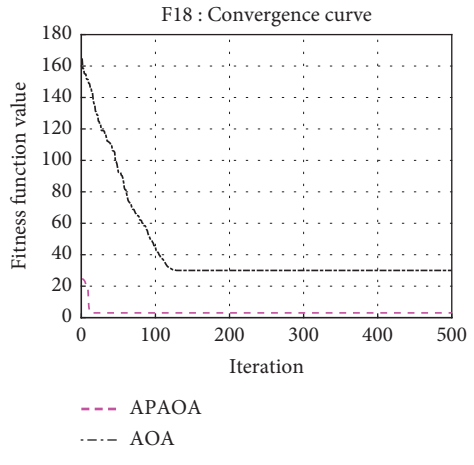


Figure 5: Continued.

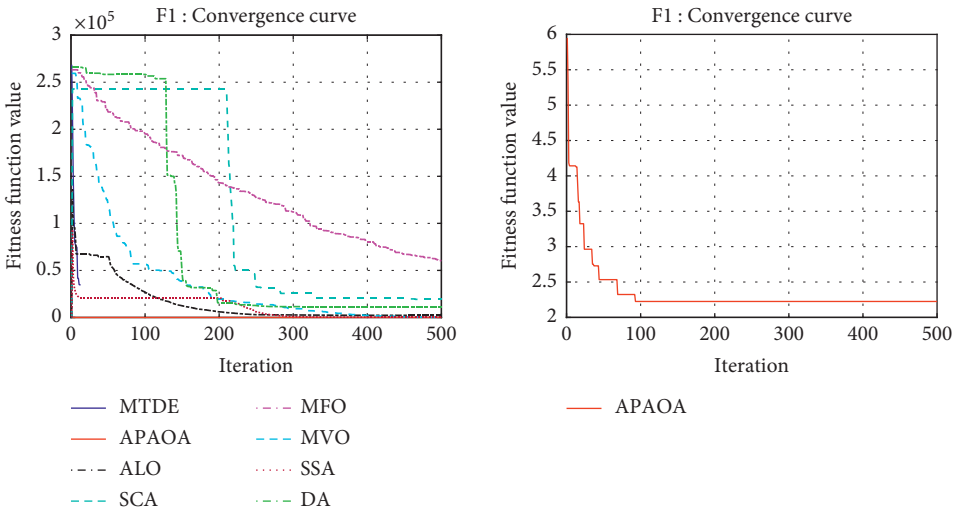


(e)

Figure 5: Comparison of convergence curves between APAOA and AOA in F14–F18.

Table 7: Parameter settings for algorithms.

Algorithm	Parameter value
ALO	None
MVO	$\mathrm{WEP}_{\mathrm{Max}} = 1$; $\mathrm{WEP}_{\mathrm{Min}} = 0.2$; $p = 6$
SCA	$a = 2$
MFO	$b = 1$, $a = [-2, -1]$ (linearly decrease)
DA	$r, \varepsilon\_\max = (\mathrm{ub} - \mathrm{lb})/10$; (ub, lb) is maximum and minimum boundary
SSA	$c_1 = [0, 2]$ (linearly reduce)
MTDE [43]	$F = 0.5$; CR = 0.8
APAOA	$\mu = 0.499$; $\alpha_{\max} = 1$; $\alpha_{\min} = 0.1$
PSO	$c_1 = c_2 = 1.5$, $w = 1$
GA	pc = 0.8, pm = 0.2
ACO	$\alpha = 1$, $\beta = 7$, Rho = 0.2, $Q = 1$



(a) (b)

Figure 6: Continued.

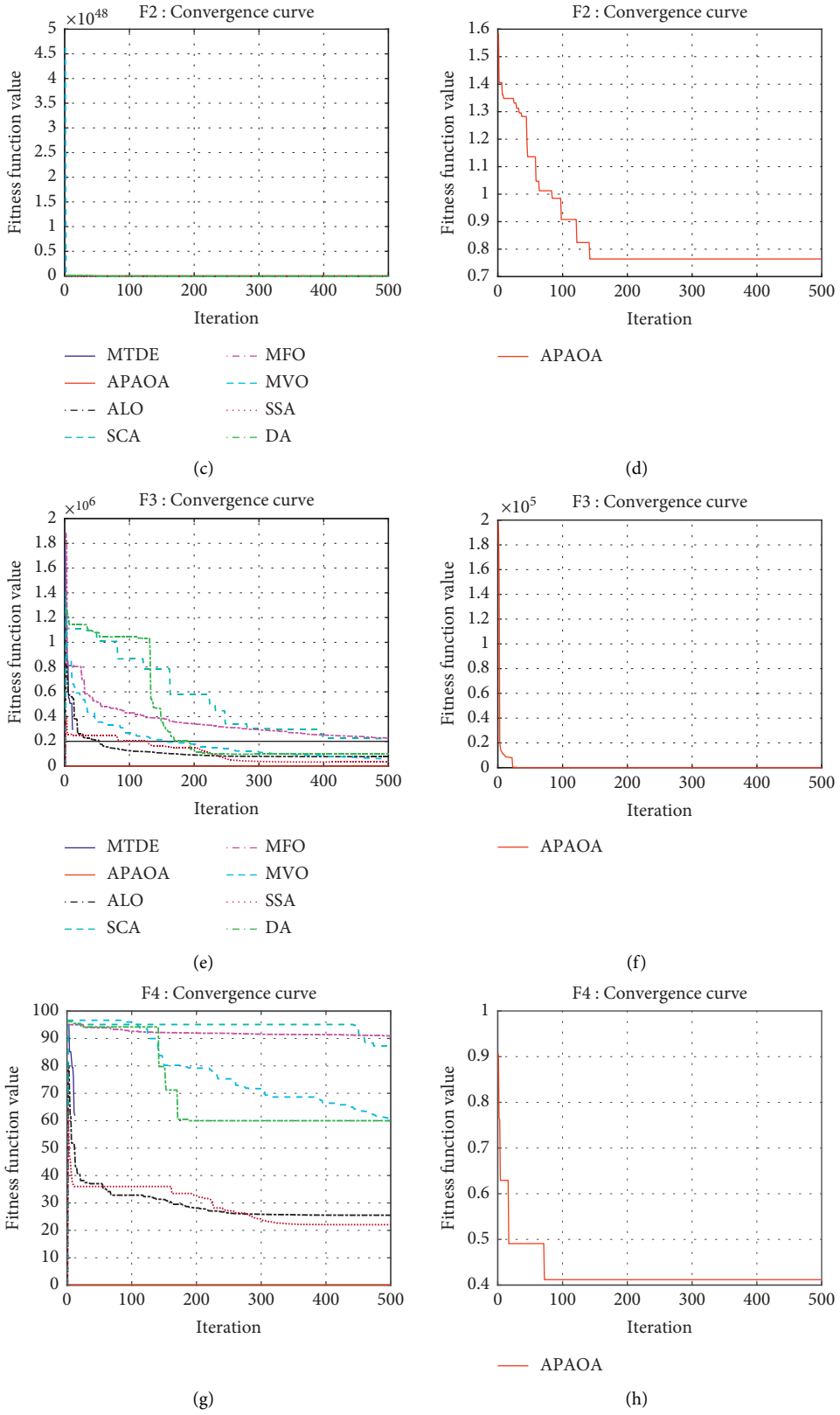


Figure 6: Continued.

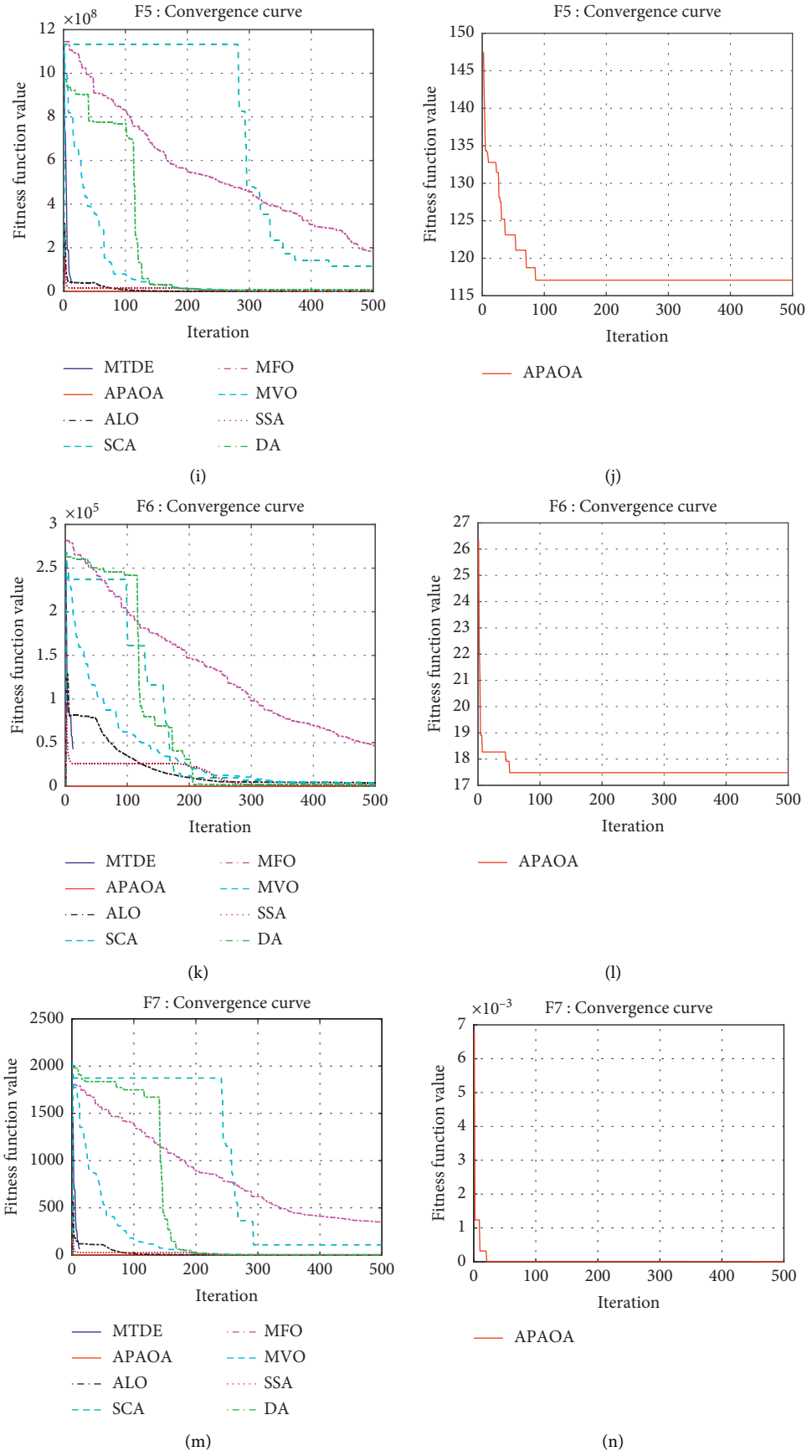


Figure 6: Continued.

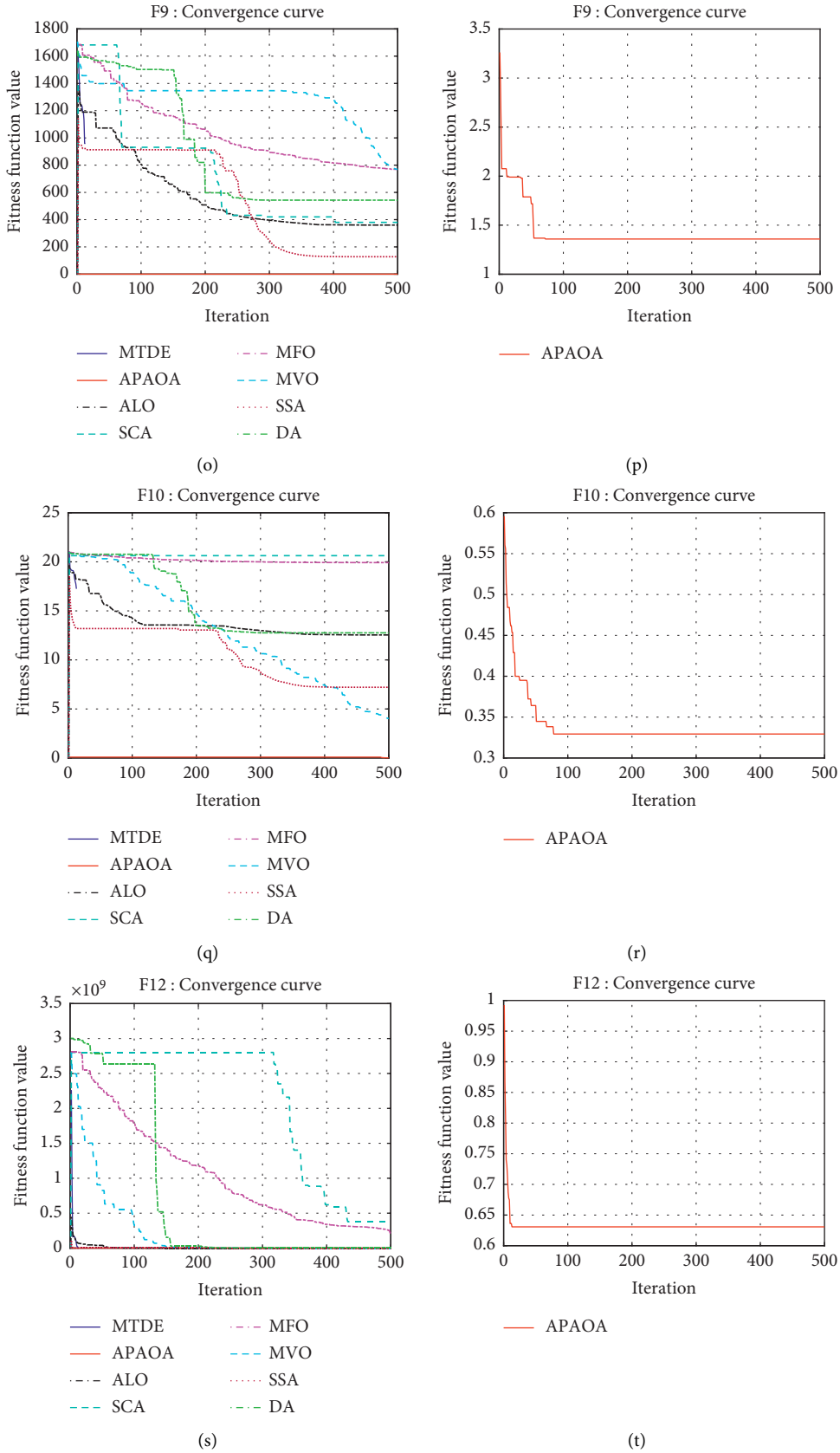


Figure 6: Continued.

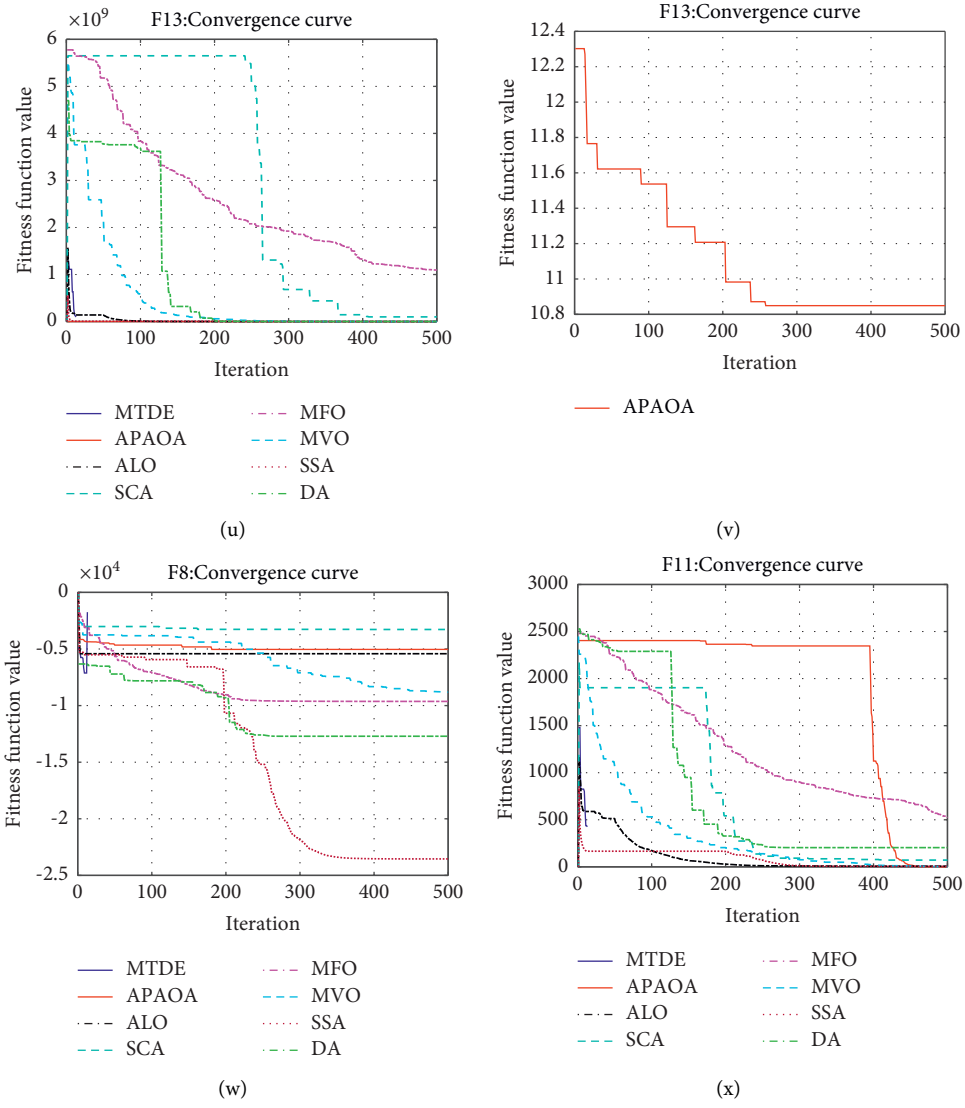


Figure 6: Comparison results of convergence performance of popular optimization algorithms.

In summary, the objective function of path planning is shown as equation (11), and the purpose is to use the penalty function to avoid barriers and achieve the shortest path, which is the optimal value.

$$\mathrm{Rp} = L * (1 + \omega * \mathrm{Penal}), \tag{11}$$

where ω is a penalty factor, which is verified by experiments that 10 is a suitable value. Therefore, it is set to 10 in this article.

5.1.3. Path Smoothing. Finally, because the connection between points is a straight line, and the path that the algorithm planned is not a feasible path, so we use Spline interpolation to smooth the solution to achieve a better accuracy.

5.2. Simulation Research. For the robot path planning mentioned in Section 5.1, we have meshed the environment and modelled the barriers. We set the two-dimensional space to be 6×6. The source coordinates and target coordinates of the two scenes are the same, which are (0, 0) and (4, 6).

To further verify the performance of the APAOA, we choose the classical algorithms, such as PSO, ACO, and GA to compare with APAOA and AOA. During the simulation, the algorithm parameters are shown in Table 7, and each algorithm runs independently 30 times. To compare the performances of the algorithms in different environments, two test environments are committed, with the number of barriers being 4 and 7. The experimental results are shown in Figures 8 and 9 and Tables 9 and 10. Tables 9 and 10 show

Table 8: Comparison results of popular algorithms in F1–F13.

F	ALO			SSA			MVO		
	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	$1.26e+03$	$6.29e+02$	**4**	$3.66e+00$	$1.69e+00$	**2**	$1.20e+02$	$2.38e+01$	**3**
F2	$2.89e+02$	$1.86e+02$	**6**	$1.14e+01$	$2.76e+00$	**3**	$5.65e+22$	$3.02e+23$	**7**
F3	$5.93e+04$	$2.17e+04$	**3**	$9.40e+03$	$3.66e+03$	**2**	$5.69e+04$	$5.91e+04$	**4**
F4	$3.13e+01$	$4.59e+00$	**3**	$1.26e+01$	$1.32e+00$	**2**	$5.46e+01$	$5.87e+00$	**5**
F5	$3.02e+05$	$2.23e+05$	**4**	$1.16e+03$	$1.14e+03$	**2**	$7.01e+03$	$7.12e+03$	**3**
F6	$1.68e+03$	$5.16e+02$	**4**	$3.22e+00$	$1.45e+00$	**1**	$1.15e+02$	$1.99e+01$	**3**
F7	$3.30e+00$	$1.02e+00$	**4**	$4.00e-01$	$8.23e-02$	**2**	$5.06e-01$	$1.05e-01$	**3**
F8	$-1.81e+04$	$3.83e-12$	**8**	$-2.43e+04$	$1.699e+03$	**2**	$-2.33e+04$	$1.32e+03$	**4**
F9	$2.98e+02$	$5.35e+01$	**4**	$9.69e+01$	$2.74e+01$	**2**	$6.77e+02$	$6.71e+01$	**5**
F10	$1.28e+01$	$1.44e+00$	**5**	$4.36e+00$	$6.58e-01$	**2**	$8.22e+00$	$6.69e+00$	**3**
F11	$1.43e+01$	$6.34e+00$	**3**	$7.76e-01$	$1.97e-01$	**1**	$2.07e+00$	$1.75e-01$	**2**
F12	$1.45e+02$	$2.70e+02$	**4**	$5.16e+00$	$1.45e+00$	**3**	$1.81e+00$	$5.79e+00$	**2**
F13	$1.63e+04$	$3.03e+04$	**4**	$1.32e+02$	$2.08e+01$	**2**	$1.47e+02$	$2.78e+01$	**3**
Mean		**4.31**			**2.00**			**3.62**	
Rank		**4**			**2**			**3**	
F	SCA			MFO			DA		
	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	$9.25e+03$	$5.25e+03$	**5**	$5.91e+04$	$1.54e+04$	**7**	$1.49e+04$	$1.07e+04$	**6**
F2	$8.01e+00$	$7.18e+00$	**2**	$2.45e+02$	$4.07e+01$	**5**	$5.36e+01$	$2.01e+01$	**4**
F3	$2.29e+05$	$4.42e+04$	**6**	$2.38e+05$	$6.24e+04$	**7**	$2.05e+05$	$7.48e+04$	**5**
F4	$8.85e+01$	$2.83e+00$	**6**	$9.25e+01$	$2.01e+00$	**7**	$5.27e+01$	$8.55e+01$	**4**
F5	$1.13e+08$	$6.02e+07$	**6**	$1.52e+08$	$7.33e+07$	**7**	$2.39e+07$	$2.10e+07$	**5**
F6	$1.22e+04$	$6.01e+03$	**5**	$5.46e+04$	$1.29e+04$	**7**	$1.55e+04$	$8.96e+03$	**6**
F7	$1.19e+02$	$6.64e+01$	**6**	$2.30e+02$	$1.30e+02$	**7**	$3.51e+01$	$3.67e+01$	**5**
F8	$-7.23e+03$	$6.41e+02$	**6**	$-2.38e+04$	$2.36e+03$	**3**	$-1.03e+04$	$9.74e+02$	**7**
F9	$2.57e+02$	$1.05e+02$	**3**	$8.28e+02$	$8.29e+01$	**7**	$7.51e+02$	$1.17e+02$	**6**
F10	$1.95e+01$	$2.86e+00$	**6**	$1.98e+01$	$1.94e-01$	**7**	$1.21e+01$	$2.47+00$	**4**
F11	$9.57e+01$	$3.69e+01$	**5**	$5.19e+02$	$1.25e+02$	**6**	$6.03e+01$	$4.28e+01$	**4**
F12	$2.56e+08$	$1.41e+08$	**7**	$2.55e+08$	$2.15e+08$	**6**	$1.39e+07$	$2.02e+07$	**5**
F13	$4.71e+08$	$2.49e+08$	**7**	$4.57e+08$	$2.20e+08$	**6**	$2.39e+07$	$1.69e+07$	**5**
Mean		**5.38**			**6.31**			**5.08**	
Rank		**6**			**7**			**5**	
F	APAOA			MTDE					
	Ave	Std	Rank	Ave	Std	Rank			
F1	$1.24e-01$	$1.00e-01$	**1**	$4.00e+05$	$0.00e+00$	**8**			
F2	$1.43e-01$	$3.02e-02$	**1**	$3.99e+62$	$3.68e+62$	**8**			
F3	$8.92e-01$	$3.69e-01$	**1**	$2.21e+08$	$0.00e+00$	**8**			
F4	$1.82e-01$	$1.47e-02$	**1**	$1.00e+02$	$0.00e+00$	**8**			
F5	$9.98e+01$	$1.19e+00$	**1**	$5.59e+10$	$2.62e+09$	**8**			
F6	$1.85e+01$	$2.74e+00$	**2**	$4.04e+05$	$0.00e+00$	**8**			
F7	$6.15e-06$	$4.27e-06$	**1**	$6.63e+09$	$8.03e+08$	**8**			
F8	$-5.91e+03$	$1.46e+03$	**1**	$-5.97e+03$	$2.18e+02$	**5**			
F9	$3.90e-02$	$4.75e-02$	**1**	$8.66e+04$	$3.88e+02$	**8**			
F10	$5.18e-02$	$2.59e-02$	**1**	$2.13e+01$	$9.05e-02$	**8**			
F11	$2.13e+03$	$5.96e+02$	**8**	$9.70e+02$	$3.81e+01$	**7**			
F12	$9.00e-01$	$2.10e-01$	**1**	$6.06e+10$	$3.63e+09$	**8**			
F13	$1.01e+01$	$2.38e-01$	**1**	$8.24e+10$	$4.28e+09$	**8**			
Mean		**1.62**			**7.69**				
Rank		**1**			**8**				

that the APAOA can find a shorter path than other algorithms, and the APAOA execution effect is more stable. Compared with AOA, the improved effect of APAOA is significant.

From the two path planning diagrams (Figures 8 and 9), it can be seen intuitively that the APAOA can always find the best path, while the other algorithms are not, and the convergence graph proves that the APAOA can find the

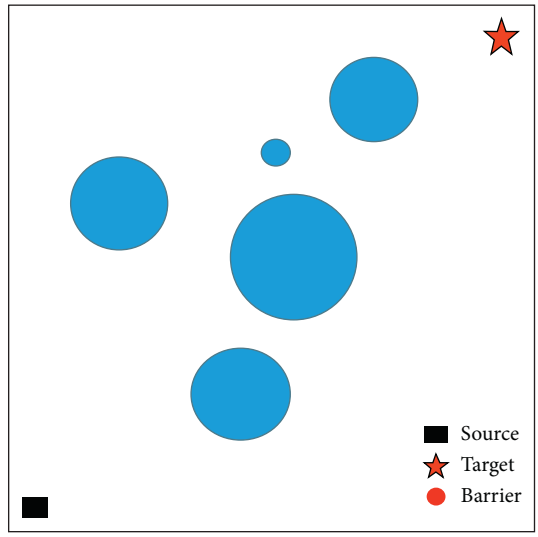


Figure 7: Robot workplace model.

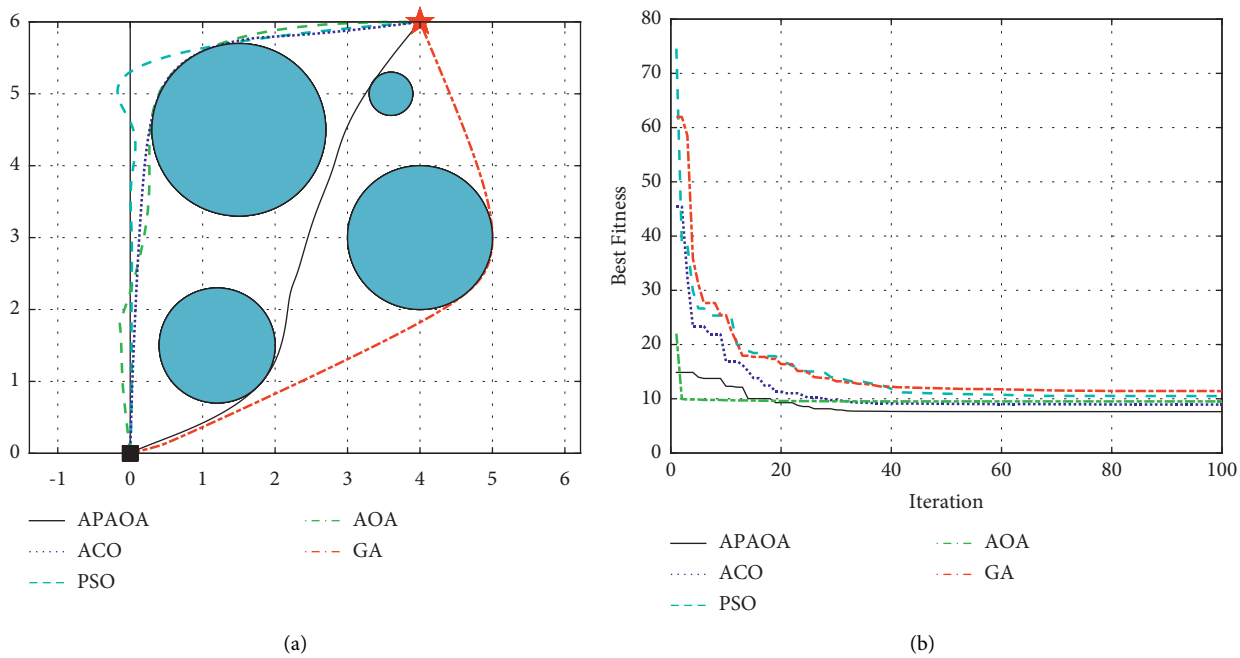


Figure 8: Path planning diagram (a) and convergence curve (b) of environment 1.

optimal path, and the convergence speed is fast. In summary, we propose the APAOA, which can better deal with the problem of robot path planning. However, there is still some problems such as insufficient accuracy and so on, which needs to be improved in the future to obtain better performance.

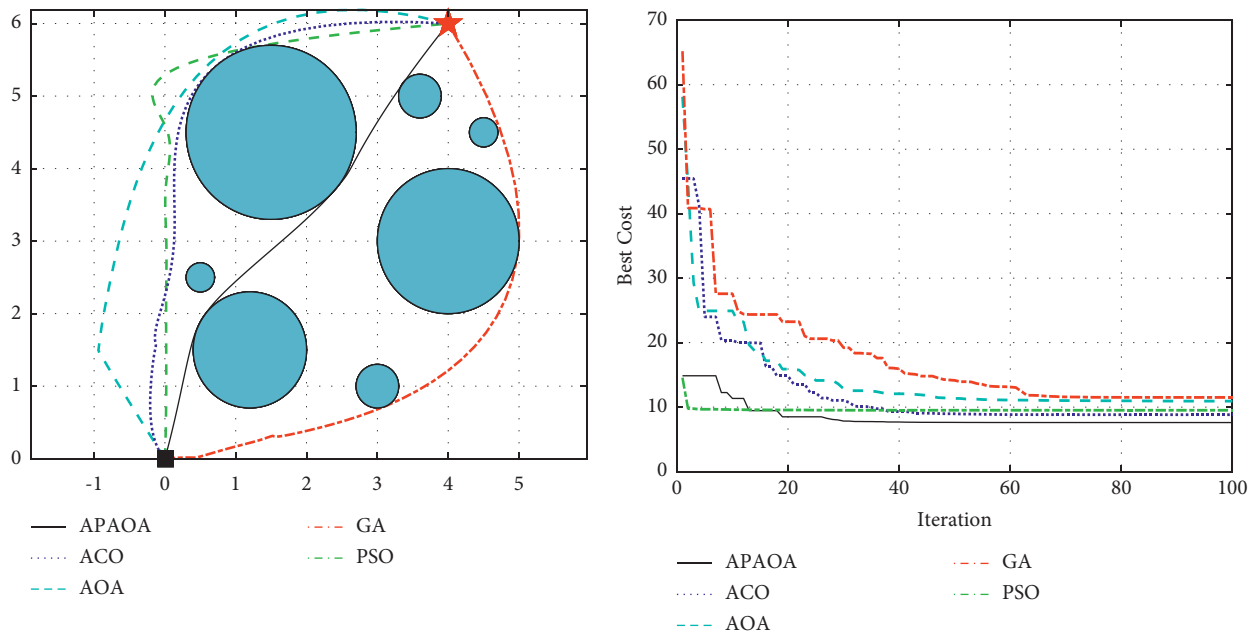


Figure 9: Path planning diagram (a) and convergence curve (b) of environment 2.

Table 9: Comparison of the APAOA with the GA, PSO, ACO, and AOA in environment 1.

Algorithm	Mean	Best
PSO	9.32	9.08
ACO	8.32	8.10
GA	9.80	9.53
AOA	9.49	9.46
APAOA	**7.55**	**7.40**

Table 10: Comparison of the APAOA with the GA, PSO, ACO, and AOA in environment 2.

Algorithm	Mean	Best
PSO	10.39	9.79
ACO	8.06	7.65
GA	11.46	9.51
AOA	11.11	9.51
APAOA	**7.64**	**7.60**

6. Conclusions

In this article, an adaptive parallel AOA is proposed and applied to solve the problem of robot path planning. First of all, the adaptive equation proposed is established based on the fitness value of the particles. The benefits of the adaptive equation can enable the algorithm to better balance the search capabilities of the development and exploration phases and effectively avoid falling into local optimal solutions. The steps for the algorithm to enter the exploration and exploitation stage are adjusted. Experiments have shown that the algorithm performs better after the adjustment, and the solution accuracy is improved. Then, we also introduce a novel parallel strategy into the AOA algorithm to strengthen the communication among particles. By adopting the rule of "survival of the fittest," leaving elite individuals, it increases the robustness of the algorithm and achieves a significant improvement compared with the original algorithm. Finally, the robot path planning problem proposed in this article is used to further evaluate the performance of the algorithm. Compared with other algorithms, APAOA achieved better performance.

Data Availability

The data used to support this study are included within the article and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] H. Sarabu, K. Ahlin, and A. Hu, "Graph-based cooperative robot path planning in agricultural environments," in *Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 519–525, Hong Kong, China, July 2019.

[2] S. Sehestedt, S. Kodagoda, and G. Dissanayake, "Robot path planning in a social context," in *Proceedings of the 2010 IEEE Conference on Robotics, Automation and Mechatronics*, pp. 206–211, Singapore, June 2010.

[3] L. Tiseni, D. Chiaradia, M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli, "UV-C mobile robots with optimized path

planning: algorithm design and on-field measurements to improve surface disinfection against SARS-CoV-2," *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 59–70, 2021.

[4] C. Xia and G. Xu, "The path planning algorithm studying about UAV attacks multiple moving targets based on voronoi diagram," *International Journal of Control & Automation*, vol. 9, no. 1, pp. 281–292, 2016.

[5] K. Zhang, P. Liu, W. Kong, Y. Lei, J. Zou, and M. Liu, "An improved heuristic algorithm for UCAV path planning," in *Proceedings of the Bio-Inspired Computing—Theories and Applications*, pp. 54–59, Xi'an, China, October 2016.

[6] G. Niu, Y. Zhang, and W. Li, "Path planning of continuum robot based on path fitting," *Journal of Control Science and Engineering*, vol. 2020, Article ID 8826749, 11 pages, 2020.

[7] J. Meng, V. M. Pawar, S. Kay, and A. Li, "UAV path planning system based on 3D informed RRT * for dynamic obstacle avoidance," in *Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1653–1659, Kuala Lumpur, Malaysia, December 2018.

[8] N. Geng, D. W. Gong, and Y. Zhang, "PSO-based robot path planning for multisurvivor rescue in limited survival time," *Mathematical Problems in Engineering*, vol. 2014, Article ID 187370, 10 pages, 2014.

[9] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[10] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[11] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[12] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2015.

[13] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[15] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[16] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," *Lecture Notes in Computer Science*, vol. 6, pp. 854–858, 2006.

[17] H. H. Inbarani, A. T. Azar, and G. Jothi, "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 175–185, 2014.

[18] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.

[19] M. Zhu, S.-C. Chu, Q. Yang, W. Li, and J.-S. Pan, "Compact sine cosine algorithm with multigroup and multistrategy for dispatching system of public transit vehicles," *Journal of Advanced Transportation*, vol. 2021, Article ID 5526127, 16 pages, 2021.

[20] T. Joyce and J. M. Herrmann, "A review of no free lunch theorems, and their implications for metaheuristic optimisation," in *Nature-Inspired Algorithms and Applied Optimization*, X.-S. Yang, Ed., Springer International Publishing, Cham, Germany, pp. 27–51, 2018.

[21] Y. Zhang, G. Guan, and X. Pu, "The robot path planning based on improved artificial fish swarm algorithm," *Mathematical Problems in Engineering*, vol. 2016, Article ID 3297585, 11 pages, 2016.

[22] Y.-Q. Qin, D.-B. Sun, L. Ning, and Y.-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 4, pp. 2473–2478, Shanghai, China, August 2004.

[23] Q. Li, W. Zhang, Y. Yin, Z. Wang, and G. Liu, "An improved genetic algorithm of optimum path planning for mobile robots," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 637–642, Jian, China, October 2006.

[24] H.-J. Wang, Y. Fu, Z.-Q. Zhao, and Y.-J. Yue, "An improved ant colony algorithm of robot path planning for obstacle avoidance," *Journal of Robotics*, vol. 2019, Article ID 6097591, 8 pages, 2019.

[25] Q. Zhang, J. Ma, and Q. Liu, "Path planning based quadtree representation for mobile robot using hybrid-simulated annealing and ant colony optimization algorithm," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 2537–2542, Beijing, China, July 2012.

[26] H. Huang and C. Tsai, "Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm," in *Proceedings of the SICE Annual Conference 2011*, pp. 1338–1343, Tokyo, Japan, September 2011.

[27] T. Dao, T. Pan, and J. Pan, "A multi-objective optimal mobile robot path planning based on whale optimization algorithm," in *Proceedings of the 2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 337–342, Chengdu, China, November 2016.

[28] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[29] Y.-P. Xu, J.-W. Tan, D.-J. Zhu, P. Ouyang, and B. Taheri, "Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm," *Energy Report*, vol. 7, no. 12, pp. 2332–2342, 2021.

[30] P. Manoharan, P. Jangir, B. S. Kumar et al., "A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations," *IEEE Access*, vol. 9, pp. 84263–84295, 2021.

[31] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8184254, 10 pages, 2020.

[32] J. Abela and D. Abramson, *A Parallel Genetic Algorithm for Solving the School Timetabling Problem*, Division of Information Technology, C.S.I.R.O, Carlton, Australia, 1991.

[33] Q.-W. Chai, S.-C. Chu, J.-S. Pan, P. Hu, and W.-M. Zheng, "A parallel WOA with two communication strategies applied in DV-Hop localization method," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 50, 2020.

[34] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao, and S.-P. Hao, "Parallel cat swarm optimization," in *Proceedings of the 2008 International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3328–3333, Kunming, China, July 2008.

[35] M. S. Nasrabadi, Y. Sharafi, and M. Tayari, "A parallel grey wolf optimizer combined with opposition based learning," in *Proceedings of the 2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 18–23, Bam, Iran, March 2016.

[36] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: a survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[37] M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: a review," *Procedia Computer Science*, vol. 133, pp. 141–152, 2018.

[38] P. Cui, W. Yan, and Y. Wang, "Reactive path planning approach for docking robots in unknown environment," *Journal of Advanced Transportation*, vol. 2017, Article ID 6716820, 11 pages, 2017.

[39] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, "A survey of parallel sequential pattern mining," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 3, p. 25, 2019.

[40] S. Pumma, M. Si, W. Feng, and P. Balaji, "Parallel I/O optimizations for scalable deep learning," in *Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 720–729, Shenzhen, China, December 2017.

[41] J. F. Roddick, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.

[42] Y. Xin, Y. Liu, and L. Guangming, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[43] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Applied Soft Computing*, vol. 97, Article ID 106761, 2020.