IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 9, 2017, accepted October 17, 2017, date of publication October 23, 2017,
date of current version November 14, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2765504

Three-Dimensional Underwater Path Planning Based on Modified Wolf Pack Algorithm

LANYONG ZHANG[1,2], (Member, IEEE), LEI ZHANG[1], (Member, IEEE), SHENG LIU[1], JIAJIA ZHOU[1], AND CHRISTOS PAPAVASSILIOU[2], (Senior Member, IEEE)

[1]College of Automation, Harbin Engineering University, Harbin 150001, China
[2]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Lanyong Zhang (zlyalf@sina.com)

This work was supported in part by the National Natural Science Foundation of China Subsidization Project under Grant 51579047, in part by the National Natural Science Foundation of China under Grant 51609048, in part by the National Key Technology Support Program under Grant 2013BAG25B01, in part by the Research Fund for the Doctoral Program of Higher Education under Grant 20132304120015, in part by the Doctoral Scientific Research Foundation of Heilongjiang under Grant LBH-Q14040, in part by the National Defense Fundamental Research Funds under Grant IEP14001, in part by the Open Project Program of State Key Laboratory of Millimeter Waves under Grant K201707, in part by the Natural Science Fund of Heilongjiang Province under Grant QC2017048, and in part by the Fundamental Research Funds for the Central Universities under Grant HEUCF160414.

ABSTRACT Path planning is an important problem in autonomous control technology. This paper aims to overcome the shortcomings of the wolf pack algorithm (WPA), such as slow rate of convergence and low convergence precision, by improving the three intelligent behaviors of the WPA, namely, scouting, summoning, and beleaguering. To improve the scouting behavior, interactive scouting is proposed to increase the interactivity among wolf pack. Furthermore, to improve the summoning behavior, a prey-based adaptive step model is established to improve the searching ability. Finally, calculation rules of new beleaguering behavior are designed, which enhance the local fine search ability considerably. A fast path planning method based on dubins path was proposed, which applied the dubins path planning to meet angle control constraint and tunes the turning radius to meet control constraint. The dubins path planning method based on the modified WPA is proposed by establishing the underwater environment threat model under the condition of autonomous underwater vehicle constraint. The path between the path points is the shortest, the threat is minimal, and the energy consumption is the least without the consideration of ocean current. Simulation results show that the modified WPA has a high rate of convergence and good local search capability in the high-precision, high-dimensional, and multi-peak function; moreover, it does not converge prematurely.

INDEX TERMS AUV, path planning, modified wolf pack algorithm, autonomous underwater vehicles.

I. INTRODUCTION

Resource depletion, population growth, and environmental degradation are three major problems that have plagued humankind for a long time. With the progress of science and the development of human society, resource depletion has emerged as the primary problem that could threaten the survival of humankind [1]. It is well known that 70% of the earth's surface is covered by water. Although the underwater environment is rich in resources, it is extremely difficult for human beings to harness those resources. With the development of automation technology and the advancement of unmanned technology, the risk associated with underwater exploration has been reduced considerably. Among the available technologies, autonomous underwater vehicles (AUV) are highly advantageous for the exploration of underwater resources owing to their low cost and superior performance [2].

In unmanned underwater control domain, an efficient and accurate path planning method is crucial, because it enables an AUV to complete a task successfully while exhibiting excellent performance in underwater exploration. The path planning is the planning of an optimal path for the AUV from the deployment platform to the intended target with the shortest distance and minimum risk under the limitations of AUV performance and marine environment complexity [3]–[5]. Schematic diagram is shown in figure 1.

There are some typical path planning algorithms such as A* modifications, JPS, D*, Phi* and RRT algorithms [9]–[14]. Existing path planning methods mainly involve land-based 2D planar path planning [25]. Compared with

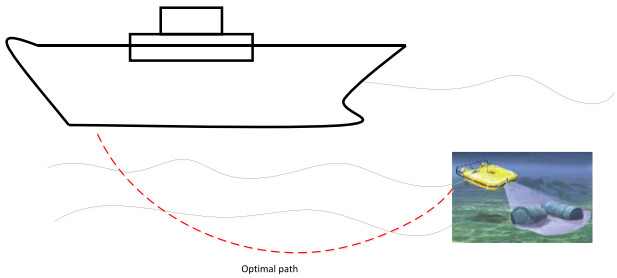


FIGURE 1. Path planning diagram.

2D environments, 3D environments have not been investigated extensively, especially with regard to underwater 3D path planning [5]. Owing to the non-linearity of underwater motion and the high complexity of the marine environment, land-based algorithms cannot be employed directly for underwater applications. A few solutions have been proposed for AUV. Cao *et al.* [6] proposed a new path planning method using a genetic algorithm combined with 3D dubins motion for an underwater glider in a 3D environment and this method can save the consumption of energy in a further way. Dong *et al.* [7] proposed a new path planning algorithm based on an extreme learning machine, which is different from analytic functions. However, this method has some limitations that the map of the environment must be built and the vehicle has been able to localize itself, the high-level task of path planning must be achieved in order for the platform to complete its mission objectives. Gautam *et al.* [8] solved the path planning problem by combining biologically inspired algorithms such as the flower pollination algorithm and the genetic algorithm. This method turns out to be better in computational complexity and simulation environment.

3D path planning problems are eventually converted into mathematical problems involving solving functions and intelligent algorithms are inevitably applied [15], [24]. By imitating the characteristics of wolf pack and exploiting concept of the survival of the fittest, the present paper proposes an improved wolf pack algorithm (WPA) [21]–[23]. To overcome the shortcomings of the wolf pack algorithm, such as slow rate of convergence and low convergence precision, by improving the three intelligent behaviors of wolf pack algorithm, namely scouting, summoning, and beleaguering. To improve the scouting behavior, interactive scouting is proposed to increase the interactivity among wolf pack. Further, to improve the summoning behavior, a prey-based adaptive step model is established to improve the searching ability. Finally, calculation rules of new beleaguering behavior are designed, which enhance the local fine search ability considerably. The simulation results show that the modified wolf pack algorithm has high rate of convergence and good local search capability in the high-precision, high-dimensional, multi-peak function, moreover, it does not converge prematurely. Artificial wolves can not only perceive information from their companions but also transmit prey information, which allows for effective control of the balance between global searching and local searching. In addition, the raiding step length was modified from a fixed value to an adaptive value to improve the global prey searching ability, preventing the algorithm from falling into local optima. In order to meet AUV constraints, a fast path planning method based on dubins path which applied the dubins path planning to meet angle control constraint and tune the turning radius to meet control constraints was proposed.

The remainder of this paper is organized as follows. Section II establishes the mathematical models, including the constraint function model of the underwater environment and the fitness function model that ensures good performance of AUV. Section III describes the modified wolf pack algorithm and explains how to apply this algorithm to 3D underwater path planning. Section IV describes the simulation of the modified wolf pack algorithm, and compares the modified wolf pack algorithm with the original wolf pack algorithm in terms of path planning. The simulation results show that the modified wolf pack algorithm requires fewer parameters for 3D underwater path planning, has better global search ability, can be quickly optimized without falling into local optima, and exhibits stronger computational robustness in complex environments. Finally, Section V concludes the paper and briefly explores directions about how to evolve the energy saving performance on MAUV for future work.

II. ESTABLISHMENT OF MATHEMATICAL MODELS

Three-dimensional underwater path planning mainly refers to the planning of an optimal path for an AUV from the deployment platform to the intended target with the least energy consumption, minimum risk, and stealthiest route under the limitations of AUV performance and marine environment complexity. Therefore, the path planning problem can be transformed into the optimal control problem with composite constraints. In order to use the proposed algorithm to optimize the calculation, besides to establishing the AUV underwater kinetic equation, the constraint function model and the fitness function model must be established.

A. CONSTRAINT FUNCTION MODEL

1) ENVIRONMENTAL THREATS CONSTRAINT

The AUV enters a predetermined sea area from the deployment platform to carry out resource exploration or cruises through the target area to find enemy targets using the sonar system. Therefore, it is faced with many environmental threats. Data of potential threats in the target area can be measured through the deployment platform and information fusion can yield accurate information. In this study, the terrain obstacles and threats such as underwater continental shelf, continental slope and trench are translated into mountain peaks of various shapes [20].

Equating the environmental and terrain threats to particular terrain, and superimposing the location and action onto a digital map are equivalent to changing the virtual terrain, and treating the terrain as a prohibited area. After this processing, the known terrain obstacles and threats in the region are

merged into integrated terrain information, which simplifies the path planning problem considerably. According to the equivalent method, the threats and terrain obstacles in the underwater environment are modeled. The constraint function model is expressed as follows:

$$\begin{cases} Z(x,y) = h_0 + \sum_{j=1}^{N} h_j^{\max} \cdot e^{-[\frac{k_j^x \cdot (x - x_j^{\max})}{x_j^{\max}}]^2 - [\frac{k_j^y \cdot (y - y_j^{\max})}{y_j^{\max}}]^2} \\ Z(x,y) < h_{\max} \end{cases} \quad (1)$$

where h0 is the height of the seabed plane relative to the reference terrain, and N is the number of threats in the seabed environment, i.e., the number of slopes and terrains. When N increases, the environment becomes more complex and more threats. Further, $h_j^{\max}$ is the height of the vertex of the jth equivalent threat, $x_j^{\max}$ and $y_j^{\max}$ are the corresponding X-axis and Y-axis coordinates of the jth equivalent threat point, k_j^x and k_j^y are the corresponding X-axis and Y-axis slopes of the jth equivalent threat, respectively. The larger the values of k_j^x and k_j^y, the steeper is the axis, i.e., the faster is the rise, and vice versa. In addition, x and y are the corresponding X-axis and Y-axis coordinates, Z is the depth corresponding to each point on the plane, and $h_{\max}$ is the maximum depth of the ocean relative to the reference terrain. The equivalent constraint function model of environmental constraints and terrain threats is shown in figure 2.

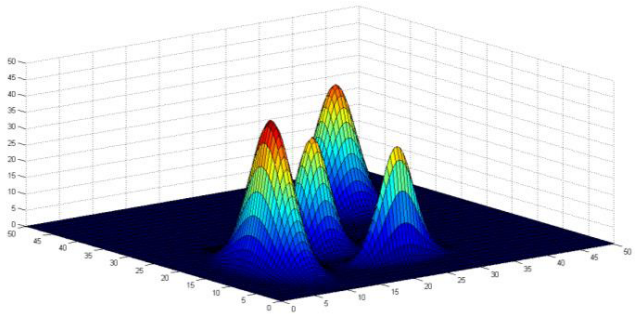


FIGURE 2. Environmental constraints and terrain threats.

2) AUV PERFORMANCE CONSTRAINTS

In practice, AUV limits the maximum turning angle. The previous path planning method cannot be directly applied, need to be smooth improvement. In this section, according to the AUV performance constraints, the dubins path is used to smooth the path.

a: DUBINS PATH

In the previous path planning study, the curve of the path planning often has a large corner between the path points. Under the realistic condition, there are position information and velocity information at each position of AUV for every time, previous path planning study can't meet the requirements of path planning in realistic condition. Therefore, it is necessary to find the optimal path between the initial position (starting position) and the end position under the direction [17].

TABLE 1. Situations of existing motion types.

Motion Type	Existence condition
LSL and RSR	always exist
LSR	$O_{AL}O_{BR} < 2r_{\min}$
RSL	$O_{AR}O_{BL} < 2r_{\min}$
LRL	$O_{AL}O_{BL} > 4r_{\min}$
RLR	$O_{AR}O_{BR} > 4r_{\min}$

The optimal path based on dubins is the constraint on the maximum turning angle of AUV, only the path that satisfies the maximum turning angle is the feasible path. The optimal dubins path exists among six types, which can be divided into two families. The first family starts with a left (L) or right (R) turning, follows by a straight segment (S), and ends up with a left or right turning (denoted by LSR, LSL, RSL and RSR, respectively). In the other family, an arc of circle can be followed with opposite directions instead of the straight segment (denoted by RLR and LRL) under certain circumstances.

The existence of each situation is based on the initial conditions, varies from 4 to 6 types accordingly. Establishing the earth-frame at the initial point A with x-axis pointing to the initial heading, and define $B = (x_f y_f)$, ψ_f are the destination and final heading; $r_{\min}$ is the turning radius. The centers of the left/right turn in dubins motion are denoted by O_{AL}, O_{AR}, O_{BL} and O_{BR}. The existences of motion types are shown in table 1.

According to the table above, the first family starts with a left (L) or right (R) turning, follows by a straight segment (S), and ends up with a left or right turning, besides LSR and RSL, there always exist a dubins path, LSR and RSL path need to exist $O_{AL}O_{BR} < 2r_{\min}$ and $O_{AR}O_{BL} < 2r_{\min}$. In the other family, the distance between the centers of the two minimum radii requires more than four times the minimum turning radius.

Therefore, the final path planning needs to meet the AUV minimum turning radius of the constraints.

This paper describes the problem by taking the dubins curve of LSR motion type as an example. LSR path as shown in figure 3.

As shown in figure 3, the AUV starts at M with the angle α and the speed vector $\mathbf{a_1}$. When the AUV takes O_1 as the center and r_1 as the radius around θ_1, the AUV moves in a straight line until it enters the radius of round 2. Like the motion of round 1, the AUV eventually moves along the $\mathbf{a_2}$, and complete the switch of direction.

In the figure 3, α and β are the AUV initial angle, and the specified termination angle, $\mathbf{r_{1s}}$, $\mathbf{r_{1f}}$, $\mathbf{r_{2s}}$ and $\mathbf{r_{2f}}$ are the vectors of arc to center. $\mathbf{a_1}$ and $\mathbf{a_2}$ are the starting vector and

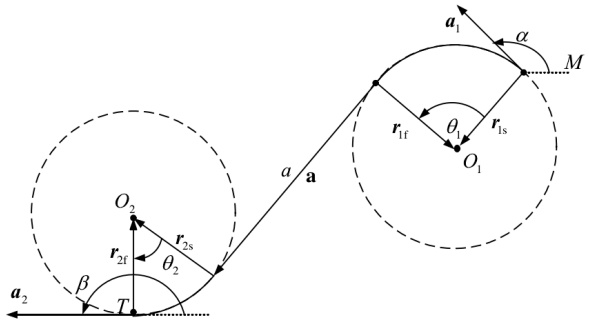


FIGURE 3. LSR path.

the terminal vector of the two arcs respectively. **a** is a tangent vector connecting two arcs, and the length of the vector is a. r_1, r_2, θ_1 and θ_2 denote the radius of the arcs and the turning angles. $M(x_0, y_0)$ and $T(x_e, y_e)$ represent the coordinate of the starting point and the turning end point respectively.

where $r_{\min} = 8000$, $x_0 = 40000$, $y_0 = 30000$, $\alpha = 135°$, and $\beta = 180°$. The LSR path with the minimum turning radius is the path with the shortest distance between the two points that satisfies the angle constraint. As the radius of the turn increases, the paths that satisfies the angle constraint and sailing distance are shown in figure 4 and figure 5.

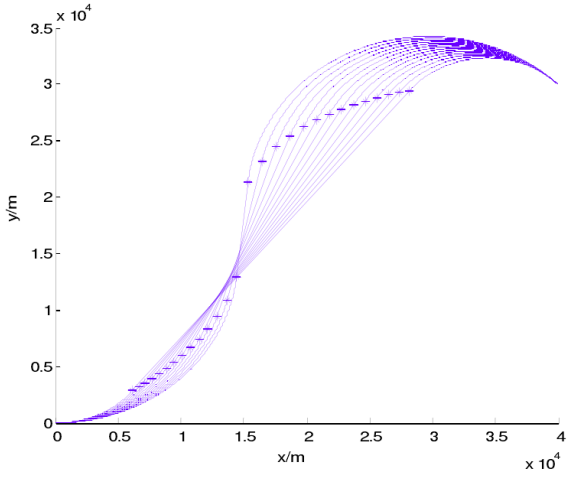


FIGURE 4. Schematic diagram of path variation with $r_{\min}$.

With the increase of the turning radius, the LSR range is linearly increased. When the radius increases to two rounds, the straight-line distance is zero, and then the radius is increased to meet the requirement of CLC (circle-line-circle) smooth connection. The path is CC (circle-circle) path, and the range meets the angle of the maximum range of constraints. Sailing distance is the maximum range that satisfies the angle constraint. The minimum turning radius must be maintained between the two stroke points to minimize the range, so that the length of the sailing distance is highly dependent on the performance of AUV. When the performance of AUV is higher, and the smaller the minimum turning radius is satisfied, a more excellent path can be obtained.

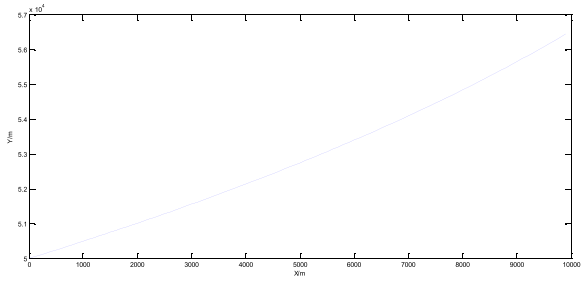


FIGURE 5. Variation of navigation distance with $r_{\min}$.

b: DUBINS PATH PARAMETER CALCULATION

The previous path planning algorithm divides the space into the form of grid. Path planning is achieved by extending nodes and grids. However, the method of node connection is difficult to adapt to the specific direction of movement and motor performance limits, and the path must be smooth. The dubins curve is calculated by using the constraints of AUV. The path is shown in figure 3, and the dubins path parameters are solved in this section.

According to the definition of α, β, θ_1, θ_2 and the constraints, we can get equation 2.

$$\theta_1 - \theta_2 = -\alpha + \beta \tag{2}$$

The starting point is M, the end point is T, and the position transformation vector **p** can be written as follows:

$$\mathbf{p} = \mathbf{r_{1s}} - \mathbf{r_{1f}} + \mathbf{a} + \mathbf{r_{2s}} - \mathbf{r_{2f}} \tag{3}$$

where $\mathbf{r_{is}}$ and $\mathbf{r_{if}}(i = (1, 2))$ are the two center vectors of the ith arc.

$$\mathbf{r_{1s}} = r_1 \begin{bmatrix} \cos(\frac{\pi}{2} + \alpha) \\ \sin(\frac{\pi}{2} + \alpha) \end{bmatrix} = r_1 \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \tag{4-a}$$

$$\mathbf{r_{1f}} = \mathbf{R}(\boldsymbol{\theta_1})\mathbf{r_{1s}} = r_1 \mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \tag{4-b}$$

$$\mathbf{a} = a\mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} \tag{4-c}$$

$$\mathbf{r_{2s}} = -\frac{r_2}{r_1}\mathbf{r_{1f}} = -r_2 \mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \tag{4-d}$$

$$\mathbf{r_{2f}} = -\frac{r_2}{r_1}\mathbf{R}(-\boldsymbol{\theta_2})\mathbf{r_{1f}} = -r_2 \mathbf{R}(-\boldsymbol{\theta_2})\mathbf{R}\boldsymbol{\theta_1} \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \tag{4-e}$$

$$\mathbf{p} = \begin{bmatrix} x_e - x_0 \\ y_e - y_0 \end{bmatrix} \tag{4-f}$$

where $R(\theta)$ is the rotation matrix, and θ is a positive value representing counterclockwise rotation.

$$\mathbf{R}(\boldsymbol{\theta}) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5}$$

According to equation 4, equation 3 can be written as follows:

$$\begin{aligned}\begin{bmatrix} x_e - x_0 \\ y_e - y_0 \end{bmatrix} &= r_1 \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} - r_1 \mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \\ &\quad + a\mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix} - r_2 \mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \\ &\quad + r_2 \mathbf{R}(-\boldsymbol{\theta_2})\mathbf{R}(\boldsymbol{\theta_1}) \begin{bmatrix} -\sin\alpha \\ \cos\alpha \end{bmatrix} \end{aligned} \tag{6}$$

$$\begin{aligned} x_e - x_0 &= -r_1\sin\alpha - (r_1 + r_2)(-\cos\theta_1\sin\alpha - \sin\theta_1\cos\alpha) \\ &\quad + a(\cos\theta_1\cos\alpha - \sin\theta_1\sin\alpha) \\ &\quad + r_2(-\cos(\theta_1 - \theta_2)\sin\alpha - \sin(\theta_1 - \theta_2)\cos\alpha) \end{aligned} \tag{7-a}$$

$$\begin{aligned} y_e - y_0 &= r_1\cos\alpha - (r_1 + r_2)(\cos\theta_1\cos\alpha - \sin\theta_1\sin\alpha) \\ &\quad + a(\sin\theta_1\cos\alpha - \cos\theta_1\sin\alpha) \\ &\quad + r_2(\cos(\theta_1 - \theta_2)\cos\alpha - \sin(\theta_1 - \theta_2)\sin\alpha) \end{aligned} \tag{7-b}$$

Simplify the formula 7, and together with equation 2, we can get equation 8.

$$\begin{cases} x_e - x_0 = -r_1\sin\alpha + (r_1 + r_2)\sin(\alpha + \theta_1) \\ \quad + a\cos(\theta_1 + \alpha) - r_2\sin\beta \\ y_e - y_0 = r_1\cos\alpha - (r_1 + r_2)\cos(\alpha + \theta_1) \\ \quad + a\sin(\theta_1 + \alpha) + r_2\cos\beta \\ \theta_1 - \theta_2 = -\alpha + \beta \end{cases} \tag{8}$$

The minimum turning radius r as a constraint can be calculated as follows:

$$r_1 = r_2 = r_{\min} = \frac{V^2}{a_{\max}} \tag{9}$$

Here, V is the sailing speed, and $a_{\max}$ is the maximum acceleration that AUV can use for turning.

According to equation 9, we can get the formula as follows:

$$\begin{aligned} &(x_e - x_0 + r_{\min}\sin\alpha + r_{\min}\sin\beta)\sin(\alpha + \theta_1) \\ &\quad - (y_e - y_0 - r_{\min}\cos\alpha - r_{\min}\cos\beta)\cos(\alpha + \theta_1) = 2r_{\min} \end{aligned} \tag{10}$$

To facilitate writing, define intermediate variables:

$$\begin{aligned} c_1 &= x_e - x_0 + r_{\min}\sin\alpha + r_{\min}\sin\beta \\ c_2 &= y_e - y_0 - r_{\min}\cos\alpha - r_{\min}\cos\beta \end{aligned} \tag{11}$$

According to equation 10, equation 11 can be simplified as follows:

$$\frac{c_1}{\sqrt{c_1^2 + c_2^2}}\sin(\alpha + \theta_1) - \frac{c_2}{\sqrt{c_1^2 + c_2^2}}\cos(\alpha + \theta_1) = \frac{2r_{\min}}{\sqrt{c_1^2 + c_2^2}} \tag{12}$$

Define the intermediate variable γ,

$$\begin{aligned} \cos\gamma &= \frac{c_1}{\sqrt{c_1^2 + c_2^2}} \\ \sin\gamma &= \frac{c_2}{\sqrt{c_1^2 + c_2^2}} \end{aligned} \tag{13}$$

$$\gamma = \begin{cases} \gamma_0 & \sin\gamma > 0 \,\&\, \cos\gamma > 0 \\ 2\pi + \gamma_0 & \sin\gamma < 0 \,\&\, \cos\gamma > 0 \\ \pi - \gamma_0 & \cos\gamma > 0 \end{cases} \tag{14}$$

According to equation 14, equation 12 can be defined as follows:

$$\sin(\alpha + \theta_1 - \gamma) = \frac{2r_{\min}}{\sqrt{c_1^2 + c_2^2}} \tag{15}$$

Considering the interval of the trigonometric function, the general solution of θ_1 can be obtained from $r_{\min} > 0$:

$$\theta_1 = \begin{cases} 2k\pi + \arcsin\dfrac{2r_{\min}}{\sqrt{c_1^2 + c_2^2}} + \gamma - \alpha \\ (2k+1)\pi + \arcsin\dfrac{2r_{\min}}{\sqrt{c_1^2 + c_2^2}} + \gamma - \alpha \end{cases} \tag{16}$$

The complete path of the dubins curve can be obtained from the above calculations. According to the formula 16 and formula 8 can be obtained θ_2 and a, thus dubins curve path parameters have been solved.

B. FITNESS FUNCTION MODEL

The AUV needs to consider not only the energy consumption from source to destination but also the degree of danger during the voyage. Therefore, it needs to determine the shortest travelling distance and effectively avoiding the threat of collision. If the entire path is divided into n + 1 segments, then the fitness function model will be equivalent to the following equation:

$$F = \sum_{i=1}^{n+1} (\omega_1 l_i + \omega_2 f_i + \omega_3 E_i) \tag{17}$$

where l_i is the length of the ith segment, f_i is the degree of danger of the ith segment, and E_i is the energy consumed of the ith segment. ω_1 is the travel distance parameter, ω_2 is the travel threat parameter, and ω_3 is the energy consumption parameter. When ω_1 is large, the AUV tends to travel through the shortest path. Otherwise, it takes the longest path. When ω_2 is large, the AUV tends to avoid the environment threat by circumventing it. On the other hand, the AUV tends to pass through the environment when ω_2 is small, avoiding the threat by moving up or down. When ω_3 is large, the AUV tends to pay attention to the minimum consumption of AUV energy and vice versa. When $\omega_2 = 0$, the path planning of the AUV depends entirely on the travelling distance. ω_3 will determine the impact of energy on path planning.

In the path planning methods, we need to pay attention to AUV path length and energy consumption. Furthermore, it is necessary to determine the best sailing depth when the AUV travels under water. The risk factors increase with the depth, and frequent up and down motions are not conducive to effective AUV control. Such fluctuations have a significant impact on the AUV performance. Hence, equation 17 can be

rewritten as follows:

$$F = \sum_{i=1}^{n+1}[\omega_1\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} + \omega_2\frac{\sqrt{z_i - h_i^{ideal}}}{2} + \omega_3 E_i] \quad (18)$$

where h_i^{ideal} is the ideal depth, meaning that it is safest to travel at this depth. When ω_1 is large, the AUV tends to travel through the shortest path. Otherwise, it travels through the longest path. When ω_2 is large, the AUV tends to avoid the threats. Therefore, in this study, to ensure that the AUV travels through the shortest distance most safely, a large value is assigned to ω_1 and ω_3 to achieve the shortest path and minimal energy consumption, and a small value is assigned to ω_2 to avoid the threats.

In equation 18, E_i can be represented by the following two parts:

$$E_i = E_b + E_r \quad (19)$$

where E_b and E_r represent the energy consumed by the buoyancy and rotary actuators.

When the height of the floating (dive) in the ith path is D, there are:

$$E_b = \frac{2m}{\eta_b \rho}(\rho g D + P_0) \quad (20)$$

where m is the quality of AUV, η_b is a constant described by the buoyancy engine, ρ represents the density of the fluid in which it is located, g denotes the gravitational acceleration, and P_0 is the standard atmospheric pressure.

When the starting position and the end position of ith path changed, the energy consumed by the rotational actuator is expressed as:

$$E_r = \frac{1}{2\eta_r} a_r^2 \gamma^4 \quad (21)$$

where $\gamma = (\gamma_1 - \gamma_0)$, γ_1 and γ_0 are the heading angles of the end position and the starting position in the current path, η_r is a constant associated with the rotational actuator, and a_r is a constant value.

III. PRINCIPLE OF WOLF PACK ALGORITHM AND THE PATH PLANNING METHOD

A. WOLF PACK ALGORITHM

The development of intelligent algorithms, especially swarm intelligence, has significantly affected our lives in various aspects. In particular, it has promoted industrial development and production. The main feature of an intelligent algorithm is solution-finding performance. Wolf pack algorithm is a swarm intelligence optimization algorithm that was originally proposed in 2013. Since then, it has been employed in various fields, such as medicine, 3D sensor optimization, artificial neural networks, water conservation, and hydro power optimization [14], [18], [19]. Owing to its highly promising results, it has become one of the most popular intelligent computing algorithms with excellent prospects [16]. According to the wolf pack hunting behavior and prey distribution method, the algorithm abstracts four types of intelligent behaviors: scouting, summoning, beleaguering and population renewing. According to these behaviors, wolf pack algorithm exhibits the cooperation characteristics of wolf pack and fully traverses all solutions in the given space. A wolf pack consists of a lead wolf, scout wolves and ferocious wolves. The hunting process model is shown in figure 6.

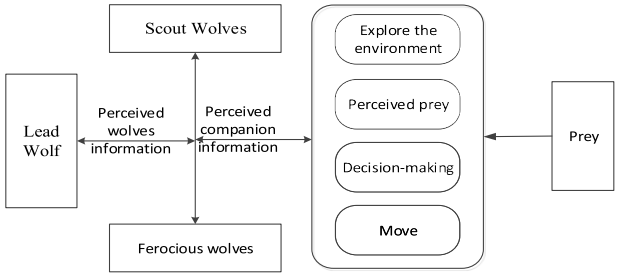


FIGURE 6. Hunting process model.

As is show in the figure 6, scout wolves and ferocious wolves perceived companion information, by explore the environment to perceive the prey and make the decision to move to the prey, then the lead wolf can compare the wolf pack information to complete the task of catching prey.

Through information transmission among one another as well as the perception of environmental information, the wolves coordinate and cooperate to complete the hunting process. Although many problems remain to be solved in the study of wolf pack algorithm, this paper presents a modified wolf pack algorithm to improve the convergence speed and convergence precision of the original wolf pack algorithm.

1) BASIC CONCEPT

Assuming that the wolf pack is in an (N × D)-dimensional space, where N is the total number of wolves and D is the number of variables to be optimized, $X_i = [x_1, x_2, \cdots, x_D]$ represents the position of the ith artificial wolf in the D-dimensional space, and x_D denotes the location of the Dth variable to be optimized within its optimal range. At this time, the function to be optimized is $Y = f(X)$, where Y represents the prey odor concentration of the artificial wolf, i.e., the fitness function value. The wolf population is N, and the lead wolf is the wolf that is closes to the prey among all the wolves, i.e., it has the maximum value of the optimization function. There is only one lead wolf. Scout wolves search for prey in the given space. The number of scout wolves' T takes integer value in the range $[N/(\alpha + 1), N/\alpha]$, where α is the scout wolf ratio factor, which determines the proportion of scout wolves in the wolf pack. Ferocious wolves hunt down the prey. The number of ferocious wolves' S equals the total number of wolves minus the lead wolf and scout wolves, expressed as $N - T - 1$.

2) MODIFIED SCOUTING BEHAVIOR

When modifying the scouting behavior, the scout wolf is treated as a basic unit in the calculation. First, the prey odor concentration of scout wolf i distributed in the hunting space is perceived. If the prey odor concentration of scout wolf i is greater than the odor concentration at the position of the lead wolf, i.e., $Y_i > Y_{lead}$, then scout wolf i replaces the lead wolf to perform the summoning behavior, i.e., $Y_{lead} = Y_i$. During the scouting behavior, if the prey odor concentration of scout wolf i is lower than the odor concentration at the position of the lead wolf, i.e., $Y_i < Y_{lead}$, then scout wolf i will take one scouting step forward ($step_a$ from its position according to equation 22) in the p direction and record its current location x_{id}^{new}. Then, it will perceive the odor concentration for the next step and record the odor concentration of the current position.

$$x_{id}^{new} = x_{id} + \sin(2\pi + p/h) \cdot step_a \tag{22}$$

where h is the search direction factor, and the greater the value of h, the finer the search is. However, if the value of h is too large, the algorithm operation will be prolonged, and it increases the likelihood of the algorithm falling into local extremum. To reduce the value of h, the search becomes coarser and the computation time becomes shorter. To enhance the communication between the lead wolf and the scout wolves as well as the diversity of the wolves while scouting, equation 23 is used to continue the search.

$$x_{id}^{new'} = x_{id} + \varphi_{id}(x_{best,\mathrm{d}} - x_{id}) + \phi_{id}(x_{jd} - x_{hd}) \tag{23}$$

where φ_{id} is a random number in the range [0 1], ϕ_{id} is a random number in the range [-1 1], x_{jd} and x_{hd} are the positions of scout wolf k and scout wolf h during scouting, respectively. After the calculation of equation 23, the communication among the population is enhanced and the diversity of the population is improved. By selecting the direction of the highest odor concentration, taking a step forward, and updating the location of the wolf, through the previous calculation, one can obtain T+1 odor concentrations $Y[1 \ldots T+1]$. If the odor concentration is greater than that of the lead wolf, i.e., $Y_{inew} > Y_{lead}$, then the scout wolf will replace the lead wolf to perform the summoning behavior. Next, the current position of the scout wolf will be updated and the next scouting task will be performed until the maximum number of scouts, $T_{\max}$ is reached, after which the wolf pack will transition into the summoning behavior.

3) MODIFIED SUMMONING BEHAVIOR

In the original summoning behavior, when the lead wolf initiates the summoning behavior, the ferocious wolves will rapidly move to the location of the lead wolf at a fixed raiding step. During the process of moving toward the lead wolf according to equation 24, the position of ferocious wolf i in the D-dimensional space is given by

$$x_{id}^{k+1} = x_{id}^{k} + step_b \cdot (g_d^k - x_{id}^k) / \left| g_d^k - x_{id}^k \right| \tag{24}$$

where x_{id}^{k+1} is the position of ferocious wolf i in the D-dimensional space at the k+1th iteration, x_{id}^k is the position of ferocious wolf i in the D-dimensional space at the kth iteration, $step_b$ is the raiding step, and g_d^k is the position of the lead wolf in the D-dimensional space at the kth iteration.

However, such raiding can be regarded as an open-loop process. If feedback is added to the raiding process, the information exchange between the ferocious wolves, the lead wolf can be increased, and the size of the step can be automatically adjusted. When the odor concentration perceived by the ferocious wolf is approximately equal to the odor concentration perceived by the lead wolf, the step size should decrease to slowly approach the lead wolf. When the odor concentration perceived by the ferocious wolf is significantly different from the odor concentration perceived by the lead wolf, the step size should increase, so that the ferocious wolves can get to the position of the lead wolf rapidly. Equation 24 is improved to obtain equation 25.

$$\begin{cases} x_{id}^{k+1} = x_{id}^{k} + step_{bid}^{k+1} \\ step_{bid}^{k+1} = w \cdot step_{bid}^{k} \cdot (g_d^k - x_{id}^k) / \left| g_d^k - x_{id}^k \right| \end{cases} \tag{25}$$

According to equation 25, the calling behavior is initiated, where w is the odor weight, which lies in the range [0, 1]. Further, w is a self-adaptive value depending on the odor concentration. The calculation formula for w is given by

$$w = \begin{cases} w_{\max} - \frac{(w_{\max} - w_{\min}) \cdot (F^k - F_{\min}^k)}{(F_{avg}^k - F_{\min}^k)}, & F^k \le F_{avg} \\ w_{\max}, F^k > F_{avg} \end{cases} \tag{26}$$

where $w_{\max}$ and $w_{\min}$ is the maximum and minimum values of the scale factor, F^k is the lead wolf prey odor concentration in the Kth iteration, F_{avg} is the average of the current prey odor concentration, and $F_{\min}^k$ is the minimum value of the prey odor concentration in the Kth iteration. If the leader of the prey odor concentration is greater than the average concentration, $w = w_{\max}$. In contrast, we will reduce the value of w according to the formula 26.

Ferocious wolves approach the lead wolf from all directions. In this process, if a ferocious wolf perceives a higher odor concentration than the lead wolf, i.e., $Y_i > Y_{lead}$, then the ferocious wolf will replace the lead wolf and continue the summoning behavior. Otherwise, the raiding will continue until the distance between the ferocious wolves and the lead wolf $d \le d_{near}$. At the same time, the algorithm enters the beleaguering behavior. The calculation of d_{near} is shown in equation 27.

$$d_{near} = \frac{1}{D \cdot \omega} \cdot \sum_{d=1}^{D} |\max_d - \min_d| \tag{27}$$

where d_{near} is the distance between the ferocious wolf and the lead wolf when initiating the beleaguering behavior, and ω is the distance determination factor, whose value directly determines whether the beleaguering behavior is to be entered. The larger the value of ω, the faster the convergence is. However, an excessively large ω will make it very

difficult for the algorithm to enter the beleaguering behavior. Further, $\max_d$ and $\min_d$ are the maximum and minimum values in the D-dimensional variable range respectively.

4) MODIFIED BELEAGUERING BEHAVIOR

When the ferocious wolf is very close to the lead wolf, it means that it is not far from the prey, thus, close beleaguering is needed to capture the prey. It can be assumed that the direction of the lead wolf, which has the highest prey odor concentration, is the prey direction, and that artificial wolves need to advance at small steps to carefully search for the prey odor concentration nearby, i.e., the fitness function value. Equation 28 is used to represent the modified beleaguering behavior of the wolf pack.

$$x_{id}^{k+1} = x_{id}^{k} + \xi^{k} \cdot \lambda \cdot step_c \cdot |G_d^k - x_{id}^k| \tag{28}$$

where x_{id}^k is the position of the k-generation artificial wolf in the D-dimensional space, $step_c$ is the beleaguering step length, and G_d^k is the position of the k-generation lead wolf in the D-dimensional space. Further, λ is a random number in the uniform distribution [-1, 1], and ξ^k is the beleaguering adjustment value. The setting method is as follows:

$$\begin{cases} \xi^0 = M \\ \xi^k = c \cdot \xi^{k-1} \end{cases} \quad c \in [0.9, 0.999], \quad M = 1 \tag{29}$$

In the beleaguering process, if the odor concentration perceived by an artificial wolf is greater than the target odor concentration, then the position of the artificial wolf is updated to the position of the target, otherwise, the target position is not changed.

5) WOLF POPULATION REGENERATION

In nature, only wolves with strong capabilities can survive, weaker wolves will starve to death. In order to avoid falling into local optima, K new artificial wolves will be randomly generated to replace K artificial wolves with the lowest capabilities in the population. The larger the value of K, the greater is the diversity and vitality of the wolf population. However, if K is excessively large, the algorithm will be biased to random search. On the other hand, if K is too small, the algorithm will fall into local optima. Therefore, K should take integer values in the range $[N/2 \times \beta, N/\beta]$, where β is the population regeneration factor.

B. PATH PLANNING STEPS

The deployment platform is the starting point S for path planning. The end point E is predicted as the end of path planning. The segment from the X-axis coordinate of point S to the X-axis coordinate of point E is divided evenly into n parts, yielding points x_i, i=[1,2...n]. The algorithm will find the optimal point (y_i, z_i) in the $x = x_i$ plane, and P_i is the optimal curve between the two planes. The calculation method for the optimal curve is given. The distance from the starting point to the end point is P.

Step 1. Parameter setting: Set the parameters required for algorithm operation, such as the starting point coordinates and end point coordinates of the AUV. Set the number of artificial wolves N, the ratio factor of scout wolves α, the population regeneration factor β, the maximum scouting number T_{max}, the scouting step length $step_a$, the raiding step length $step_b$, the beleaguering step length $step_c$, and the distance determination factor ω. The initialization of the parameters terminates when the maximum number of iterations $K_{\max}$ is set.

Step 2. Generate lead wolf: First, calculate the fitness function value of N artificial wolves, and obtain the prey odor concentration at the locations of the N wolves. Select the artificial wolf with the highest prey odor concentration as the lead wolf and the T artificial wolves with the highest concentrations (except the lead wolf) as the scouting wolves. The remaining artificial wolves are the ferocious wolves. Record the odor concentration and location of each artificial wolf, and proceed to step 3.

Step 3. Scouting behavior: T artificial wolves perform the scouting behavior according to equation 25. When the prey odor concentration of scout wolf i exceeds the odor concentration perceived by the lead wolf, or the maximum scouting number is reached, proceed to the summoning behavior.

Step 4. Summoning behavior: The artificial wolf with the highest odor concentration initiates the summoning behavior as the lead wolf. The ferocious wolves raid the vicinity of the lead wolf in large steps according to equation 28. During the raiding process, if a ferocious wolf perceives a higher odor concentration than the prey odor concentration at the location of the lead wolf, then this ferocious wolf replaces the lead wolf to initiate the summoning behavior. Otherwise, the ferocious wolves continue raiding until the distance from the lead wolf is smaller than the beleaguering distance d_{near}.

Step 5. Beleaguering behavior: After entering the beleaguering behavior, the wolves are very close to the prey. According to equation 28, update the location of the beleaguering artificial wolves.

Step 6. Population regeneration: Update the wolf pack according to the population regeneration principle and initial parameter settings so that the entire population can maintain vitality.

Step 7. Determine the constraints: Determine whether the location of artificial wolf x_{id} is in the equivalent constraint conditions, and whether the points on the segments formed by connecting the adjacent points are outside the bounds of the equivalent constraint conditions. If the conditions are met, then set x_{id} as the location of the artificial wolf, otherwise, take the location of the artificial wolf with the highest prey odor concentration (highest fitness function value) that satisfies the conditions.

Step 8. Determine whether the iteration condition is satisfied: Determine if the maximum number of iterations $K_{\max}$ is reached at this time. If it is reached, the algorithm output the optimal solution, otherwise, return to step 3 to continue the iterative operation.

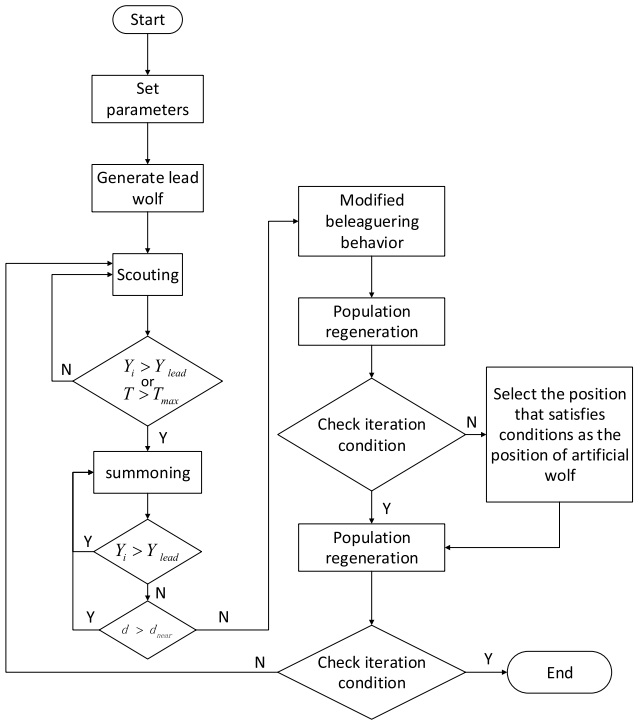


FIGURE 7. The path planning flowchart.

The path planning flowchart is shown in figure 7.

Step 9. The dubins path parameters between each path point are calculated by the description in Section II to generate the dubins path.

In general, the algorithm performs a calculation firstly and generate the lead wolf, then scout wolves scout the prey, and ferocious wolves beleaguering the prey when the lead wolf is summoning. Finally, the wolf pack population is regenerated. The modified WPA enhance the interactive communication of wolf pack by improving the three intelligent behaviors of wolf pack algorithm.

IV. ALGORITHM VERIFICATION AND SIMULATION

The experimental environment consisted of MATLAB® 2013b installed on a computer with Windows®7 Home Basic (64-bit) operating system, Intel®Core™ i3-2330M CPU (2.20 GHZ), and 4 GB RAM.

This simulation is divided into two parts. Firstly, the paper verifies the search ability of wolf pack algorithm (using modified wolf pack algorithm) on high dimensional functions. Compared with other intelligent algorithms in the search results, the paper shows the advantage of wolf pack algorithm in high-dimensional path planning. Secondly, the paper compares the path planning results with wolf pack algorithm and modified wolf pack algorithm, and verifies the advance of the modified wolf pack algorithm.

A. MODIFIED WOLF PACK ALGORITHM VERIFICATION

According to the rules of modified wolf pack algorithm, the procedure of the improved wolf pack algorithm is prepared by MATLAB. The modified wolf pack algorithm is used to solve some complex equations. The selected equations involve multidimensional, multi-peak and other complex situations to judge wolf pack algorithm and other optimization algorithms which is more advantageous. The single peak means that the function has only one extreme value, and the multimodal function has multiple extremes, which leads to the failure of the most optimal search in the algorithm with low global search capability. The multidimensional and multimodal function will be more difficult, and the requirements of the algorithm is even higher. The search function is shown in table 2.

TABLE 2. Search function.

Function	Equations	Dim	Range	Optimal solution
Easorm	$f(x) = -\cos(x_1)\cos(x_2) \cdot \exp\left[-(x_1-\pi)^2-(x_2-\pi)^2\right]$	2	[-100 100]	$f_{\min} = -1$
Trid6	$f(X) = \sum_{i=1}^{D}(x_i-1)^2 - \sum_{i=2}^{D} x_i x_{i-1}$	6	[-36 36]	$f_{\min} = -50$
Sphere	$f(x) = \sum_{i=1}^{D} x_i^2$	30	[-1.5 1.5]	$f_{\min} = 0$
Rastrign	$f(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	60	[-10 10]	$f_{\min} = 0$
Quadric	$f(x) = \sum_{i=1}^{D}(\sum_{k=1}^{i} x_k)^2$	120	[-30 30]	$f_{\min} = 0$

NOTES: The formulas and standard values in the table come from reference 11.

TABLE 3. The parameters of the algorithm.

Algorithm	Parameters
Particle swarm algorithm	Inertia weight $w = 0.7$, learning factor $c_1 = c_2 = 2$, The particle size is 50, the maximum number of iterations is 10.
Genetic algorithm	Crossover probability $p_c = 0.8$, mutation probability, choose the use of roulette way, cross and variation using a random multi-point approach.
Modified wolf pack algorithm	Ratio factor of scout wolves $\alpha = 0.8$, maximum scouting number $T_{\max} = 20$, distance determination factor $w = 500$, population regeneration factor $\beta = 6$

NOTES: The parameters of GA and PSO are derived from the reference 12 and 13.

Using the modified wolf pack algorithm, genetic algorithm, and particle swarm algorithm to optimize the function of 10 times, determines the advantages and performance of each algorithm. The parameters of the algorithm are shown in table 3, and the results are shown in table 4.

And we found that all three of these algorithms are very precise, but for different functions have different characteristics.

TABLE 4. Calculation results.

Function	Algorithm	Optimal	Worst	Average	Average time consuming (s)
Easorm	MWPA	-1	-0.9963	-0,9998	2.8624
	PSO	-1	-1	-1	1.6213
	GA	-1	-1	-1	3.6351
Trid6	MWPA	-49.9999	-49.9961	-49.9995	2.0192
	PSO	-50	-50	-50	1.6011
	GA	-50	-49.9986	-49.9996	7,3417
Sphere	MWPA	0	0	0	0.7224
	PSO	0.1674	0.2843	0.1835	2.5340
	GA	0.2011	0.4269	0.3378	26.6625
Rastrign	MWPA	0	0	0	1.3567
	PSO	22.3114	73.4401	40.8835	2.9841
	GA	35.3936	42.6349	39.1620	56.1134
Quadric	MWPA	0	0	0	2.0326
	PSO	11.0788	59.2433	22.0566	29.5144
	GA	18.1023	77.9319	24.6056	140.7281

NOTES: The parameters of GA and PSO are derived from the reference 14.

It can be found that the relative effect of the particle swarm optimization algorithm is the best among the three algorithms for the single peak and low dimension function. Not only the search precision is very high, but also the calculation time is shorter. Compared with the particle swarm algorithm and the genetic algorithm, the function calculation time of the modified wolf pack algorithm does not show its absolute advantage. However, according to the data in the table, with the increase of the dimension of the calculation function, the difficulty of the calculation increases, and the advantage of modified wolf pack algorithm is emerging. In the Sphere function, only the particle swarm algorithm finds the nearest minimum, and the results obtained by the genetic algorithm can be considered to be failed. Continue to increase the function dimension to 120 dimensions, in addition to the modified wolf pack algorithm, other algorithms all failed, and the simulation time is very long. The result shows that the modified wolf pack algorithm has advantages in the high-precision, high-dimensional, multi-peak function.

Through the average value, we can find that the calculation error of the modified wolf pack algorithm is very small, and there is an unparalleled advantage for the robustness comparison of genetic algorithm and particle swarm algorithm for different functions.

In the case of low priority, the search time of particle swarm algorithm is the shortest, the search time of genetic algorithm is the slowest, and the time of modified wolf pack algorithm is centered in three algorithms. However, with the increase of the dimension, the simulation time of the particle swarm optimization algorithm is relatively increased due to the decrease of the optimization effect, and it is very likely to fall into the local optimal solution. The simulation time and the optimization result of the genetic algorithm is very balanced.

Therefore, we can conclude that the modified wolf pack algorithm has stronger robustness and stability compared with particle swarm algorithm and genetic algorithm. The search time of low dimension is not prominent, but the result is satisfactory, and it is especially suitable for high dimension, multi-peak complex function, not easy to fall into the local optimal solution, and the modified wolf pack algorithm has a good global search capability.

B. COMPARISON BETWEEN WOLF PACK ALGORITHM AND MODIFIED WOLF PACK ALGORITHM

The simulation environment is the equivalent threat shown in figure 2. The environment space is a $50km \times 50km \times 50km$ underwater space. The deployment platform is (3, 15, 40), the predetermined destination position is set to (30, 10, 10), and the X-axis segment between the deployment platform and the destination is divided into 31 segments, i.e., n = 30. In the course of navigation, the algorithm primarily follows the main environment, i.e., it avoided the obstacles.

The main parameters of the modified wolf pack algorithm are set as follows: the number of artificial wolves N=300, the maximum number of iterations $K_{\max} = 300$, the maximum scouting number $T_{\max} = 30$, the distance determination factor $\omega = 50$, the scouting step length $step_a = 0.2$, the raiding step length $step_b = 0.4$, the beleaguering step length $step_c = 0.1$, the ratio factor of scout wolves $\alpha = 4$, and the population regeneration factor $\beta = 5$.

The simulation results are shown in figure 8 and figure 9.

As shown in figure 9, the convergence curve of the fitness function tends to stabilize after 150 iterations, and the minimum value is reached after 220 iterations. The final optimization results are shown in the table 5.

The advantages of the modified wolf pack algorithm are verified by comparing the algorithm with the original wolf pack algorithm when solving the path planning problem. The parameters are consistent for both algorithms, and the results of path planning are shown in figure 10.

Compared to the figure 8, it can be seen from figure 10 that original wolf pack algorithm has obvious bending and corner, indicating that the wolf pack algorithm in the final calculation did not find the optimal solution, the final part of the calculation is still not convergence. Therefore, the modified wolf pack algorithm curve is smoother, and AUV sport control is friendlier.

The comparison of path planning using the modified wolf pack algorithm and the original wolf pack algorithm are shown in table 5.

It can be seen from the simulation that both the modified wolf pack algorithm and the original wolf pack algo-

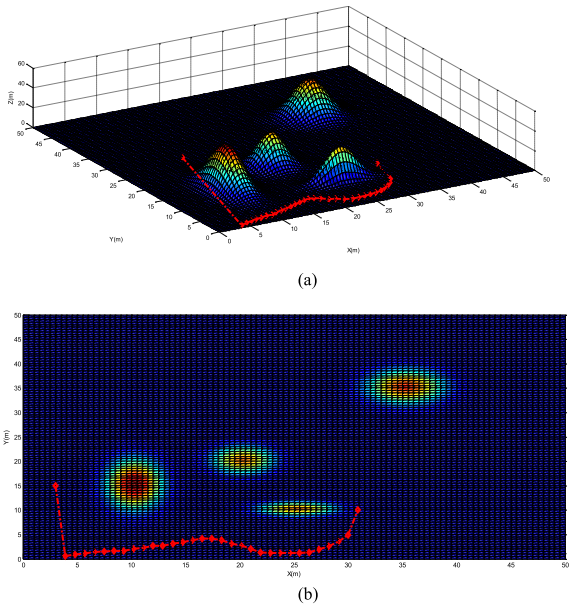


FIGURE 8. (a) Path-planning simulation diagram based on modified wolf pack algorithm. (b) Path-planning simulation diagram based on modified wolf pack algorithm.

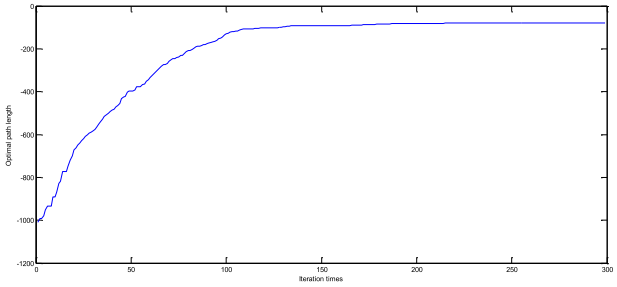


FIGURE 9. Convergence curve of fitness function.

TABLE 5. Performance comparison between modified wolf pack algorithm and original wolf pack algorithm.

	Modified WPA	WPA
Path length (m)	72.2521	77.0736
Simulation time (s)	4.3712	10.2408

rithm can effectively avoid collision with threats and complete the task using the shortest path possible. However, as shown in table 5, the modified wolf pack algorithm outperforms the original wolf pack algorithm in terms of path length and convergence speed. The modified wolf pack algorithm finds the shorter path length with less time consumed, and the results mean that it will be more energy-efficient for AUV.

The artificial wolves can not only perceive information from their companions but also transmit prey information, which allows for effective control of the balance between global searching and local searching. In addition, the raiding step length is modified from a fixed value to an adaptive value to improve the global prey searching ability, preventing

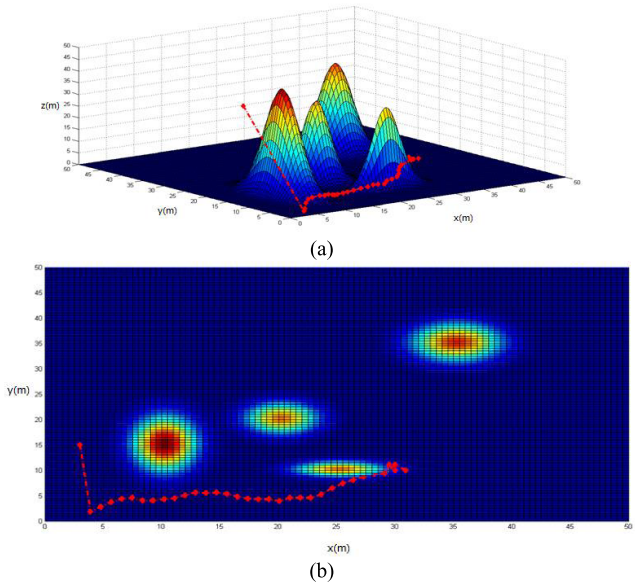


FIGURE 10. (a) Path planning diagram of original wolf pack algorithm. (b) Path planning diagram of original wolf pack algorithm.

the algorithm from falling into local optima. For the reasons given above, the modified wolf pack algorithm has faster rate of convergence and higher convergence precision.

V. CONCLUSIONS

Starting from a real-life application, this paper aimed to develop an easy-to-implement path planning method for underwater resource exploration. First, the seabed environment is equated to the threat model and it serves as a constraint function along with the seabed depth. Then, the optimal path is obtained by solving the fitness function using the modified wolf pack algorithm. The simulation results show that the modified wolf pack algorithm has high rate of convergence and good local search capability in the high-precision, high-dimensional, multi-peak function. Moreover, it does not converge prematurely. Artificial wolves can not only perceive information from their companions but also transmit prey information, which allows for effective control of the balance between global searching and local searching. In addition, the raiding step length is modified from a fixed value to an adaptive value to improve the global prey searching ability, preventing the algorithm from falling into local optima. In order to meet constraints of AUV, a fast path planning method based on dubins path is proposed, which applies the dubins path planning to meet angle control constraint and tune the turning radius to meet control constraint. However, some problems remain to be solved in the modified wolf pack algorithm. Specifically, there are too many parameters that need to be set in the modified wolf pack algorithm. The setting of some parameters affects the convergence speed and the accuracy of the results significantly, and improper parameter setting may result in low accuracy or convergence that is too fast or too slow.

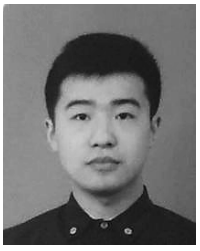
Therefore, in the future we plan to improve the parameter setting of the algorithm and change the current state in which the parameters are difficult to set. Furthermore, the wolf pack algorithm has excellent engineering prospects. The wolf pack algorithm can solve complex multidimensional functions and it does not fall into local optima easily, hence, various engineering applications of wolf pack algorithm can be optimized, e.g., by solving the controller parameters or normalization coefficient. Finally, duo to the wolf pack algorithm is still a relatively new algorithm, there is relatively few applications, and the algorithm is considerably extensible. Through additional research, the calculation accuracy and efficiency of wolf pack algorithm can be enhanced further by using some reduction of space representation such as quadtree, octree, and rectangular symmetry reduction etc.

REFERENCES

[1] C. Kalogeri *et al.*, "Assessing the European offshore wind and wave energy resource for combined exploitation," *Renew. Energy*, vol. 101, pp. 244–264, Feb. 2017.
[2] V. Ganesan, M. Chitre, and E. Brekke, "Robust underwater obstacle detection and collision avoidance," *Auto. Robots*, vol. 39, no. 7, pp. 1130–1146, 2015.
[3] R. Dai and J. Cochran, "Path planning and state estimation for unmanned aerial vehicles in hostile environments," *J. Guid. Control, Dyn.*, vol. 33, no. 2, pp. 595–601, 2010.
[4] A. Bircher, M. Kamel, K. Alexis, M. Burri, and P. Oettershagen, "Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots," *Auto. Robots*, vol. 40, no. 6, pp. 1–20, 2015.
[5] Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, "A survey on path planning for persistent autonomy of autonomous underwater vehicles," *Ocean Eng.*, vol. 110, pp. 303–313, Dec. 2015.
[6] J. Cao, J. Cao, Z. Zeng, and L. Lian, "Optimal path planning of underwater glider in 3D Dubins motion with minimal energy consumption," in *Proc. IEEE OCEANS*, Shanghai, China, Apr. 2016, pp. 1–7.
[7] D. Dong, B. He, Y. Liu, R. Nian, and T. Yan, "A novel path planning method based on extreme learning machine for autonomous underwater vehicle," in *Proc. IEEE OCEANS*, Washington, DC, USA, Oct. 2015, pp. 1–7.
[8] U. Gautam, R. Malmathanraj, and C. Srivastav, "Simulation for path planning of autonomous underwater vehicle using flower pollination algorithm, genetic algorithm and Q-learning," in *Proc. IEEE Int. Conf. Cognit. Comput. Inf. Process. (CCIP)*, Noida, India, Mar. 2015, pp. 1–5.
[9] P. F. Felzenszwalb and D. Mcallester, "The generalized A* architecture," *J. Artif. Intell. Res.*, vol. 29, no. 1, pp. 153–190, 2011.
[10] D. Harabor and A. Grastien, "The JPS pathfinding system," in *Proc. 5th Annu. Symp. Combinat. Search*, 2012, pp. 207–208.
[11] D. Ortiz-Boyer, C. Hervá-Martínez, and N. García-Pedrajas, "CIXL2: A crossover operator for evolutionary algorithms based on population features," *J. Artif. Intell. Res.*, vol. 24, no. 1, pp. 1–48, 2005.
[12] J. Kennedy and R. Eberhart, "Particle swarm optimaization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. Aug. 2002, pp. 1942–1948.
[13] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, Jun. 1994.
[14] H. Wu, F. Zhang, and L. Wu, "New swarm intelligence algorithm-wolf pack algorithm," *J. Syst. Eng., Electron.*, vol. 35, no. 11, pp. 2430–2438, 2013.
[15] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an AUV," *Ocean Eng.*, vol. 34, nos. 11–12, pp. 1734–1744, Aug. 2007.
[16] H. Wu and F. Zhang, "A uncultivated wolf pack algorithm for high-dimensional functions and its application in parameters optimization of PID controller," in *Proc. IEEE Evol. Comput.*, Beijing, China, Jul. 2014, pp. 1477–1482.
[17] Y. Liang, Y. A. Zhang, and J. W. Lei, "New method of online fast path planning based Dubins path," *J. Syst. Simul.*, vol. 25, pp. 291–296, 2013, doi: 10.16182/j.cnki.joss.2013.s1.030.
[18] H. S. Wu and F. M. Zhang, "Wolf pack algorithm for unconstrained global optimization," *Math. Problems Eng.*, vol. 2014, no. 1, pp. 1–17, 2014.
[19] H. S. Wu, F. M. Zhang, H. Li, and X. L. Liang, "Discrete wolf pack algorithm for traveling salesman problem," *Control Decision*, vol. 30, no. 10, pp. 1861–1867, 2015.
[20] A. Lissovoi and C. Witt, "Runtime analysis of ant colony optimization on dynamic shortest path problems" *Theor. Comput. Sci.*, vol. 561, pp. 73–85, Jan. 2015.
[21] H. Li and H. Wu, "An oppositional wolf pack algorithm for parameter identification of the chaotic systems," *Opt. Int. J. Light Electron Opt.*, vol. 127, no. 20, pp. 9853–9864, 2016.
[22] H. Wu, F. M. Zhang, R. J. Zhan, L. I. Hao, and X. L. Liang, "Improved binary wolf pack algorithm for solving multidimensional knapsack problem," *Syst. Eng. Electron.*, vol. 37, no. 5, pp. 1084–1091, 2015.
[23] Y. Chen, Z. Wang, E. Yang, and Y. Li, "Pareto-optimality solution recommendation using a multi-objective artificial wolf-pack algorithm," in *Proc. Int. Conf. Softw., Knowl., Inf. Manage., Appl.*, 2016, pp. 116–121.
[24] Y. Li, T. Ma, P. Chen, Y. Jiang, R. Wang, and Q. Zhang, "Autonomous underwater vehicle optimal path planning method for seabed terrain matching navigation," *Ocean Eng.*, vol. 133, pp. 107–115, Mar. 2017.
[25] R. Song, Y. Liu, and R. Bucknall, "A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment," *Ocean Eng.*, vol. 129, pp. 301–317, Jan. 2017.

LANYONG ZHANG (M'15) received the Ph.D. degree in theory and control engineering from Harbin Engineering University in 2011. His current research interests include the fast transient analysis and modeling of field-excited multiconductor networks, power-line carrier propagation, electromagnetic field interference from overhead multiconductor lines, and electromagnetic interaction with advanced composite materials. In 2009, he received the Excellent Graduate Award.

He is currently a Master's Supervisor with the Harbin Engineering University. His main research interests are electromagnetic compatibility prediction and measurement, and stochastic signal processing.



LEI ZHANG (M'17) was born in Dezhou, China, in 1992. He received the B.S. degree in automation from the Shandong University of Science and Technology. He is currently pursuing the M.S. degree in control science and engineering from Harbin Engineering University, Harbin, China. He is a volunteer of the IEEE Computer Society in 2017.

SHENG LIU was born in Baicheng, China, in 1957. He received the B.Eng. degree in industrial automation from the Harbin University of Civil Engineering and Architecture, Harbin, China, in 1982, and the master's and Ph.D. degrees in theory and control engineering from Harbin Engineering University, Harbin, in 1982 and 2000, respectively. He is currently a Professor with Harbin Engineering University. His current research interests include electromagnetic compatibility prediction and measurement, optimization estimate and control of random system, robust control, and ship control system.

JIAJIA ZHOU received the Ph.D. degree in control theory and control engineering from Harbin Engineering University, China, in 2012. He is currently a Lecturer with Harbin Engineering University. His research interests include control and navigation for underwater vehicles.

CHRISTOS PAPAVASSILIOU (M'96–SM'05) was born in Athens, Greece, in 1960. He received the B.Sc. degree in physics from the Massachusetts Institute of Technology, Cambridge, MA, USA, and the Ph.D. degree in applied physics from Yale University, New Haven, CT, USA.

He was with the Foundation for Research and Technology, Hellas, Crete, Greece, where he was involved in GaAs monolithic microwave integrated circuit design and measurements and several European and regional projects on GaAs MMIC technology. Since 1996, he has been with the Imperial College London, where he is currently involved in SiGe technology development, RF IC, and instrumentation. He has contributed to over 70 publications. His current research interests include memristor applications, electromagnetic surface wave propagation on interfaces, and antenna.