



RESEARCH ARTICLE

Path planning of patrol robot based on modified grey wolf optimizer

Qian Zhang¹ , Xucheng Ning¹, Yingying Li^{2,*} , Lei Pan¹, Rui Gao¹ and Liyang Zhang¹

¹School of Control and Mechanical Engineering, Tianjin Chengjian University, Tianjin, 300384, China and ²Tianjin Research Institute of Construction Machinery, Tianjin, 300409, China

*Corresponding author. E-mail: liyingscholar@163.com

Received: 19 September 2022; **Revised:** 31 January 2023; **Accepted:** 4 February 2023; **First published online:** 13 March 2023

Keywords: path planning, motion planning, mobile robots, patrol robot, grey wolf optimizer

Abstract

The grey wolf optimizer (GWO) as a new intelligent optimization algorithm has been successfully applied in many fields because of its simple structure, few adjustment parameters and easy implementation. This paper mainly aims at the defects of GWO in path planning application, such as easily falling into local optimization, poor convergence and poor accuracy, and turn point grey wolf optimization (TPGWO) algorithm is proposed. First, the idea of cross-mutation and roulette is used to increase the initial population of GWO and improve the search range. At the same time, the convergence factor function is improved to become a nonlinear update. In the early stage, the search range is expanded, and in the later stage, the convergence speed is increased, while the parameters in the convergence factor function can be adjusted according to the number of obstacles and the map area to change the turning point of the function to improve the convergence speed and accuracy of the algorithm. The turning times and turning angles of the obtained path are added to the fitness function as penalty values to improve the path accuracy. The optimization test is carried out through 16 test functions, and the test results prove the convergence and robustness of TPGWO algorithm. Finally, the TPGWO algorithm is applied to the path planning of patrol robot for simulation experiments. Compared with the GWO algorithm and Particle Swarm Optimization, the simulation results show that the TPGWO algorithm has better convergence, stability and accuracy in the path planning of patrol robot.

1. Introduction

The traditional patrol inspection work [1] is completed manually, which will not only consume a lot of human and material resources but also will result in certain dangers for patrol inspectors and concerns about inadequate inspection in dangerous environments and special weather. Therefore, the birth of intelligent inspection robot has well solved the problems of large investment of manpower, high cost, low efficiency, unreliability and certain dangers in inspection work. The operation of the inspection robot [2–4] is only related to the power and its own quality and will not be affected by dangerous environment and special weather. In addition, the flexibility of the inspection robot is not limited by time and place. Therefore, compared with manual inspection, it will greatly reduce the cost and improve the efficiency and reliability of the inspection process, when the important, complex and repeated inspection tasks are handed over to the inspection robots. The inspection robot working in a specific environment needs to design and plan its path according to the actual conditions. Path planning [5, 6] means that the inspection robot searches an optimal or suboptimal path from the starting point to the target point according to some specific constraints, such as meeting the requirements of the shortest time, the shortest distance and the least energy, so as to achieve the purposes of high efficiency, high precision and low consumption. In order to achieve these purposes, the path planning research of the inspection robot is generally divided into three steps: (1) Environment modelling; (2) Planning a path

that meets the requirements using the optimization algorithm; (3) Processing the planned path (such as smoothing the path to reduce the number of turns).

The path planning of inspection robot is mainly the selection of intelligent optimization algorithm [7, 8]. An excellent optimization algorithm plays a vital role in the path planning of inspection robot. In recent years, bionic intelligent optimization algorithm has become a hot spot to solve various optimization problems. The popular algorithms include Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony optimizer (ABC), Cuckoo Search algorithm (CS), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA). Although various optimization algorithms have made great achievements in the application of path planning, there are still some problems in the operation process, such as falling into local optimization, poor stability, slow convergence speed and poor selection accuracy. Because of these limitations, many scholars have made further research and improvement. Xiaohai Wang et al. introduced the restricted mutation generation area into the GA, which increased the search ability of the algorithm near the optimal path in the later stage, shortened the planning time and improved the feasibility of the algorithm [9]. A. Arikere et al. used multi-objective genetic algorithm to search the optimal solution of the double wishbone design problem [10]. Yubing Wang et al. proposed a distributed PSO algorithm for fast convergence, random cross-search and accurate search to avoid falling into local optimization and improve algorithm performance [11]. Yuesheng Tan et al. used PSO to train the initial parameters of the ACO, so that the ACO has higher solution quality and faster convergence speed [12]. C. Pozna et al. proposed a hybrid meta-heuristic optimization algorithm that combines particle filter and particle swarm optimization (PSO) algorithms and applied it to the optimal tuning of PI fuzzy controller [13]. Priyanka sudhakara et al. optimized the fitness function and initialization strategy of the ABC optimization algorithm, improving the performance of the algorithm [14]. Wenjie Wang et al. set the size of the step size control factor in the CS algorithm as a variable varying with the number of iterations, which accelerated the convergence speed of the algorithm and prevented falling into the local optimal solution prematurely [15]. Guangqiang Li et al. proposed an algorithm to select the best individuals from the population to create an adaptive elite set and improve its guidance strategy in view of the lack of balance in the exploration and utilization of WOA algorithm in different positions in the search space [16].

The GWO is a new bionic intelligent algorithm proposed by Mirjalili team in 2014 based on the grey wolf group predation process [17]. The GWO algorithm has the advantages of simple structure, small adjustment parameters and easy realization, and it has the convergence factor that can be adjusted adaptively. It is widely used in robot, UAV, industrial control and other industries. However, the algorithm still has some problems, such as easily falling into local optimization, slow convergence and poor accuracy. Scholars have made many improvements to these problems of GWO algorithm. Zhang Sen et al. used elite reverse learning strategy and simplex method to improve the population diversity, convergence speed and accuracy of the algorithm [18]. Shijin Li and Fucai Wang used the anti-learning model to solve the problem that the grey wolf algorithm may fall into local optimization and introduced the limit learning machine algorithm model to improve the convergence speed of the improved algorithm [19]. Jingyi Liu et al. integrated the Lion optimization algorithm and dynamic weight into the GWO optimization algorithm to improve the search ability of the algorithm and avoid falling into local optimization [20]. Qifang Luo et al. used the method of copy coding to improve the precision, stability and convergence speed of GWO [21]. AA heidari and P pahlavini combined Lévy flight and greedy selection strategy with the hunting phase of the GWO algorithm to improve population diversity and jump out of local optimization [22]. Wei Zhang et al. proposed an adaptive convergence factor adjustment strategy and an adaptive weight factor to improve the convergence accuracy, speed and stability of GWO algorithm [23]. Jing Li and Fan Yang used Kent chaos algorithm to initialize the population, enhance the diversity of the population and propose an adaptive control convergence factor to achieve the balance between exploration and mining, and combine it with particle swarm optimization algorithm to accelerate the convergence speed [24]. JS Wang and SX Li proposed an improved GWO based on evolution and elimination mechanism to achieve an appropriate compromise between exploration and development, which can further accelerate the convergence of exploration and development, and

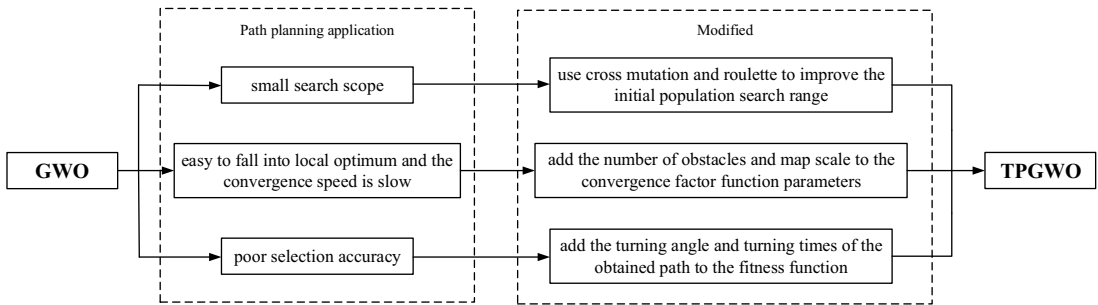


Fig. 1. Improvement of TPGWO in the application of path planning.

improve the optimization accuracy of GWO [25]. M Kohli and S Arora introduced chaos theory into GWO algorithm to accelerate its global convergence speed and improve the performance of the algorithm in engineering problems [26]. Teng Zhijun et al. proposed a hybrid GWO based on tent mapping to increase population diversity and improve global search capability [27]. Mohammad H. Nadimi shahraki et al. improved the GWO algorithm through the hunting search strategy based on dimension learning, enhanced the balance between local search and global search, and maintained diversity [28]. M. W. Guo improved the tracking mode and search mode of GWO algorithm to improve the diversity of the population and the balance between exploration and utilization [29]. Chengzhi Qu et al. proposed a new grey wolf optimization algorithm based on reinforcement learning, which can control individuals to achieve adaptive switching operation according to cumulative performance [30]. Zhang Sen et al. proposed a new meta-heuristic GWO to improve the accuracy, speed and stability of UCAV path planning [31]. Hui Xu et al. combined the GWO algorithm with the CS algorithm to improve the search mechanism and enhance the global search ability of the GWO [32].

This paper proposes a turn point grey wolf optimization (TPGWO) algorithm to solve the defects of GWO algorithm in the application of path planning, such as easily falling into local optimization, slow convergence in the later stage, poor selection accuracy and stability. It is achieved by taking the path planning of inspection robot as the research object, taking the obtained path optimization as the objective function and the environment as the constraint conditions. As shown in Fig. 1, the algorithm is improved from three aspects: population initialization, convergence factor function and fitness function, and the TPGWO algorithm is further applied to the path planning of the inspection robot. It enables the inspection robot to effectively avoid obstacles in different map environments and then select the optimal path better, faster and more stable under the premise of reaching the target point. The optimization performance of the TPGWO algorithm is verified by test functions and simulation experiments.

2. Establishment of environmental model

2.1. Problem description

In the process of path planning of inspection robot with obstacle avoidance, the environment of inspection robot should be mathematically modelled firstly, and the actual environment should be replaced by a virtual environment. Secondly, the starting point and ending point of inspection robot are given in the environment model, and an intelligent optimization algorithm is used to find a continuous curve from starting point to ending point that meets certain performance indicators and can avoid obstacles, and the curve is considered as the optimal path. This paper improves the GWO algorithm from three aspects: population initialization, convergence factor and fitness function. Combined with the actual working environment of the inspection robot, the simulation experiment is carried out finally. Therefore, before the simulation experiment, it is necessary to establish an appropriate environmental model. The TPGWO algorithm is compared with the GWO algorithm and PSO algorithm, in terms of the length of the path obtained by the three algorithms after path planning, the time consumed on planning and the convergence of the algorithm, so that the TPGWO algorithm is validated.

S	2	3	4	5	6	7	8	9	10
11	12		14	15		17	18	19	20
	22	23	24	25	26	27	28		30
31	32	33		35	36	37		39	40
41	42		44	45	46	47	48	49	50
51	52	53	54		56		58	59	60
61		63	64	65	66		68		70
71		73	74	75	76	77	78	79	80
81	82		84			87	88	89	90
91	92	93	94	95	96	97	98	99	G

Fig. 2. Path planning grid map model.

2.2. Grid map model establishment

Because the grid map [33, 34] is simple, effective and easy to implement, the grid method is used to model the environment of the mobile robot’s running space, and the size and number of grids are determined by comparing the size of obstacles and the size of the robot’s running space. Figure 2 shows the map model used in this paper. The grids in the figure are called grid nodes *S*, such as 2, 3, . . . , 99, *G*. The white grids represent the area that the robot can pass through, and the black grids represent the location of obstacles in the operating environment, where *S* is the starting point location, and *G* is the target point location. It is assumed that the boundaries of the map and obstacles are established considering the safe distance of the robot, and the robot can be regarded as a particle in the grid map. The robot is in a two-dimensional space, so the height of the robot is ignored, and multiple obstacles are distributed in the environment, and the grid method is used to establish the model. Each obstacle occupies several grids, when there is less than one grid, it is counted as a grid. In this paper, three kinds of grid maps, 10 * 10, 15 * 15 and 20 * 20, are established, respectively, to represent the actual environment with different map areas, and grid maps of each size generate different numbers of obstacles at random.

3. Grey wolf optimization algorithm

The GWO algorithm is proposed according to the hunting process of grey wolves. Grey wolf hunting is a group behaviour, and the grey wolf group has a very strict hierarchy. The whole grey wolf group is arranged like a pyramid and is divided into four hierarchy systems. As shown in Fig. 3, the leader at the first layer is called α wolf; at the second layer is the β wolf, the direct reports of α wolf; δ wolf is located on the third layer and follows the decision of α and β ; on the fourth layer are the lowest level wolves, called the ω wolves.

The basic idea of the GWO is as follows: after initializing the wolves, select the three wolves with the best fitness as the head wolves and define them as α , β and δ , and other wolves led by the head wolf, whose positions are updated according to the distance between them and their prey. The prey is then rounded up, and the position of the prey represents the optimal solution. The mathematical model of GWO is mainly established by searching prey, surrounding prey and attacking prey. The mathematical model of grey wolves surrounding its prey is as follows:

$$\vec{D} = |\vec{C} * \vec{X}_p(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} * \vec{D} \tag{2}$$

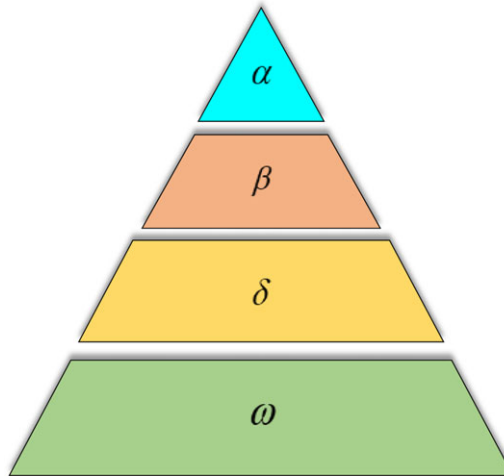


Fig. 3. Grey wolf group rating chart.

Equations (1) and (2), respectively, represent the distance between grey wolf, prey and the position update of grey wolf individuals, where t is the current iteration number, \vec{D} represents the length vector between grey wolf and prey, $\vec{X}_p(t)$ represents the position vector of the current prey, $\vec{X}(t)$ represents the position vector of the current grey wolf, $\vec{X}(t + 1)$ represents the updated position vector of grey wolf individuals, \vec{A} and \vec{C} are coefficients vector, and the calculation formula is as follows:

$$\vec{A} = 2\vec{a} * \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 * \vec{r}_2 \tag{4}$$

In Eqs. (3) and (4), \vec{r}_1 and \vec{r}_2 are random numbers between [0,1], and \vec{a} is the convergence factor. The calculation formula is as follows:

$$|\vec{a}| = 2 - \frac{2t}{t_{max}} \tag{5}$$

In Eq. (5), t_{max} is the maximum number of iterations, as the number of iterations increases, $|\vec{a}|$ decreases linearly from 2 to 0, the range of $|\vec{A}|$ also decreases, its range varies within the interval $[-a, a]$. When $|\vec{A}|$ is within the range, the next position of the grey wolf can be anywhere between its current position and the prey position. As shown in Fig. 4(a), when $|\vec{A}| < 1$, wolves attack prey, which indicates the development ability of GWO, but it is easy to fall into local optimization. As shown in Fig. 4(b), when $|\vec{A}| > 1$, grey wolves will not attack prey, and the grey wolves are forced to separate from prey to find more suitable prey, which emphasizes the exploration ability of GWO and the optimal solution can be searched globally. In Eq. (4), we can see that $|\vec{C}|$ is a random value between [0,2]. Unlike A, C has a nonlinear change, which represents the random weight of the grey wolf's location on the prey, where $|\vec{C}| > 1$ means significant influence; otherwise, the influence weight is small. The randomness of C helps GWO avoid falling into local optimization in the optimization process.

In the process of predation, the grey wolf gradually approaches and surrounds the prey and finally launches an attack on the prey. With continuous iteration, all initial solutions keep approaching the optimal solution, and finally, the optimal solution is obtained; that is, the optimal solution is the head wolf α , the second and third solutions are β wolf and δ wolf, respectively. The location update of grey wolf individuals in the GWO is shown in Fig. 5, where \vec{a}_1, \vec{a}_2 and \vec{a}_3 means the convergence factor between grey wolf ω and the head wolves α, β and δ , respectively, and \vec{C}_1, \vec{C}_2 and \vec{C}_3 are the coefficients vector between them, and $\vec{D}_\alpha, \vec{D}_\beta$ and \vec{D}_δ represent the distance vectors between them.

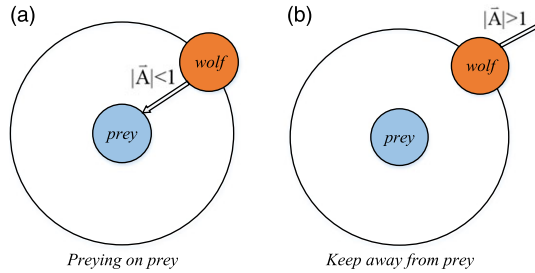


Fig. 4. Schematic diagram of grey wolf preying on and away from prey. (a) Preying on prey. (b) Keep away from prey.

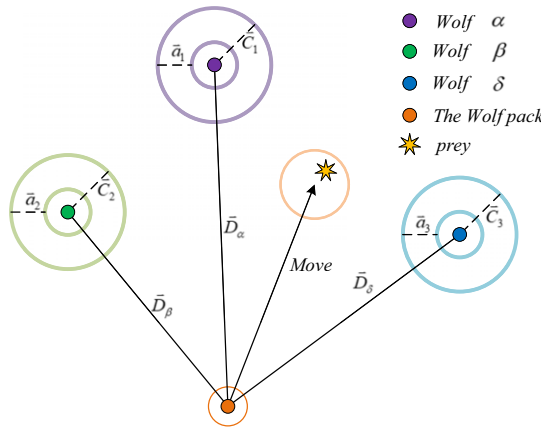


Fig. 5. GWO grey wolf position update graph.

The mathematical model of attacking prey in GWO is as follows:

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 * \vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta = |\vec{C}_2 * \vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta = |\vec{C}_3 * \vec{X}_\delta - \vec{X}| \end{cases} \tag{6}$$

In Eq. (6), \vec{X}_α , \vec{X}_β and \vec{X}_δ represent the current position vector of α , β and δ , respectively; \vec{X} indicates the position vector of grey wolf ω , and the update process is shown in Eqs. (7) and (8).

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 * (\vec{D}_\alpha) \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 * (\vec{D}_\beta) \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 * (\vec{D}_\delta) \end{cases} \tag{7}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{8}$$

Equations (7) and (8) show location update of grey wolf ω and the final position of grey wolf ω , respectively, where \vec{X}_1 , \vec{X}_2 , \vec{X}_3 represent the updated position vector between grey wolf ω and the head wolf α , β and δ , respectively; \vec{A}_1 , \vec{A}_2 , \vec{A}_3 refer to coefficients vector between grey wolf ω and the head wolf α , β and δ ; $\vec{X}(t+1)$ represents the vector sum of \vec{X}_1 , \vec{X}_2 and \vec{X}_3 , the final updated location of grey wolf ω .

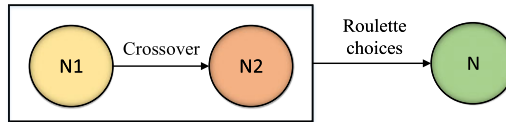


Fig. 6. Schematic diagram of population initialization improvement.

The optimization process of GWO algorithm starts from the random initialization of the population. In the iterative process, the position is updated according to the distance between the three wolves with the best fitness α , β and δ and the prey. If the range of the random variable $|\vec{A}| < 1$, it determines that the grey wolf is approaching the prey; $|\vec{A}| > 1$ means that the grey wolf is forced to stay away from the prey to find a more suitable prey, and the best prey will be found in the last iteration. The basic calculation steps of GWO are as follows:

1. Initialize the population number N of wolves, the convergence factor \vec{a} and the vector coefficients \vec{A} and \vec{C} .
2. Calculate the fitness of each grey wolf and save the three wolves with the best fitness as α , β and δ .
3. Update the position of grey wolf and parameters \vec{a} , \vec{A} , \vec{C} according to the formula.
4. Calculate the individual fitness of grey wolf and update the fitness and position of three head wolves.
5. Judge whether the maximum number of iterations has been reached and output the position of the head wolf α as the optimal solution, otherwise return to step 3 to continue the calculation.

4. TPGWO

4.1. Initialization improvements

The initialization population of the GWO algorithm is randomly generated, which has the disadvantages of small population and easily falling into local optimization. In order to improve the diversity of the initialization population of the GWO, the population initialization of GWO is improved by using the idea of population crossover and roulette. In the application of TPGWO algorithm in path planning of patrol robot, its initialization population represents each path in the algorithm. The better the fitness of grey wolf individual, the better the superiority of the path is. According to the environmental model in Section 2, the initialization population of GWO in path planning application is the path nodes randomly generated in the grid map. Each path is connected by nodes to form a grey wolf individual. Therefore, the numbers of nodes of each path are the same under the same map. As shown in Fig. 6, the initialized population $N1$ is crossed to obtain population $N2$ in TPGWO algorithm, and then, the roulette idea is used to calculate the best fitness in populations $N1$ and $N2$ as the initial population N for iterative updating, which can not only improve the search range of the initial population but also improve the quality of the initial population.

4.2. Improvement of convergence factor

The convergence factor a of the GWO algorithm is a linear decreasing function whose value is from 2 to 0. Its updating mechanism is that the convergence factor searches in the range of $[1, 2]$ and converges in the range of $[1, 0]$. Therefore, it is easy to fall into local optimization and the convergence speed is slow. Therefore, the arctangent function and logarithmic function are used to improve the decline curve of the convergence factor a , so that the search range of the convergence factor in the early stage is wider and the convergence speed in the later stage is faster. On this basis, the ratio of the number of obstacles to the

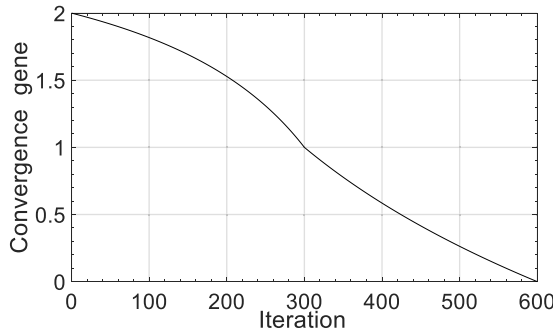


Fig. 7. Variation curve of convergence factor of logarithmic function with the base of 1/2.

map area is applied to it, and the parameter value in the function model can be adjusted appropriately to change the turning point of the convergence factor to achieve the optimal selection. The convergence factor function model of the TPGWO algorithm is as follows:

$$k = \frac{\left[\tan \left(\frac{1-8 \arctan 10}{8} \right) + 10 \right]}{(p * T)} \tag{9}$$

$$a = \begin{cases} -8 \arctan (k * t - 10) - 8 \arctan (10) + 2 & t \leq p * T \\ \log_p^{(t/600)} & t > p * T \end{cases} \tag{10}$$

In Equations (9) and (10), k is the adjustment parameter, p is the base of the logarithmic function, T is the maximum number of iterations, and t is the current number of iterations. Some examples are as below:

- ① Taking $T = 600$ as an example, when the logarithmic function is based on $1/2$,

$$k = \frac{\left[\tan \left(\frac{1-8 \arctan 10}{8} \right) + 10 \right]}{300} \tag{11}$$

$$a = \begin{cases} -8 \arctan (k * t - 10) - 8 \arctan (10) + 2 & t \leq 300 \\ \log_{0.5}^{(t/600)} & t > 300 \end{cases} \tag{12}$$

Equation (12) is the improved convergence factor function when the logarithmic function takes $1/2$ as the base, and Fig. 7 shows the change curve. When the logarithmic function takes $1/2$ as the base, the denominator of the adjustment parameter k in Eq. (11) corresponds to 300.

- ② Taking $T = 600$ as an example, when the logarithmic function is based on $1/3$,

$$k = \frac{\left[\tan \left(\frac{1-8 \arctan 10}{8} \right) + 10 \right]}{200} \tag{13}$$

$$a = \begin{cases} -8 \arctan (k * t - 10) - 8 \arctan (10) + 2 & t \leq 200 \\ \log_{1/3}^{(t/600)} & t > 200 \end{cases} \tag{14}$$

Equation (14) is the improved convergence factor function when the logarithmic function takes $1/3$ as the base, and Fig. 8 shows the change curve. When the logarithmic function takes $1/3$ as the base, the denominator of the adjustment parameter k in Eq. (13) corresponds to 200.

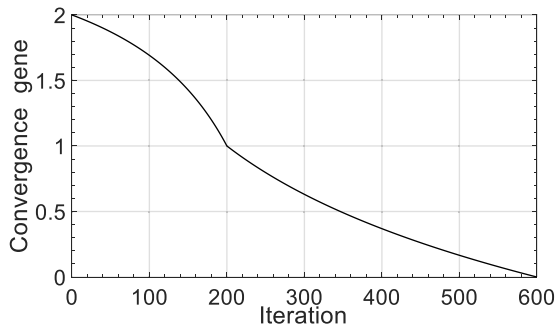


Fig. 8. Variation curve of convergence factor of logarithmic function with the base of 1/3.

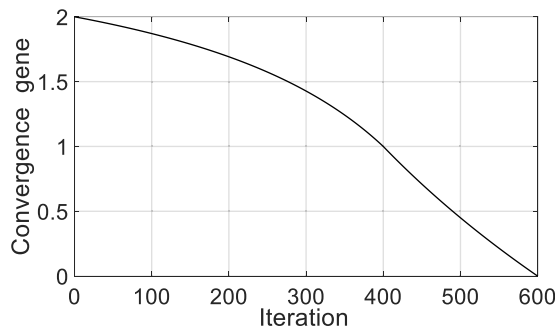


Fig. 9. Variation curve of convergence factor of logarithmic function with base 2/3.

③ Taking $T = 600$ as an example, when the logarithmic function is based on $2/3$,

$$k = \frac{\left[\tan \left(\frac{1 - 8 \arctan 10}{8} \right) + 10 \right]}{400} \tag{15}$$

$$a = \begin{cases} -8 \arctan (k * t - 10) - 8 \arctan (10) + 2 & t \leq 400 \\ \log_{2/3}^{(t/600)} & t > 400 \end{cases} \tag{16}$$

Equation (16) is the improved convergence factor function when the logarithmic function takes $2/3$ as the base, and Fig. 9 shows the change curve. When the logarithmic function takes $2/3$ as the base, the denominator of the adjustment parameter k in Eq. (15) corresponds to 400.

From the examples above we can see that, in the research of path planning, it is difficult to find the optimal path in the complex environment map with a large number of obstacles, while it is simple to find the optimal path in the simple environment map with a small number of obstacles. Therefore, in the complex environment map with a large number of obstacles, the search range and search time of the optimization algorithm can be appropriately increased to find the optimal path. On the contrary, in a simple environment map with a small number of obstacles, the search range and search time of the optimization algorithm can be appropriately reduced to find the optimal path. Therefore, the ratio of the number of obstacles to the map area can be added to the function model through the above laws. When the number of obstacles is large and complex, the turning point can be delayed, the iteration times during search can be increased, and the iteration times of convergence can be reduced. When the number of obstacles is small and simple, the turning point can be advanced to reduce the number of iterations during search and increase the number of iterations during convergence.

As a variable value, the base p of the logarithmic function can only be used as a fine adjustment in the application of the ratio of the number of obstacles to the map area. In the Eqs. (9) and (10),

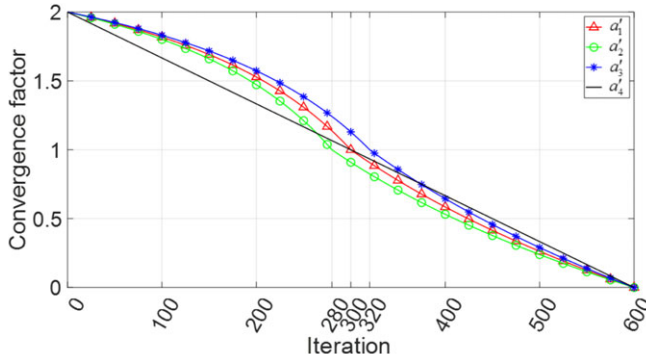


Fig. 10. Variation curves of functions with different convergence factors.

taking the maximum number of iterations of 600 as an example, when the number of obstacles remains unchanged, let $p = \frac{1}{2}$, the turning point remains unchanged, as shown in a_1' in Eq. (18). When the number of obstacles is relatively small, let $p < \frac{1}{2}$, turning point moves forward, as shown in a_2' in Eq. (20), and let p be 280/600 to reduce the search time of the algorithm and improve the convergence speed. When there are relatively more and complex obstacles, let $p > \frac{1}{2}$, turning point moves backwards, as shown in a_3' in Eq. (22), and let p be 320/600 to increase the search time of the algorithm, so that to avoid falling into local optimization and improve the accuracy of the selected path. Equation (23) is the convergence factor function a_4' of the GWO algorithm. Figure 10 shows the comparison of the convergence factor function curves a_1' , a_2' , a_3' and a_4' .

$$k_1 = \frac{[\tan(\frac{1-8\arctan 10}{8}) + 10]}{300} \tag{17}$$

$$a_1' = \begin{cases} -8 \arctan(k_1 * t - 10) - 8 \arctan(10) + 2 & t \leq 300 \\ \log_{1/2}^{(t/600)} & t > 300 \end{cases} \tag{18}$$

$$k_2 = \frac{[\tan(\frac{1-8\arctan 10}{8}) + 10]}{280} \tag{19}$$

$$a_2' = \begin{cases} -8 \arctan(k_2 * t - 10) - 8 \arctan(10) + 2 & t \leq 280 \\ \log_{280/600}^{(t/600)} & t > 280 \end{cases} \tag{20}$$

$$k_3 = \frac{[\tan(\frac{1-8\arctan 10}{8}) + 10]}{320} \tag{21}$$

$$a_3' = \begin{cases} -8 \arctan(k_3 * t - 10) - 8 \arctan(10) + 2 & t \leq 320 \\ \log_{320/600}^{(t/600)} & t > 320 \end{cases} \tag{22}$$

$$a_4' = 2 - \frac{t}{300} \tag{23}$$

4.3. Improvement of fitness function

In the application of TPGWO in path planning, its fitness function is the sum of the distances between each node of the path calculated by Euclidean distance. At the same time, the turning times and turning angles of the obtained path are added to the fitness function as penalty values to improve the accuracy of the selected path. In the grid map that built, (x, y) is the coordinates of the current node; (x_1, y_1) is the coordinates of the previous node; (x_2, y_2) is the coordinates of the next node; V is the number of turns;

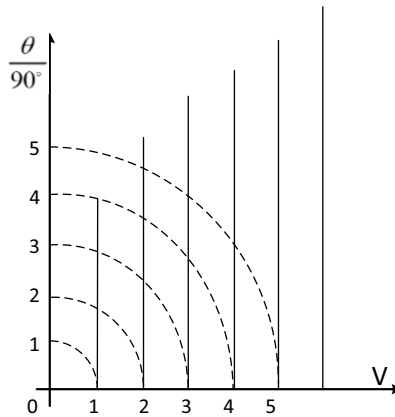


Fig. 11. Normalization of turning times and turning angles.

θ is the turning angle; r and c are the number of rows and columns of the map, respectively. The fitness function is as follows:

$$L = L(i) + \text{fix}(M) \tag{24}$$

$$L(i) = \sqrt{(y - y_1)^2 + (x - x_1)^2} \tag{25}$$

$$\tan \theta_1 = \frac{y - y_1}{x - x_1} \Rightarrow \theta_1 = \arctan \frac{y - y_1}{x - x_1} \tag{26}$$

$$\tan \theta_2 = \frac{y_2 - y}{x_2 - x} \Rightarrow \theta_2 = \arctan \frac{y_2 - y}{x_2 - x}$$

$$\theta_i = |\theta_2 - \theta_1| \tag{27}$$

In Eq. (24), $L(i)$ is the sum of all nodes in the path, and $\text{fix}()$ is a rounding function to the left. Equation (25) is the calculation expression of $L(i)$. In Eq. (26), θ_1 and θ_2 are the tangent angles between the current node and the previous node, and between the current node and the next node, respectively. In Eq. (27), θ_i is the angle difference between the two tangent angles, that is, the angle generated by the current turn. In path planning, if the angle of the obtained path changes, the angle difference of the tangent angle between nodes will occur. Therefore, in the calculation of fitness function, when $\tan \theta_1 \neq \tan \theta_2$, it means that there is a turn in the path, and then $V = V + 1$. Every turn will generate a turning angle. In order to facilitate the addition of turning times and angles to the fitness function, the turning times and angles are normalized.

The normalization processing method is shown in Fig. 11, where V is the number of turns ($V = 1, 2, \dots$), θ is the sum of turning angles, and the turning angle generated each time is between 0° and 360° . When the number of turns is 1, $\theta/90^\circ$ is 0 to 4; when the number of turns is 2, $\theta/90^\circ$ is 0 to 8; when the number of turns is 3, $\theta/90^\circ$ is 0 to 12; when the number of turns is n , $\theta/90^\circ$ is 0 to $4n$. The turning times and turning angles are normalized by the radius M of the arc. The indication shown in Fig. 10 is as follows: when $0 \leq M < 1, f = 0$; when $1 \leq M < 2, f = 1$; when $2 \leq M < 3, f = 2$; when $3 \leq M < 4, f = 3$; \dots ; when $n - 1 \leq M < n, f = n - 1$, where M is the corresponding arc radius and f is the rounding function to the left. Therefore, the penalty values of turning times and turning angles can be determined by calculating the arc radius M , and the arc radius can be determined by the horizontal and vertical coordinates in Fig. 11.

$$\theta = \sum_{i=1}^V \theta_i \tag{28}$$

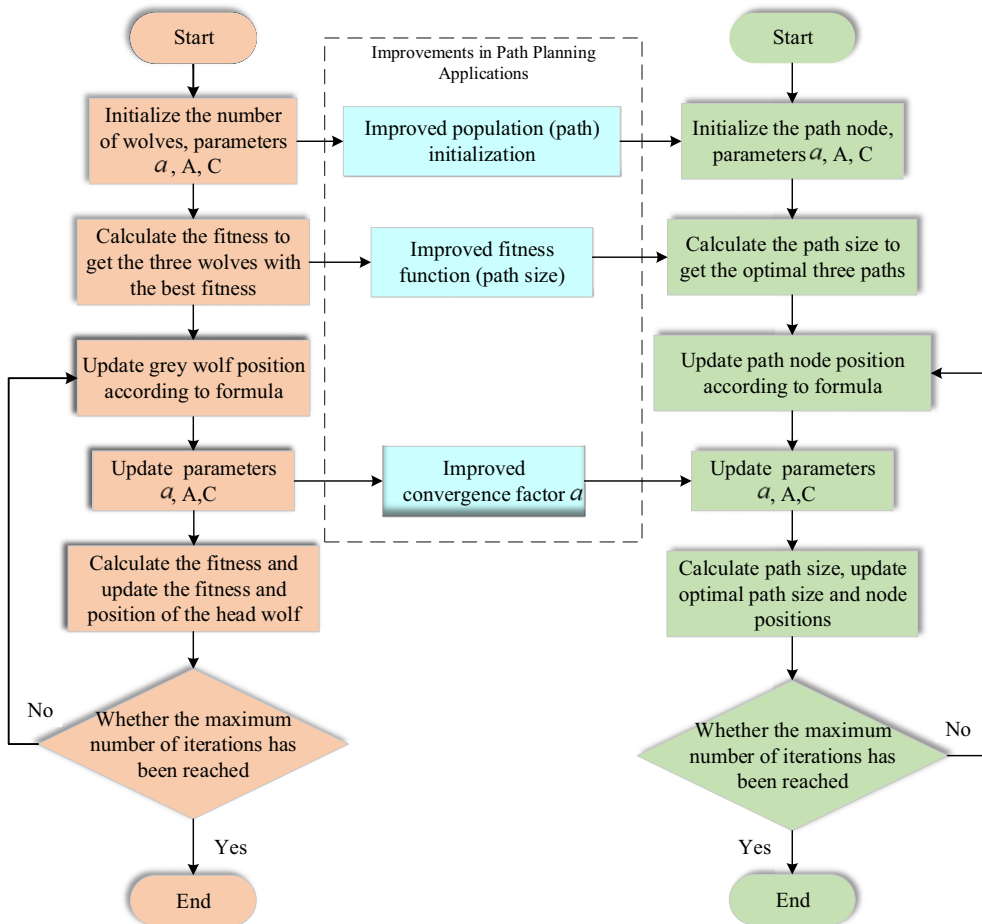


Fig. 12. Improved flow chart of TPGWO algorithm.

$$A = \frac{\sum_{i=1}^v \theta_i}{90} = \frac{\theta}{90} \tag{29}$$

$$M = \sqrt{V^2 + A^2} \tag{30}$$

Equation (28) is the sum of the turning angles. Equation (29) is to calculate the angle degrees, which divide the sum of the turning angles by 90 to obtain the angle judgment value A, which is the vertical coordinate in Fig. 11. Its horizontal coordinate is the turning times V, and then, (V, A) is the coordinate value in Fig. 11. Equation (30) is the size M of the arc radius calculated from the coordinate value, and the penalty value fix(M) can be obtained from the arc radius.

4.4. Summary

This section describes the improvement of TPGWO algorithm in path planning. TPGWO algorithm improves the population initialization of GWO algorithm through the idea of population crossover and roulette to improve the population diversity and changes the linear convergence factor function in GWO algorithm to a nonlinear function to improve the early search range and late convergence speed. Finally, the turning times and turning angles are added to the fitness function to improve the accuracy of the path. The algorithm improvement process is shown in Fig. 12.

5. Simulation results

In the application of TPGWO, each wolf indicates a path from the start point to the destination in the map. Therefore, the best path can be selected using TPGWO theory proposed in the sections above. In order to verify the superiority of TPGWO in the application of path planning, the convergence of TPGWO is analysed firstly. Secondly, test functions are used to analyse the performance of TPGWO, PSO and GWO. Finally, grid maps with different sizes and obstacles are established, and the planning time and size of the paths obtained by the three algorithms under different maps are compared and analysed. At the same time, the convergence factor function is adjusted in TPGWO according to the number of obstacles, and the planning time and length of the path obtained are compared and analysed.

5.1. Convergence analysis

Theorem 1: *When the parameter a in the algorithm gradually decreases from 2 to 0 with the increase of iteration times, the GWO algorithm has convergence.*

Proof: It can be seen from the updated formula of GWO

$$\begin{aligned}
 X(t+1) &= \frac{1}{3} [X_1(t+1) + X_2(t+1) + X_3(t+1)] \\
 &= \frac{1}{3} [X_\alpha - A_1 * |C_1 * X_\alpha - X(t)| + X_\beta - A_2 * |C_2 * X_\beta - X(t)| + X_\delta - A_3 * |C_3 * X_\delta - X(t)|] \\
 &= \frac{1}{3} [X_\alpha - (2a_t r_{11} - a_t) |2r_{12} X_\alpha - X(t)| + X_\beta - (2a_t r_{21} - a_t) |2r_{22} X_\beta - X(t)| \\
 &\quad + X_\delta - (2a_t r_{31} - a_t) |2r_{32} X_\delta - X(t)|] \\
 &= \frac{1}{3} (X_\alpha + X_\beta + X_\delta) - \frac{1}{3} a_t [(2r_{11} - 1) |2r_{12} X_\alpha - X(t)| + (2r_{21} - 1) |2r_{22} X_\beta - X(t)| \\
 &\quad + (2r_{31} - 1) |2r_{32} X_\delta - X(t)|] \tag{31}
 \end{aligned}$$

In Eq. (31), $X(t)$ is the current position of grey wolves, $X(t+1)$ is the next position of grey wolves, $X_\alpha, X_\beta, X_\delta$ is the current iteration values of wolf α, β, δ , respectively, a_t is the value of convergence factor a of the current iteration, $r_{11}, r_{12}, r_{21}, r_{22}, r_{31}, r_{32}$ are random values in $[0, 1]$.

Let

$$r_1 = 2r_{11} - 1, r_2 = 2r_{21} - 1, r_3 = 2r_{31} - 1, r_1, r_2, r_3 \in [-1, 1] \tag{32}$$

then

$$X(t+1) = \frac{1}{3} (X_\alpha + X_\beta + X_\delta) - \frac{1}{3} a_t [2r_1 r_{12} X_\alpha - r_1 X(t) + 2r_2 r_{22} X_\beta - r_2 X(t) + 2r_3 r_{32} X_\delta + r_3 X(t)] \tag{33}$$

Let

$$r'_1 = 2r_1 r_{12}, r'_2 = 2r_2 r_{22}, r'_3 = 2r_3 r_{32}, r'_1, r'_2, r'_3 \in [-2, 2] \tag{34}$$

then

$$X(t+1) = \frac{1}{3} (X_\alpha + X_\beta + X_\delta) - \frac{1}{3} a_t (r'_1 X_\alpha + r'_2 X_\beta + r'_3 X_\delta) + \frac{1}{3} a_t (r_1 + r_2 + r_3) X(t) \tag{35}$$

In GWO, a_t is a linear decreasing function from 2 to 0, that is, $\lim_{t \rightarrow t_{max}} a_t = 0$.

Therefore, when $X_\alpha, X_\beta, X_\delta$ is constant,

$$\lim_{t \rightarrow t_{max}} X(t+1) = \frac{1}{3} (X_\alpha + X_\beta + X_\delta) \tag{36}$$

t_{max} is the maximum number of iterations. Proof completed.

Table I. Parameter settings of three algorithms.

Main parameter settings	
PSO	Particle number $n = 30$, learning factor $c1, c2 = 2$, Inertia weight $w = 0.9$
GWO	Wolves number $n = 30$, Convergence factor $a = a'_4$
TPGWO	Wolves number $n = 30$, Convergence factor $a = a'_1$

Definition 1: The grey wolf state is composed of the grey wolf position vector, remember that the grey wolf position vector is X , and Y is the feasible region space of the problem, $X \in Y$, and then, the status of grey wolves is recorded as $\mu = (X_1, X_2, \dots, X_n)$, X_i indicates the status of the i th grey wolf.

It can be obtained from reference [35], the state sequence $\{\mu(t) : t > 0\}$ of grey wolf population in GWO is a finite Markov chain, and when $t_{\max} \rightarrow +\infty$, the grey wolf group state has ergodicity in the finite state space Y . Therefore, the corresponding state $X_\alpha, X_\beta, X_\sigma$ of the three leading wolves α, β, δ in the algorithm can be guaranteed to be the global optimal state, the suboptimal state and the third optimal state, respectively. According to the above formula, when $t_{\max} \rightarrow +\infty$, the GWO algorithm has global convergence.

To sum up, the GWO algorithm has convergence, and convergence factor \vec{a} has a large impact on convergence. Therefore, this paper mainly changes the convergence factor function from linear decline to nonlinear decline on the basis of GWO and retains the update strategy of the grey wolf algorithm, so it improves the convergence speed of the algorithm in the later period and enhances convergence.

5.2 Algorithm test

In order to test the optimization performance of PSO algorithm, GWO algorithm and the proposed TPGWO algorithm, 16 test functions are used to study and compare, through the statistics of the results of different performance of the three algorithms. Their optimization performance is compared and analysed to verify the effectiveness and feasibility of TPGWO algorithm. In the comparative experiment, in order to compare the fairness, the same experimental parameters and initialization mode are used in the three algorithms. The three algorithms use the same initialization method to get the same starting point. The main parameters of the three algorithms are shown in Table I, and the maximum iteration t_{\max} is 600. The test function and its dimension, truth value and function range are shown in Table II. In order to prevent the influence of randomness on the results, each algorithm is run 30 times to take the mean value, and the optimal value, mean value and standard deviation of the algorithm are recorded, respectively. The optimal value reflects the quality of the algorithm, and the mean value shows the accuracy that can be achieved under a given number of iterations; that is, it reflects the convergence speed of the algorithm, and the standard deviation reflects the stability and robustness of the algorithm.

Three groups of 16 test functions with different characteristics [36, 38] are used to test the performance of three algorithms: unimodal function f_1-f_7 , multimodal function f_8-f_{13}, f_{15} and fixed multimodal function f_{14}, f_{16} . The 3D graphs of the test function and the convergence curves of the corresponding three algorithms are shown in Fig. 13. According to the test results in Table III, TPGWO algorithm can be used to test function $f_1-f_7, f_9-f_{13}, f_{15}-f_{16}$. Although no true value is obtained, its optimal value and mean value are closer to the true value than PSO algorithm and GWO algorithm, and the standard deviation is small. In test function f_8 , it is slightly worse than PSO algorithm and better than GWO algorithm; the optimal values of the three optimization algorithms in testing function f_{14} are the same, but the mean and standard deviation of TPGWO algorithm are worse than PSO algorithm and better than GWO algorithm. From the convergence curve of each test function, it can be seen that the convergence accuracy and convergence stability of TPGWO algorithm are greatly improved compared with the other two algorithms, which proves that TPGWO algorithm has good solution accuracy, stability and robustness.

Table II. Test function and its dimension, range and true value.

Function	Dim	Range	f_{\min}
$f_1 \quad x = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$f_2 \quad x = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$f_3 \quad x = \sum_{i=1}^n \sum_{j=1}^i x_j^2$	30	[-100,100]	0
$f_4 \quad x = \max_j x_j , 1 \leq i \leq n$	30	[-100,100]	0
$f_5 \quad x = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$f_6 \quad x = \sum_{i=1}^n (x_i + 0.5)^2$	30	[-100,100]	0
$f_7 \quad x = \sum_{i=1}^n ix_i^4 + \text{random}(0, 1)$	30	[-1.28, 1.28]	0
$f_8 \quad x = \sum_{i=1}^n -x_i^2 \sin \sqrt{ x_i }$	30	[-500, 500]	-418.9829*Dim
$f_9 \quad x = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$f_{10} \quad x = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30,	[-32,32]	0
$f_{11} \quad x = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$f_{12} \quad x = \frac{\pi}{n} 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	[-50, 50]	0
$y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$f_{13} \quad x = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i) + 1] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50, 50]	0
$f_{14} \quad x = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right)$	2	[-65, 65]	1
$f_{15} \quad x = \sum_{i=1}^D x_i^2 + \sum_{i=1}^D (0.5x_i)^2 + \sum_{i=1}^D (0.5x_i)^4$	30	[-5, 10]	0
$f_{16} \quad x = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

Table III. Optimization results of three algorithms in test function.

Function	f_{\min}	Dim	Reference	PSO	GWO	TPGWO
f_1	0	30	Best	0.0072189	2.3383e-41	2.0765e-42
			Ave	0.0229	7.4611e-41	8.8323e-42
			Std	0.0162	7.0023e-41	9.1656e-42
f_2	0	30	Best	0.0089475	5.3554e-25	2.5246e-25
			Ave	0.0138	4.4838e-24	1.5158e-24
			Std	0.0077	5.0381e-24	2.2670e-24
f_3	0	30	Best	1145.0194	2.7747e-12	1.0625e-13
			Ave	1.7047e+03	3.0270e-11	1.2216e-12
			Std	455.7244	2.5969e-11	1.0388e-12
f_4	0	30	Best	4.6133	1.3305e-10	5.1614e-11
			Ave	6.9703	3.90795e-10	5.77355e-11
			Std	2.357	2.57745e-10	601215e-12
f_5	0	30	Best	72.6214	26.1377	25.3457
			Ave	278.1537	27.2483	26.2195
			Std	220.4144	0.7431	0.6478
f_6	0	30	Best	0.0028525	1.4377e-05	3.9615e-06
			Ave	0.0065	0.1977	8.6778e-06
			Std	0.0036	0.2041	3.6505e-06
f_7	0	30	Best	0.032609	0.0007933	0.00057399
			Ave	0.0403	0.0012	7.4287e-04
			Std	0.0088	2.5790e-04	1.6822e-04
f_8	-12569.487	30	Best	-9014.6267	-5942.7052	-6781.8466
			Ave	-8.5802e+03	-5.4094e+03	-6.3349e+03
			Std	385.9680	1.0779e+03	415.6486
f_9	0	30	Best	33.1906	0	0
			Ave	46.1688	0.8903	1.1369e-14
			Std	13.5863	1.9908	2.5421e-14
f_{10}	0	30	Best	0.040004	2.931e-14	1.5099e-14
			Ave	0.1922	2.7178e-14	2.3625e-14
			Std	0.2999	6.9256e-15	3.1779e-15
f_{11}	0	30	Best	0.01114	0	0
			Ave	0.0774	0.0028	0
			Std	0.0853	0.0063	0
f_{12}	0	30	Best	0.0022723	0.013165	1.4508e-06
			Ave	0.0867	0.0338	0.0181
			Std	0.0829	0.0177	0.0132
f_{13}	0	30	Best	1.5471e-05	0.17871	1.3974e-05
			Ave	0.0703	0.3632	0.0334
			Std	0.1215	0.2226	0.0578
f_{14}	1	2	Best	0.998	0.998	0.998
			Ave	0.9977	5.0529	1.5643
			Std	7.5593e-04	4.6424	0.9685
f_{15}	0	30	Best	0.00030822	0.00030749	0.00030749
			Ave	0.0039	0.0071	4.6020e-04
			Std	0.0081	0.0102	3.7379e-04
f_{16}	-10.5363	4	Best	-10.1532	-10.1531	-10.1532
			Ave	-6.6291	-10.1529	-10.1531
			Std	3.3624	1.4832e-04	8.3666e-05

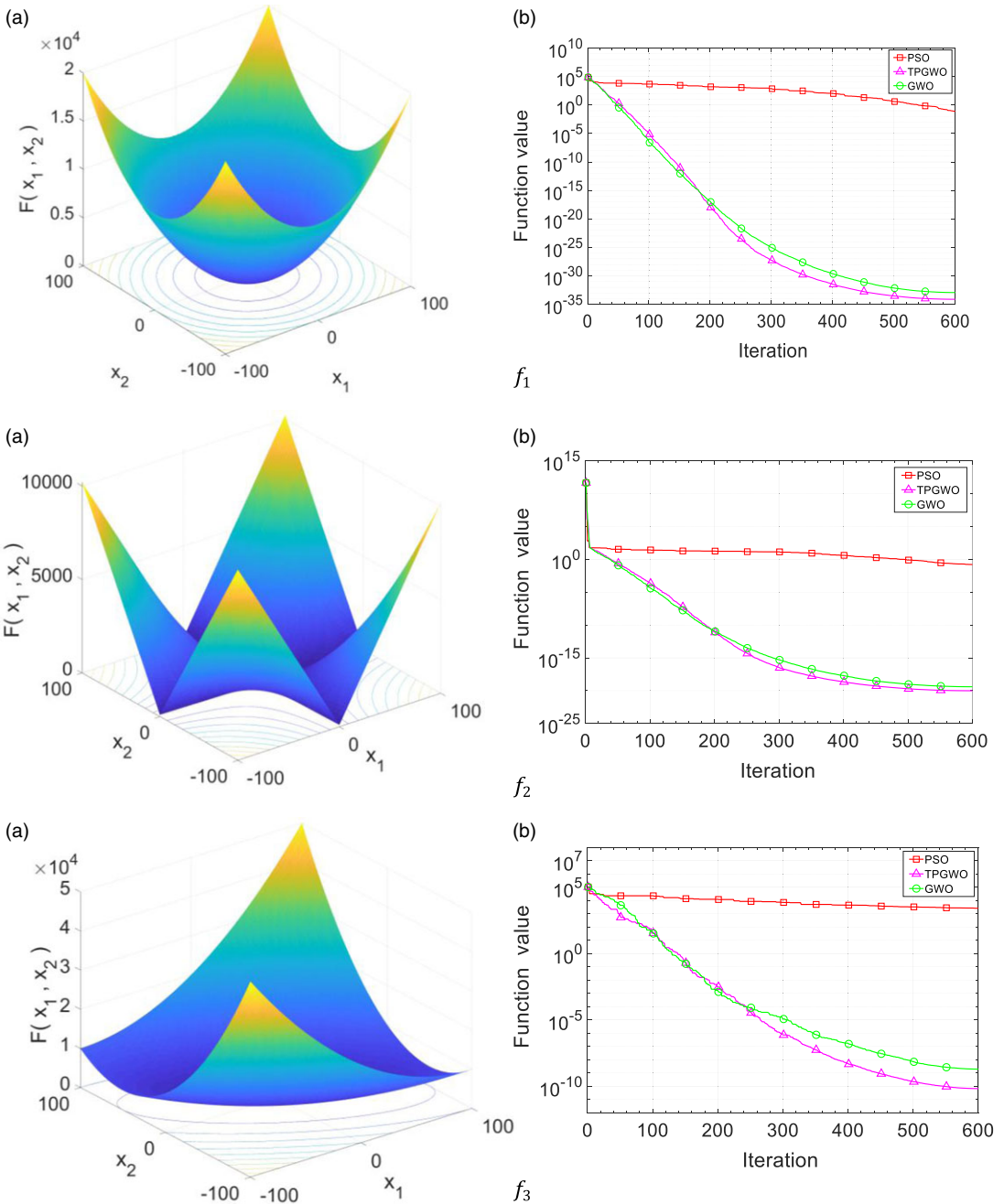


Fig. 13. Space graph of test function and convergence curve.

5.3 Simulation analysis of path planning

The TPGWO algorithm, GWO algorithm and PSO algorithm are applied to path planning, respectively, for simulation test. Let the initial population number of the three algorithms N be 30 and the maximum number of iterations t_{max} be 600, and the initialization method proposed in Section 4.1 and the fitness function proposed in Section 4.3 in this paper are used. The environment maps are the established grid maps of $10 * 10$, $15 * 15$ and $20 * 20$. Each grid map randomly generates different number of obstacles.

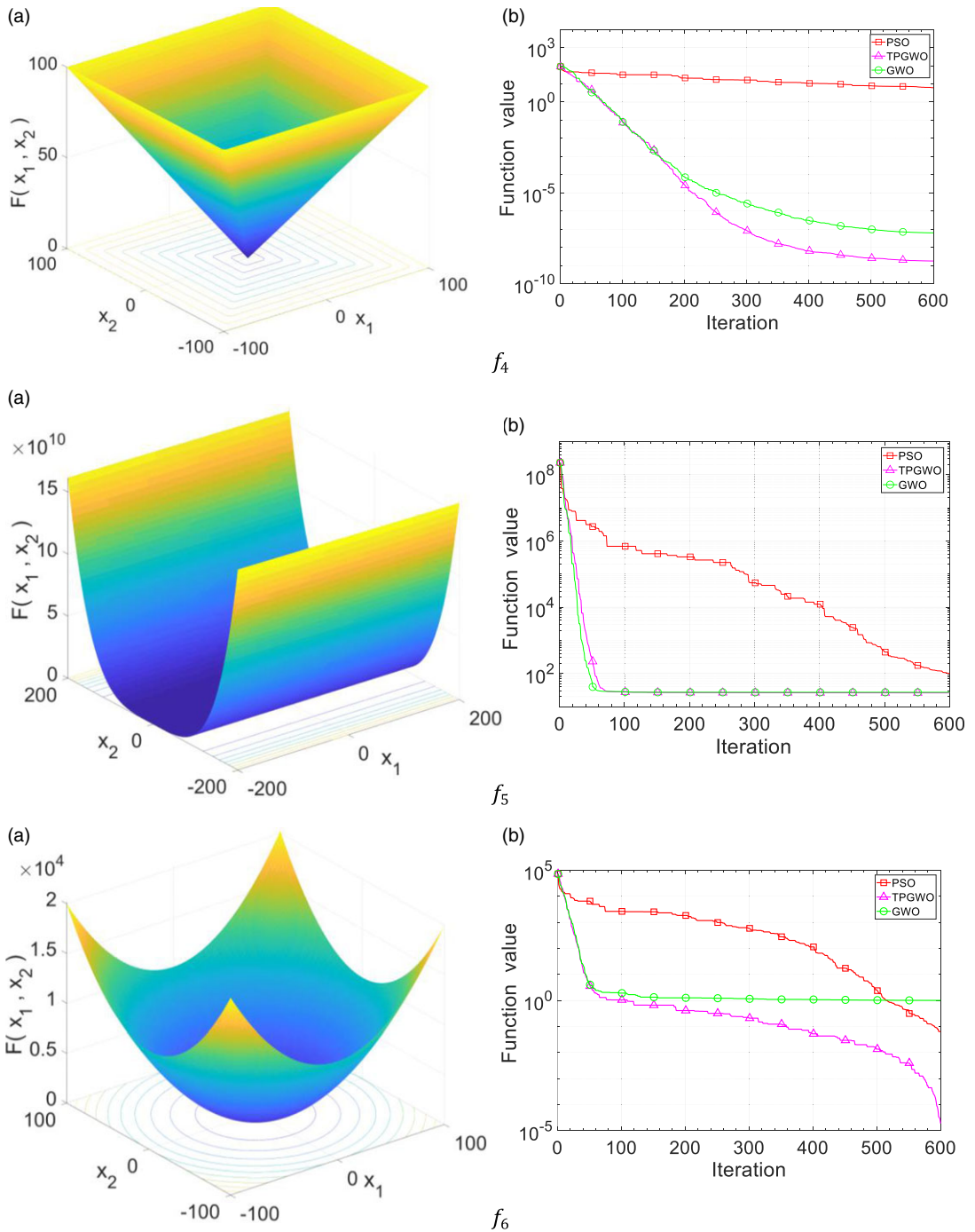


Fig. 13. Continued.

Finally, the path length and path planning time obtained by the three algorithms under three different maps are analysed.

Figures 14–19 show the path diagrams and convergence curves of the three algorithms under grid map $10 * 10$, $15 * 15$ and $20 * 20$, respectively. It can be seen from the figures that under the same maximum iteration times and initial population, the three algorithms can find a reachable path from the starting

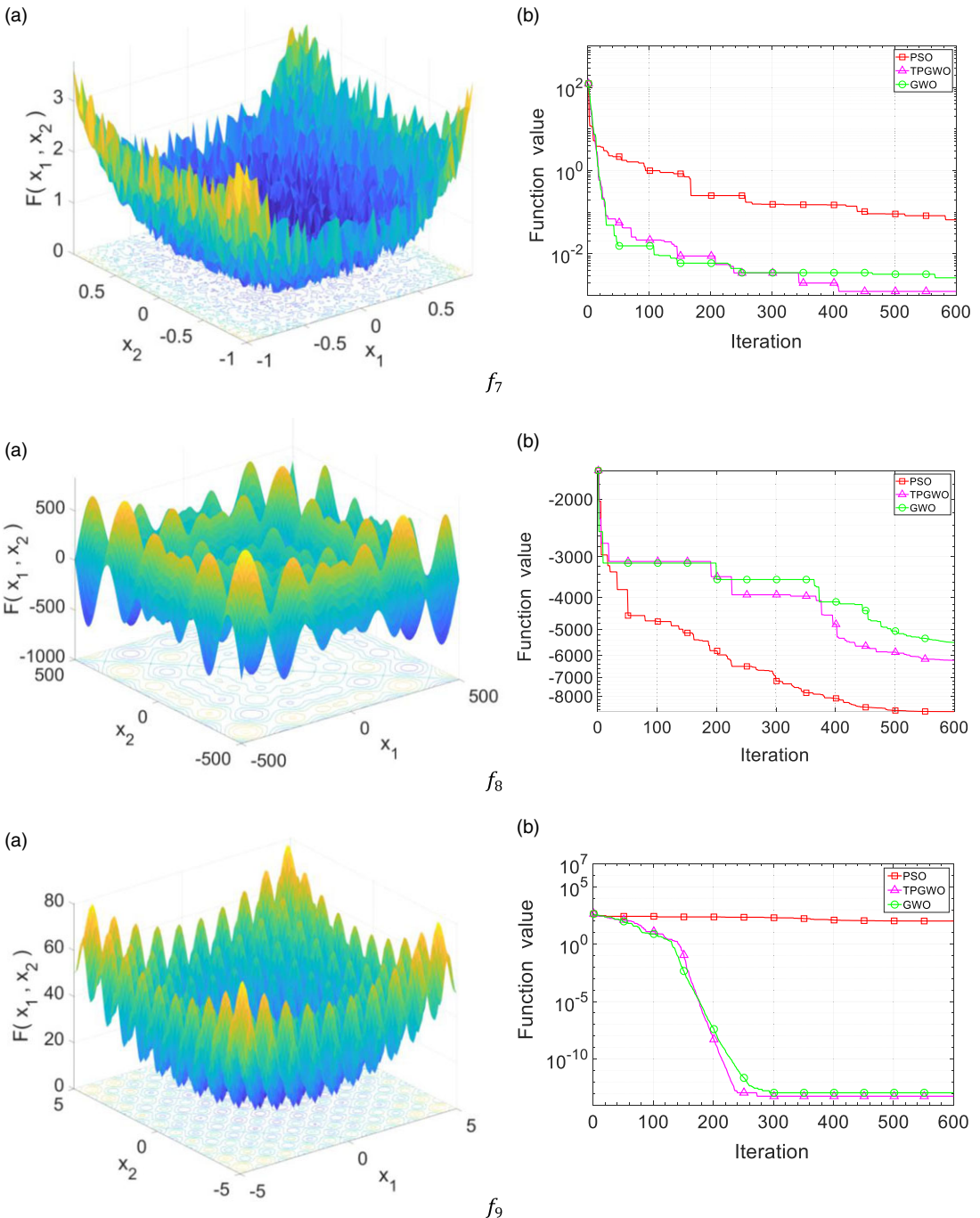


Fig. 13. Continued.

point to the target point. Among them, TPGWO algorithm has high convergence accuracy under three grid maps with different areas. In the convergence process, PSO algorithm and GWO algorithm are easy to fall into local optimization, while TPGWO algorithm can better jump out of local optimization, and can find global optimization and improve accuracy. It can be seen from the operation results in Table IV that the path selected by the TPGWO algorithm is better and the path planning takes less time. In order

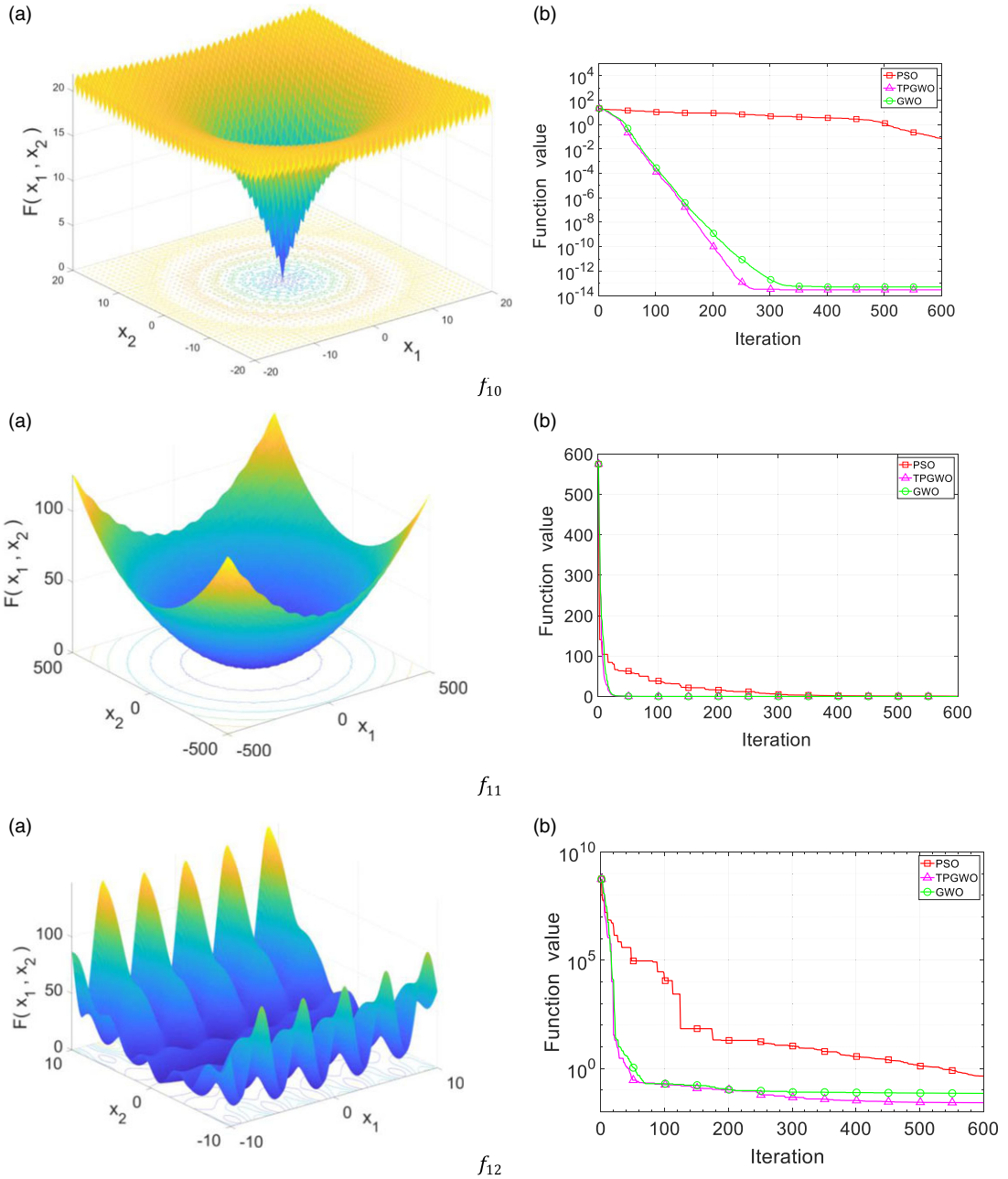


Fig. 13. Continued.

to avoid the occasionality of the three algorithms in the path planning, 15 different maps are generated at random for each different map size, under the condition of a certain number of obstacles. The length and running time of the path obtained by the three algorithms under each map are recorded, and the product of them is used to show the performance of the algorithm. As shown in the Fig. 20, horizontal axis represents the map order under certain map size and certain number of obstacles, and vertical axis represents the product of path length and running time. It can be seen from the figure that the TPGWO algorithm has good accuracy and stability in path planning under different maps. In conclusion, under the same conditions, TPGWO algorithm can find the reachable path from the starting point to the target

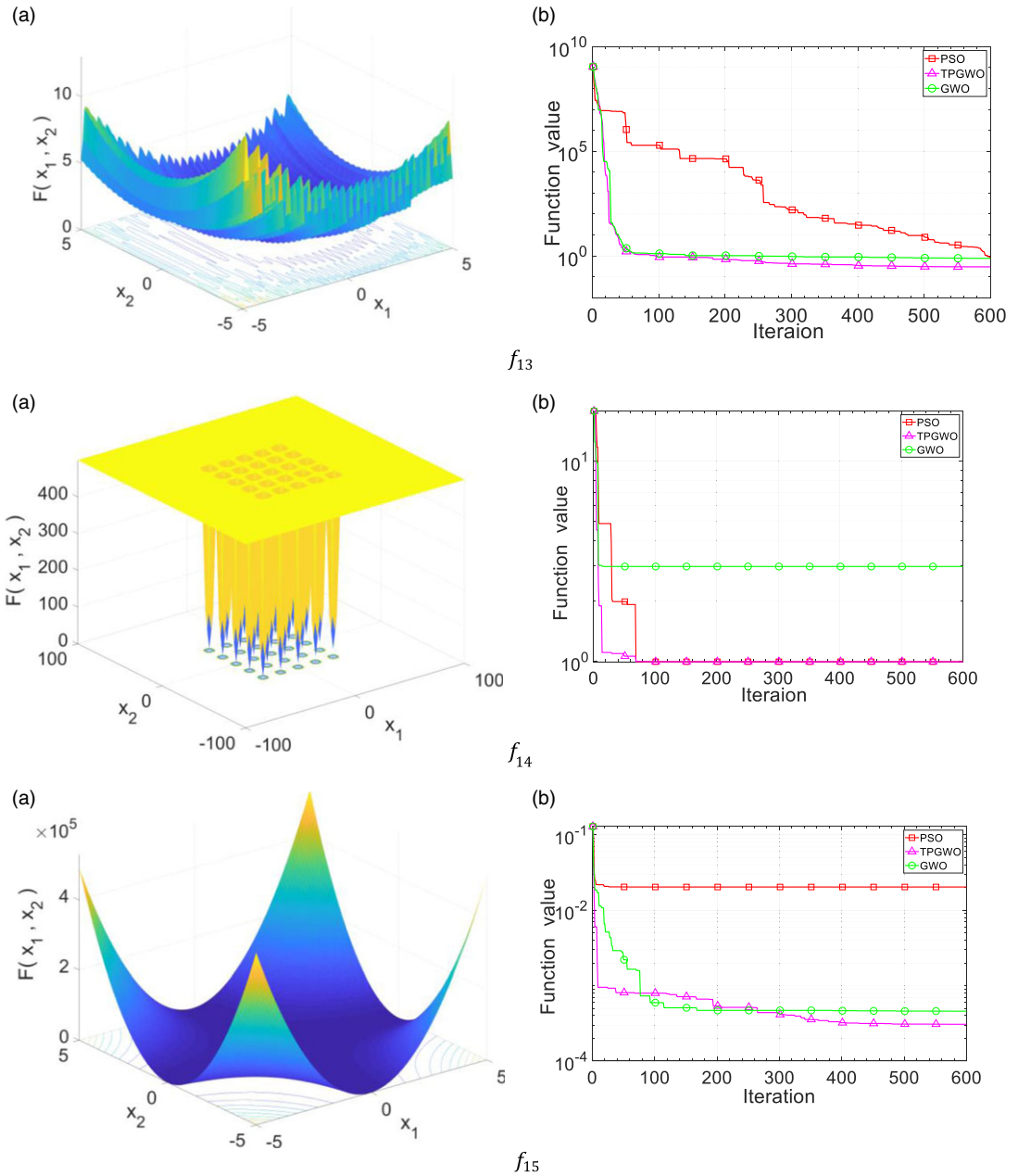


Fig. 13. Continued.

point better, faster and more stably than PSO algorithm and GWO algorithm in the path planning of the detection robot.

In order to verify that the convergence factor function in TPGWO algorithm in Section 4.2 can improve the convergence speed and accuracy by adjusting the parameter value, the simulation is carried out in the map of different number of obstacles. Taking the 15×15 grid map as an example, a simple map with fewer obstacles and a complex map with more obstacles are randomly generated. As shown in Table V, the initial population number of TPGWO algorithm N is 30, and the maximum number of iterations t_{\max} is 600, and the convergence factor functions are a'_1, a'_2, a'_3 .

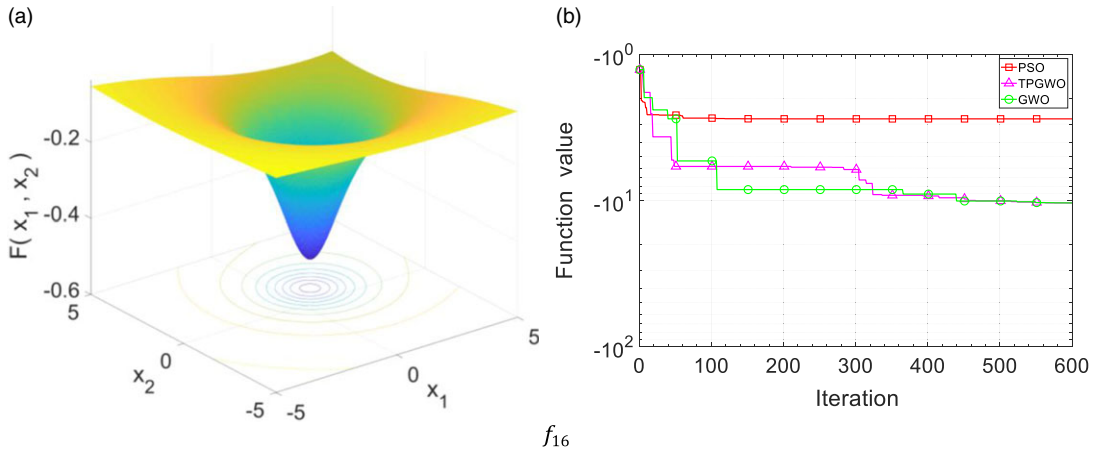


Fig. 13. Continued.

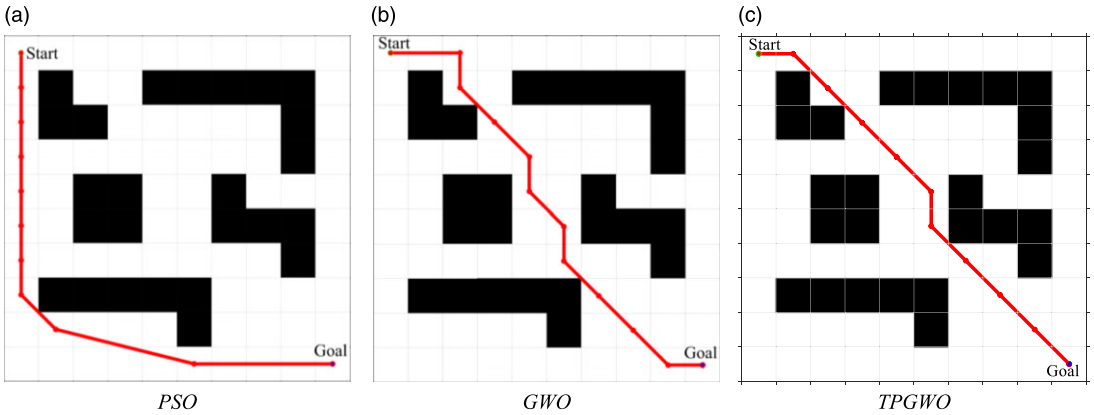


Fig. 14. Results of path planning of PSO, GWO and TPGWO under 10×10 grid map. (a) PSO. (b) GWO. (c) TPGWO.

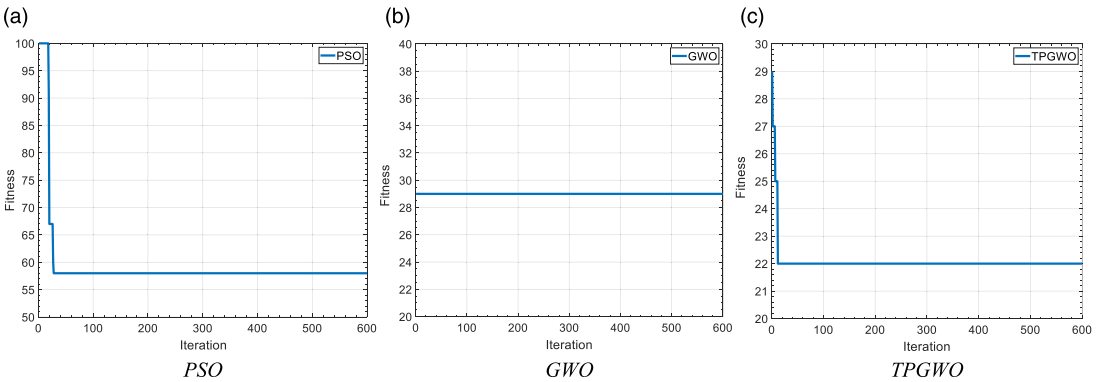


Fig. 15. Convergence curve of PSO, GWO and TPGWO for path planning under 10×10 grid map. (a) PSO. (b) GWO. (c) TPGWO.

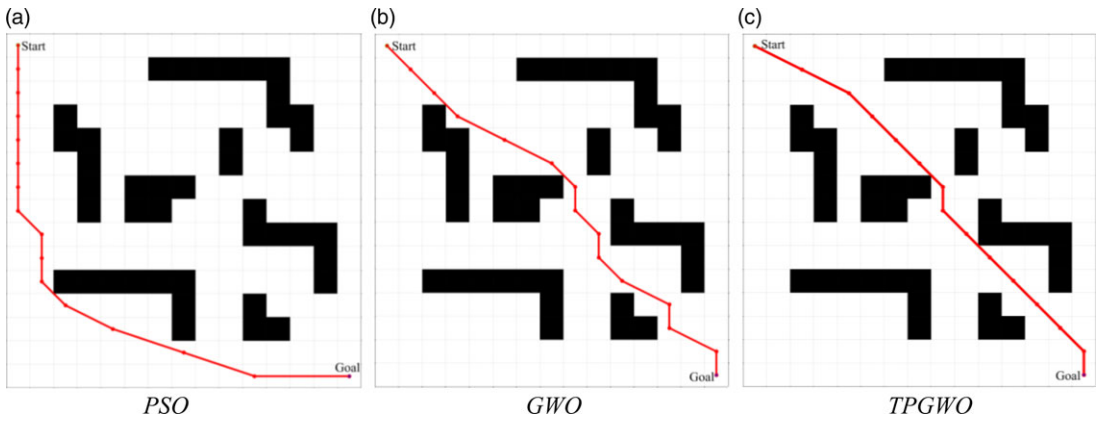


Fig. 16. Results of path planning of PSO, GWO and TPGWO under 15*15 grid map. (a) PSO. (b) GWO. (c) TPGWO.

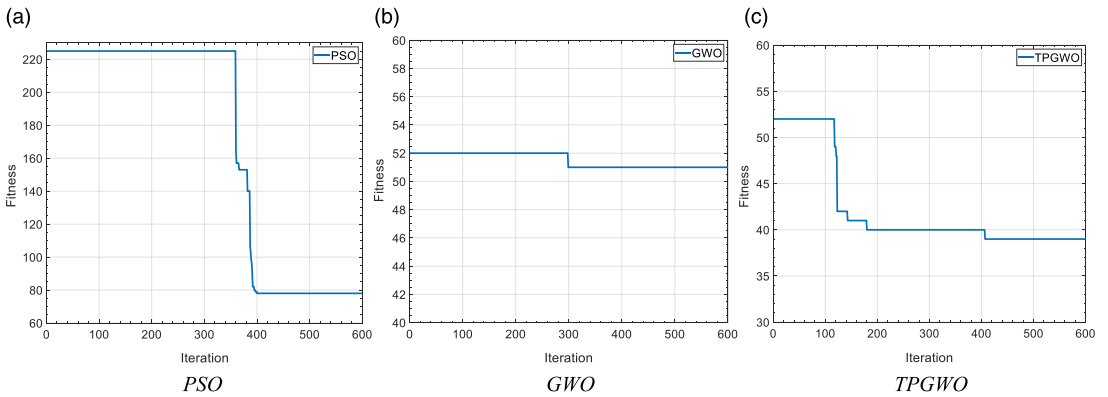


Fig. 17. Convergence curve of PSO, GWO and TPGWO for path planning under 15*15 grid map. (a) PSO. (b) GWO. (c) TPGWO.

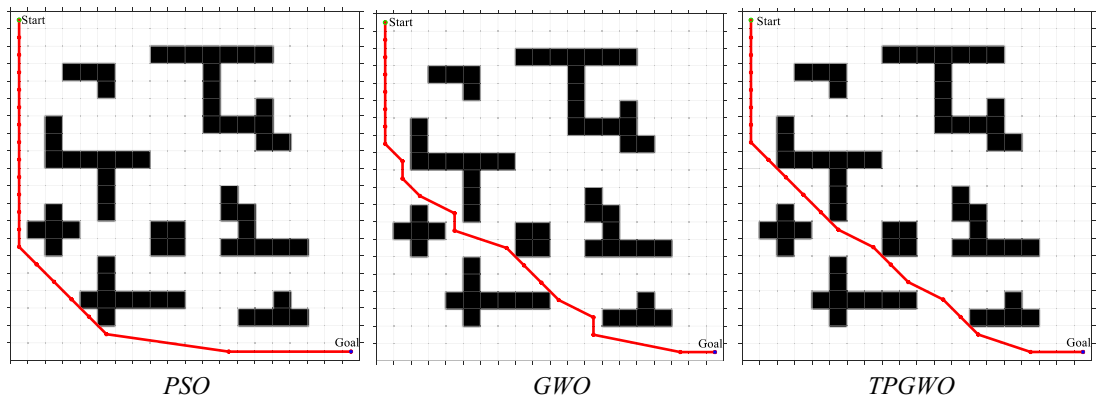


Fig. 18. Results of path planning of PSO, GWO and TPGWO under 20*20 grid map. (a) PSO. (b) GWO. (c) TPGWO.

Table IV. Path length and running time of different algorithms in different map environments.

Acreage	PSO		GWO		TPGWO	
	Length	Time (s)	Length	Time (s)	Length	Time (s)
10 * 10	58	1.83	29	1.62	22	1.26
15 * 15	78	2.62	51	2.27	39	1.72
20 * 20	122	3.44	70	2.86	54	1.91

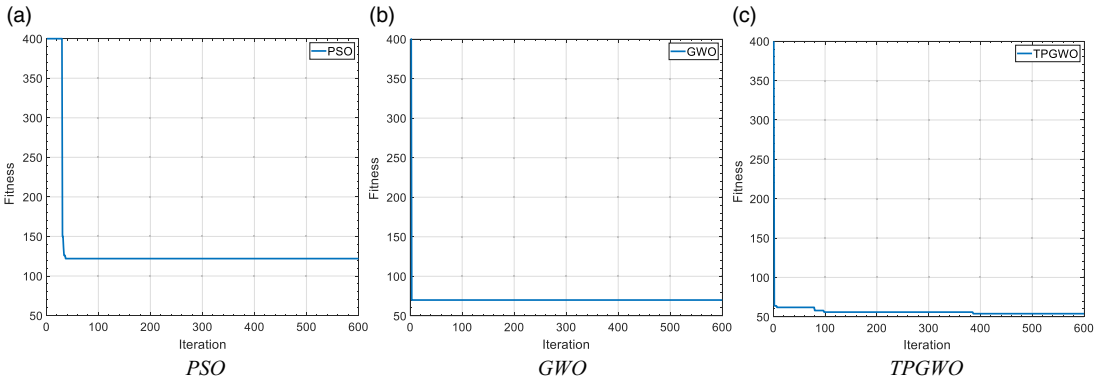


Fig. 19. Convergence curve of PSO, GWO and TPGWO for path planning under 20 * 20 grid map. (a) PSO. (b) GWO. (c) TPGWO.

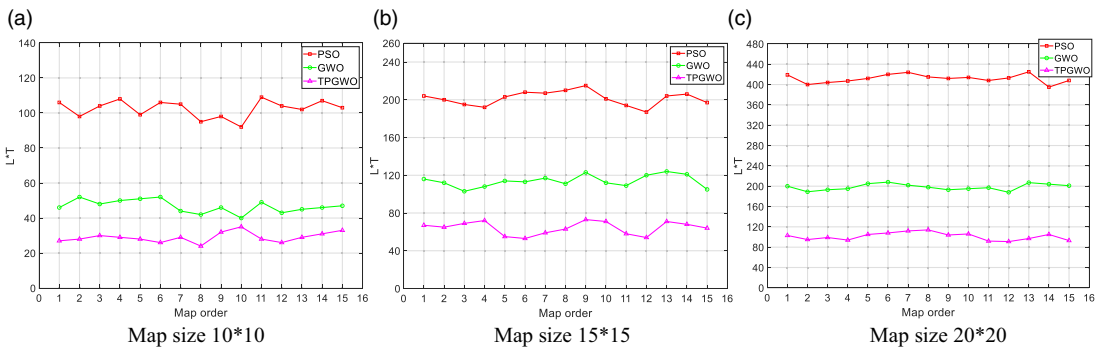


Fig. 20. Statistical results of three algorithms under different map sizes. (a) Map size 10 * 10. (b) Map size 15 * 15. (c) Map size 20 * 20.

Figures 21–24 show the path diagram and convergence curve of TPGWO algorithm under three convergence factor functions. According to the graph analysis, in the simple 15 * 15 grid map with few obstacles, the same effective path can be obtained by using the convergence factor function a_2' with earlier turning points and using the convergence factor function a_1' with constant turning point, but the convergence speed of the former is faster. In the 15 * 15 grid map with more and complex obstacles, the resulting path is more accurate using the convergence factor function a_3' with slightly delayed turning point than using function a_1' . According to the data in Table VI, the convergence factor function a_2' can effectively shorten the running time of the algorithm, and the convergence factor function a_3' can effectively improve the accuracy of the algorithm. In order to avoid the occasionality of the path planning, two kinds of maps with different numbers of obstacles are generated. Ten different maps are generated

Table V. Parameter selection of TPGWO path planning.

Main parameter settings	
TPGWO	Wolves number $n = 30$, Convergence factor $a = a'_1$ ($p = 300/600$)
	Wolves number $n = 30$, Convergence factor $a = a'_2$ ($p = 280/600$)
	Wolves number $n = 30$, Convergence factor $a = a'_3$ ($p = 320/600$)

Table VI. Running results with different convergence factor functions under different maps of obstacles with the same area.

Function	Few obstacles (Map 15 * 15)		More obstacles (Map 15 * 15)	
	Length	Time (s)	Length	Time (s)
a'_1	35	1.63	86	1.73
a'_2	35	1.24	–	–
a'_3	–	–	73	1.75

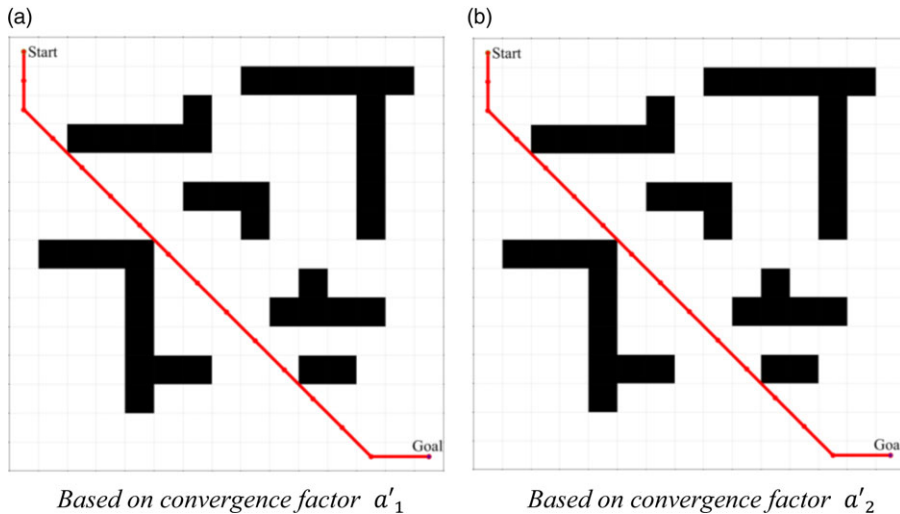


Fig. 21. Path planning results with convergence factor a'_1 , a'_2 under simple map with few obstacles. (a) Based on convergence factor a'_1 . (b) Based on convergence factor a'_2 .

for fewer and more obstacles, respectively. The path size and running time of different convergence factors under each map are analysed, as shown in Fig. 25, horizontal axis represents the map order, and vertical axis represents the product of path length and running time. It can be seen from the figure that adjusting the convergence factor parameters does not affect the stability of the TPGWO. Therefore, when the map area and the number of obstacles are certain, properly adjusting the turning point of the convergence factor curve according to the proportion of the number of obstacles to the map area can effectively improve the convergence speed and convergence accuracy of TPGWO algorithm in the application of patrol robot path planning.

The comparison of different algorithms in path planning is shown in Fig. 26. Figure 26(a) shows the path results of three algorithms under different maps, and Fig. 26(b) shows the path results of different convergence factor parameters in TPGWO algorithm under different number of obstacles under the same map size. The following conclusions can be drawn from the simulation results: (1) it is verified that TPGWO algorithm has better convergence, stability, rapidity and accuracy than PSO algorithm and

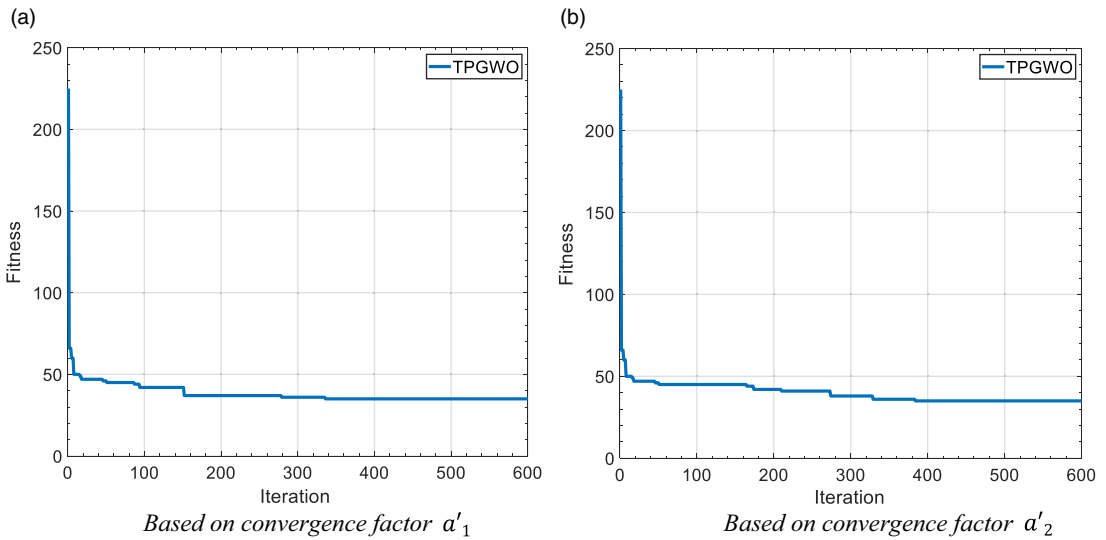


Fig. 22. Convergence curve of path planning with different convergence factors under simple map with few obstacles. (a) Based on convergence factor a'_1 . (b) Based on convergence factor a'_2 .

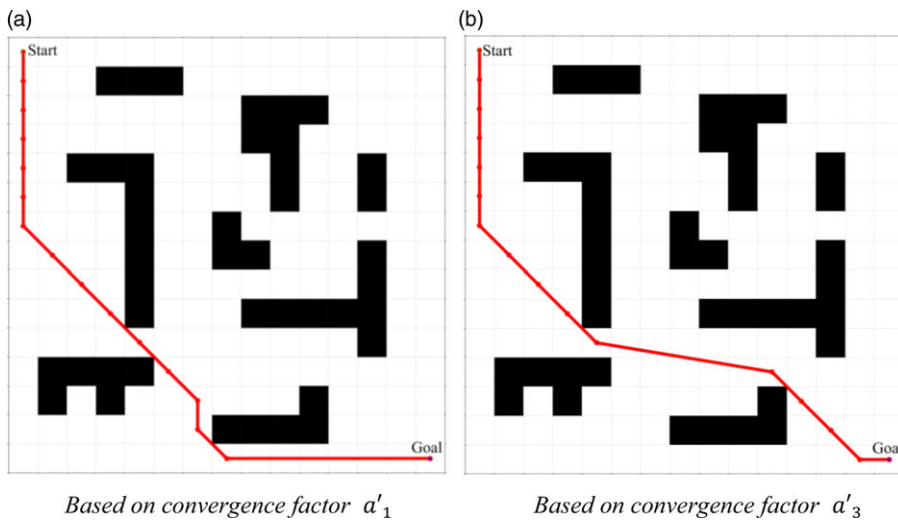


Fig. 23. Path planning results with convergence factor a'_1 , a'_3 under a complex map with a large number of obstacles. (a) Based on convergence factor a'_1 . (b) Based on convergence factor a'_3 .

GWO algorithm under the path planning of different maps; (2) it is verified that when the TPGWO algorithm is used for path planning, the parameters in the convergence factor function can be adjusted by analysing the number of obstacles in the map, thus improving the time and accuracy of path planning.

6. Conclusion

Many algorithms can be used in path planning of inspection robot. However, different optimization algorithms have their defects and limitations in the application of path planning. In this paper, TPGWO algorithm is proposed to overcome the shortcomings of GWO algorithm in the path planning of patrol robot, such as small search range, slow convergence speed and easily falling into local optimization. TPGWO algorithm uses the idea of cross-mutation and roulette to increase the initial population number and improves the convergence factor function in GWO algorithm to a nonlinear convergence factor

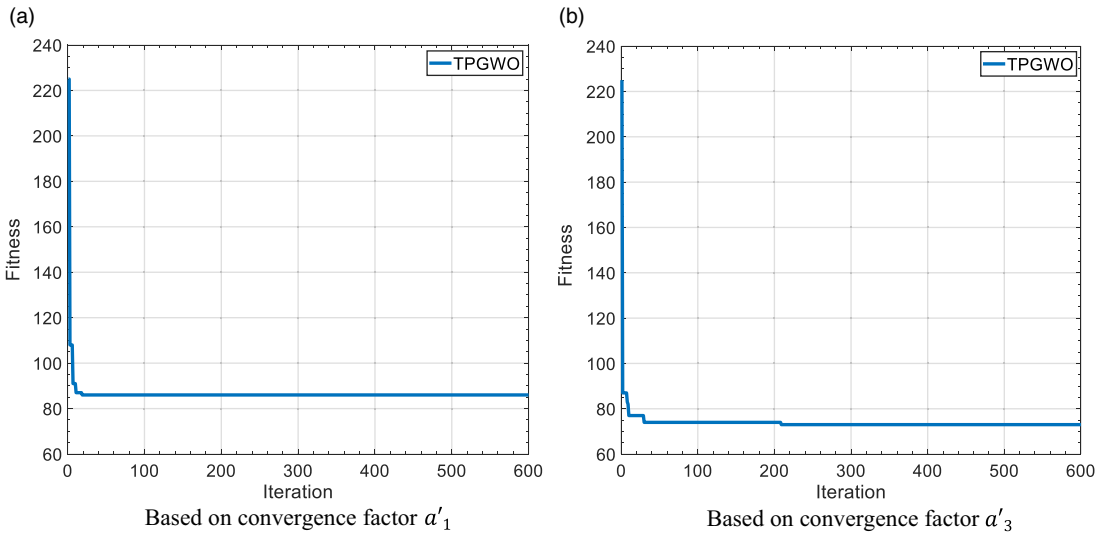


Fig. 24. Convergence curve of path planning with different convergence factors on complex map with many obstacles. (a) Based on convergence factor a'_1 . (b) Based on convergence factor a'_3 .

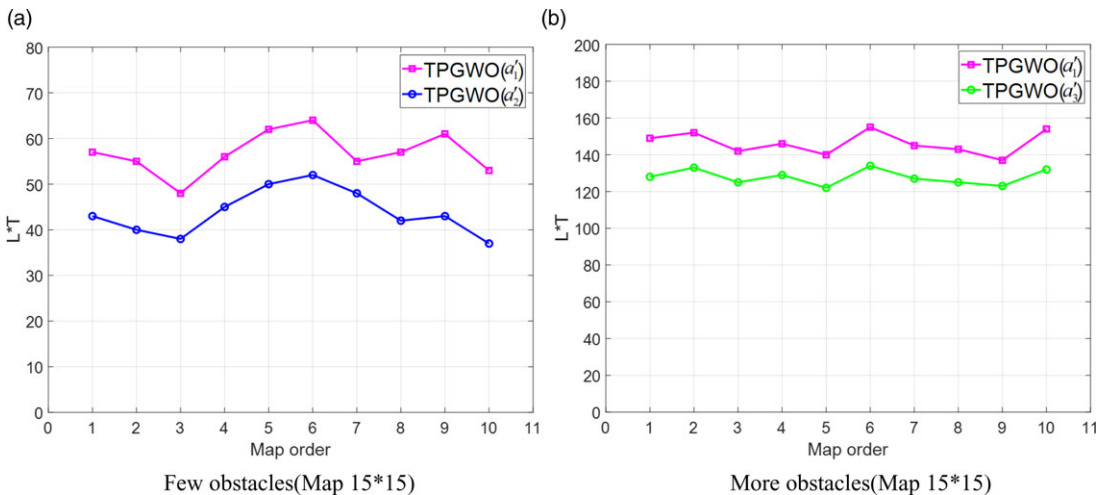


Fig. 25. Values of $L * T$ in TPGWO algorithm under different convergence factor functions. (a) Few obstacles (Map 15 * 15). (b) More obstacles (Map 15 * 15).

function that can adjust the turning point, which can not only expand the early search range but also speed up the convergence speed in the later stage to avoid falling into local optimization. At the same time, the calculated turning time and turning angle of the path are added to the fitness function of path planning to improve the accuracy of path selection. The performance of TPGWO is tested by 16 test functions. The results show that TPGWO has better convergence, stability and accuracy than PSO and GWO under the same conditions. Finally, through the path planning simulation experiment, it is found that TPGWO has better path and shorter path planning time than PSO algorithms and GWO algorithms in different map environments. At the same time, the parameters in the convergence factor function of TPGWO are adjusted according to the number of obstacles in the map. The simulation results of path planning show that adjusting the parameters in the convergence factor function properly can effectively improve the accuracy and speed of path planning. As a global path planning optimization algorithm, TPGWO has achieved good results in different map environments, but it has certain defects in local and

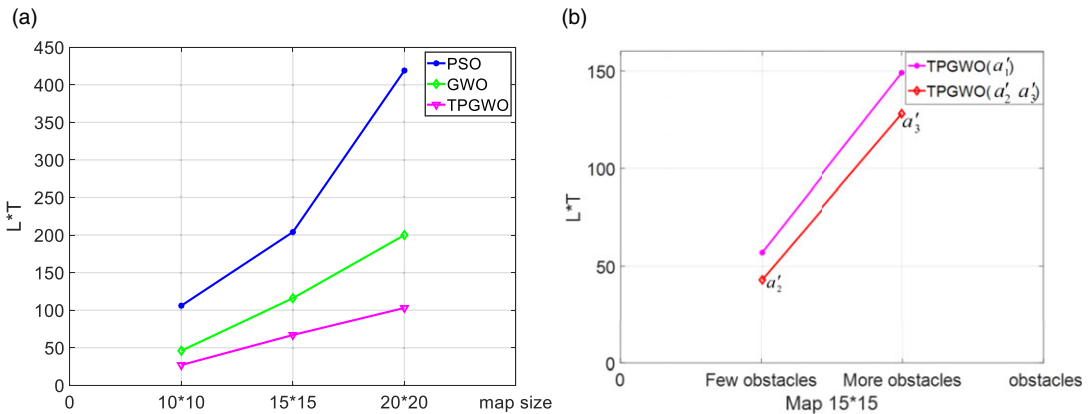


Fig. 26. Simulation verification results.

dynamic obstacles. Therefore, in subsequent research, local optimization algorithms such as artificial potential field method and dynamic window approach can be improved and combined with TPGWO to solve such problems, so that the patrol robot can effectively avoid dynamic obstacles and complete the inspection task more accurately.

Author contributions. Qian Zhang and Yingying Li conceived and designed the study. Xucheng Ning and Lei Pan conducted programming, and Xucheng Ning performed the analyses. Qian Zhang and Xucheng Ning wrote the article and revised it according to the nice comments and suggestions from the reviewers. Rui Gao and Liyang Zhang gave necessary help during simulations and valuable suggestions on the analyses.

Financial support. This research is funded by National Natural Science Foundation of China (Grant No. 62101377) and Tianjin Municipal Education Commission Scientific Research Program Project (Grant No. 2020KJ057 and Grant No. 2018KJ176).

Conflicts of interest. The authors declare none.

References

- [1] Y. Wu, L. Sun, W. Han, L. Chen, X. Liu and Z. Wu, "Characteristics and development prospect of computer automatic patrol inspection technology," *J. Phys. Conf. Ser.* **1650**(3), 032172 (2020).
- [2] Y. Gao, S. Li, X. Wang and Chen Q., "A Patrol Mobile Robot for Power Transformer Substations Based on ROS," *In: 2018 Chinese Control And Decision Conference (CCDC)* (2018).
- [3] W. Sun, K. Wei, Z. Liu, Q. Li and X. Xu, "Linear quadratic Gaussian control for wireless communication reliability for a mobile monitoring robot in a UHV power substation," *IEEE Syst. J.* **16**(3), 4149–4159 (2022). doi: [10.1109/JSYST.2022.3143866](https://doi.org/10.1109/JSYST.2022.3143866).
- [4] Y. H. Liang, C. Cai, J. Zhao, Chang G.-X., Wang L.-H. and Lv X.-L., "Local Environments Modelling and Path Planning for Patrol Robot in the Substation," *In: IEEE International Conference on Computer Supported Cooperative Work in Design (IEEE, 2016)*.
- [5] B. Hichri, A. Gallala, F. Giovannini and S. Kedziora, "Mobile robots path planning and mobile multirobots control: A review," *Robotica* **40**(12), 1–14 (2022). doi: [10.1017/S0263574722000893](https://doi.org/10.1017/S0263574722000893).
- [6] A. Ras, A. Drr and B. Pr, "A Boundary Node Method for path planning of mobile robots," *Robot. Auton. Syst.* **123**(21), 103320.
- [7] Z. Tang and H. Ma, "An overview of path planning algorithms," *IOP Conf. Ser. Earth Environ. Sci.* **804**(2), 022024(10pp) (2021).
- [8] L. Deng, H. Chen, H. Liu, H. Zhang and Y. Zhao, "Overview of UAV Path Planning Algorithms," *In: 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)* (2021) pp. 520–523. doi: [10.1109/ICETCI53161.2021.9563566](https://doi.org/10.1109/ICETCI53161.2021.9563566).
- [9] X. Wang and X. Meng, "UAV Online Path Planning Based on Improved Genetic Algorithm," *In: 2019 Chinese Control Conference (CCC)* (2019) pp. 4101–4106. doi: [10.23919/ChiCC.2019.8866205](https://doi.org/10.23919/ChiCC.2019.8866205).
- [10] A. Arikere, G. S. Kumar and S. Bandyopadhyay, "Optimisation of Double Wishbone Suspension System Using Multi-Objective Genetic Algorithm," *In: Simulated Evolution and Learning - 8th International Conference, SEAL 2010, Kanpur, India, December 1-4, 2010, Proceedings (DBLP, 2010)*.

- [11] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang and Q. Fu, "Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms," *IEEE Access* **7**(1), 105086–105099 (2019). doi: [10.1109/ACCESS.2019.2932008](https://doi.org/10.1109/ACCESS.2019.2932008).
- [12] Y. Tan, J. Ouyang, Z. Zhang, Y. Lao and P. Wen, "Path planning for spot welding robots based on improved ant colony algorithm," *Robotica* **41**(3), 1–13 (2022). doi: [10.1017/S026357472200114X](https://doi.org/10.1017/S026357472200114X).
- [13] C. Pozna, R-E. Precup, E. Horvath and E. M. Petriu, "Hybrid particle filter-particle swarm optimization algorithm and application to fuzzy controlled servo systems," *IEEE Trans. Fuzzy Syst.* **30**(10), 4286–4297 (Oct. 2022).
- [14] P. Sudhakara, V. Ganapathy and K. Sundaran, "Mobile Robot Trajectory Planning Using Enhanced Artificial Bee Colony Optimization Algorithm," In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (2017) pp. 363–367. doi: [10.1109/ICPCSI.2017.8392316](https://doi.org/10.1109/ICPCSI.2017.8392316).
- [15] W. Wang, Q. Tao, Y. Cao, X. Wang and X. Zhang, "Robot time-optimal trajectory planning based on improved Cuckoo search algorithm," *IEEE Access* **8**(1), 86923–86933 (2020). doi: [10.1109/ACCESS.2020.2992640](https://doi.org/10.1109/ACCESS.2020.2992640).
- [16] G. Li, W. Dong, Y. Wang, D. Zhu and Q. Liu, "Path Planning of Underwater Vehicles Based on Improved Whale Optimization Algorithm," In: *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)* (2021) pp. 444–448. doi: [10.1109/CACRE52464.2021.9501356](https://doi.org/10.1109/CACRE52464.2021.9501356).
- [17] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.* **69**(Mar.), 46–61 (2014).
- [18] S. Zhang, Q. Luo and Y. Zhou, "Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method," *Int. J. Comput. Intell. Appl.* **16**(02), 1750012 (2017).
- [19] S. Li and F. Wang, "Research on optimization of improved grey wolf optimization-extreme learning machine algorithm in vehicle route planning," *Discrete Dyn. Nat. Soc.* **2020**(Oct.), 1–7 (2020).
- [20] J. Liu, X. Wei and H. Huang, "An improved grey wolf optimization algorithm and its application in path planning," *IEEE Access* **9**(1), 121944–121956 (2021). doi: [10.1109/ACCESS.2021.3108973](https://doi.org/10.1109/ACCESS.2021.3108973).
- [21] Q. Luo, S. Zhang, Z. Li and Y. Zhou, "A novel complex-valued encoding grey wolf optimization algorithm," *Algorithms* **9**(1), 4 (2016).
- [22] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Appl. Soft Comput.* **60**(c), 115–134 (2017).
- [23] W. Zhang, S. Zhang, F. Wu and Y. Wang, "Path planning of UAV based on improved adaptive grey wolf optimization algorithm," *IEEE Access* **9**(1), 89400–89411 (2021). doi: [10.1109/ACCESS.2021.3090776](https://doi.org/10.1109/ACCESS.2021.3090776).
- [24] J. Li and F. Yang, "Task assignment strategy for multi-robot based on improved grey wolf optimizer," *J. Ambient Intell. Human Comput.* **11**(4), 6319–6335 (2020).
- [25] J. S. Wang and S. X. Li, "An improved grey wolf optimizer based on differential evolution and elimination mechanism," *Sci. Rep.* **9**(1), (2019). doi: [10.1038/s41598-019-43546-3](https://doi.org/10.1038/s41598-019-43546-3).
- [26] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Design Eng.* **5**(4), 458–472 (2018).
- [27] Z. Teng, J. Lv, L. Guo and Yuanyuan X., "An improved hybrid grey wolf optimization algorithm based on Tent mapping," *J. Harbin Inst. Technol.* **50**(11), 46–55 (2018).
- [28] M. H. Nadimi-Shahraki, S. Taghian and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Syst. Appl.* **166**(Mar.), 113917 (2020).
- [29] M. W. Guo, J. S. Wang, L. Zhu, S. S. Guo and Xie W., "An improved grey wolf optimizer based on tracking and seeking modes to solve function optimization problems," *IEEE Access* **PP**(99), 1 (2020).
- [30] C. Qu, W. Gai, M. Zhong and J. Zhang, "A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning," *Appl. Soft Comput.* **89**(1), 106099 (2020).
- [31] S. Zhang, Y. Zhou, Z. Li and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.* **99**(Sep.), 121–136 (2016).
- [32] J. Hui, L. Xiang and J. Su, "An Improved Grey Wolf Optimizer Algorithm Integrated with Cuckoo Search," In: *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (IEEE, 2017).
- [33] X. Zhao, P. Zhan and Y. Liu, "Rapid Construction Algorithm of 3D Urban Road Network from Raster Maps," In: *2017 International Conference on Virtual Reality and Visualization (ICVRV)* (2017) pp. 109–114. doi: [10.1109/ICVRV.2017.00029](https://doi.org/10.1109/ICVRV.2017.00029).
- [34] L. Ren, "A novel raster map exchange scheme based on visual cryptography," *Adv. Multimed.* **2021**(1), 1–7 (2021).
- [35] M. J. Zhang, D. Y. Long, X. Wang and Yang J., "Research on convergence of grey wolf optimization algorithm based on Markov chain," *Acta Electronica Sinica* **48**(8), 1587–1595 (2020).
- [36] N. H. Awad, M. Z. Ali and P. N. Suganthan, "Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood for Solving CEC2017 Benchmark Problems," In: *2017 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2017) pp. 372–379.
- [37] P. Zhang and J. Zhang, "Lower limb exoskeleton robots' dynamics parameters identification based on improved beetle swarm optimization algorithm," *Robotica* **40**(8), 2716–2731 (2022). doi: [10.1017/S0263574721001922](https://doi.org/10.1017/S0263574721001922).
- [38] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.* **376**(1), 113609 (2021).