

Research Article

Cooperative Search and Rescue with Artificial Fishes Based on Fish-Swarm Algorithm for Underwater Wireless Sensor Networks

Wei Zhao, Zhenmin Tang, Yuwang Yang, Lei Wang, and Shaohua Lan

Computer Department, Nanjing University of Science and Technology, Jiangsu 210094, China

Correspondence should be addressed to Yuwang Yang; yang-yuwang@163.com

Received 23 December 2013; Accepted 29 January 2014; Published 5 March 2014

Academic Editors: Y. Mao, X. Meng, and Z. Zhou

Copyright © 2014 Wei Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a searching control approach for cooperating mobile sensor networks. We use a density function to represent the frequency of distress signals issued by victims. The mobile nodes' moving in mission space is similar to the behaviors of fish-swarm in water. So, we take the mobile node as artificial fish node and define its operations by a probabilistic model over a limited range. A fish-swarm based algorithm is designed requiring local information at each fish node and maximizing the joint detection probabilities of distress signals. Optimization of formation is also considered for the searching control approach and is optimized by fish-swarm algorithm. Simulation results include two schemes: preset route and random walks, and it is showed that the control scheme has adaptive and effective properties.

1. Introduction

Recently, with the development of wireless sensor networks, especially the mobile wireless sensor networks, more and more applications emerge and have received general attention. Examples include pollution detection, wildfire monitoring, search and rescue missions, reconnaissance, and surveillance. In this paper, we position the application background at the problem of underwater search and rescue [1–3]. As we all know, underwater search and rescue is a very dangerous activity which requires professionals to conduct, and the rescue workers generally need to receive professional training to do this job. The primary task of search and rescue is to find and position the victim, and due to the complicated underwater environment, this is a very difficult job. With the development of bionics, the emergence of artificial fish [4–6] maybe has provided us with a cooperative search and rescue plan. We can install the searching device onto the artificial fish, and when the fish find the victim, they will send the position signal to the rescuing ship. In addition, through coordinated searching by multiple artificial fishes, the probability to find the victim can be significantly increased, and in the meantime, the searching time can be shortened, which is vital in rescuing tasks. This paper studies

the problem of how to coordinate multiple artificial fishes and conduct efficient and timely search, and it will be a meaningful task to discuss this problem.

The problem proposed in this paper is similar to the fundamental problem of cooperative coverage control or active sensing [7–10]. Cooperative control refers to settings which involve multiple controllable agents cooperating toward a common objective. In [7], the authors propose a coverage control algorithm aimed at maximizing target exposure in some surveillance applications, while in [11–14], heuristic algorithms based on potential fields and virtual forces are applied to push nodes away from each other and disperse them into the unoccupied areas in the mission space to enhance the coverage of a sensor network. The potential fields and virtual forces approach imitates the behavior of electromagnetic particles: when two electromagnetic particles are too close in proximity, a repulsive force pushes them apart. Applied to a sensor network, this method helps move sensors from high-density to low-density areas, thereby minimizing sensing overlap and improving the overall network coverage. In [8], a decentralized coverage control algorithm is proposed based on centroidal Voronoi partitioning, and a dynamic version of the Lloyd algorithm [15] has been used to iteratively find such a configuration. Other related works based on

Voronoi's partitions are included in [16, 17]. However, Lloyd's method suffers from two critical issues when it is used in mobile sensor networks. First, it does not consider the limited sensor communication range. Secondly, it does not optimize sensor movement distance; hence, it can lead to excessive energy consumption, a primary concern in sensor networks. Much of the active sensing literature [10] also concentrates on the problem of tracking specific targets using mobile sensors and the Kalman filter is extensively used to process observations and generate estimates.

In this paper, we consider a setting which involves a team of fish nodes and a set of target points (victims) in a three-dimensional space (e.g., under water). Each target point represents a victim. A mission is defined as the process of controlling the movement of the fish nodes and ultimately assigning them to target points so as to find the victims by visiting points within a given mission time T . In [18], the authors consider a setting where multiple vehicles form a team cooperating to visit multiple target points and collect rewards associated with them. Related work was proposed in their papers [19, 20]. Different from the papers mentioned above, in this paper, the target points are totally unknown in advance, or just located in an approximate range area. In addition, we pay more attention to find the target points in the shortest time. Because the earlier the victim can be found, the less the losses there will be; thus, we propose a cooperative coverage control scheme aiming at finding the target points in the shortest time under an uncertain environment.

The rest part of this paper is organized as follows: in Section 2, we propose the distributed cooperative searching control scheme for underwater mobile sensor networks. In Section 3, we optimize the scheme mentioned in Section 2 by fish-swarm algorithm. In Section 4, we simulate the distributed cooperative searching control algorithm by using computer software and evaluate its performance. Finally, in Section 5, we reach the main conclusions.

2. Fish-Swarm Searching Trajectory Model

We imagine a classic scenario: consider Area A with $C(x, y)$ as its center of circle, and it has a radius of R and a depth of d . This could be a case of accident in the ocean, the accident center is the circle center C , the survivors may be scattered in Area A with a radius of R , but their specific locations are unknown, and the task objective is to find all the survivors in Area A with no omission, as shown in Figure 1.

Under the actual situation, the survivors tend to be scattered around the accident center, and the further from the center, the less the survivors there will be. In this paper, the mobile nodes participating in the search task are assumed as the fish swarm, and based on the situation mentioned above, the following two schemes are considered: in one scheme, after the fish swarm (mobile nodes) reaches the accident center, random walk is adopted; in another scheme, cruise is conducted in accordance with a certain preset route. The so-called random walk refers to that the fish swarm moves

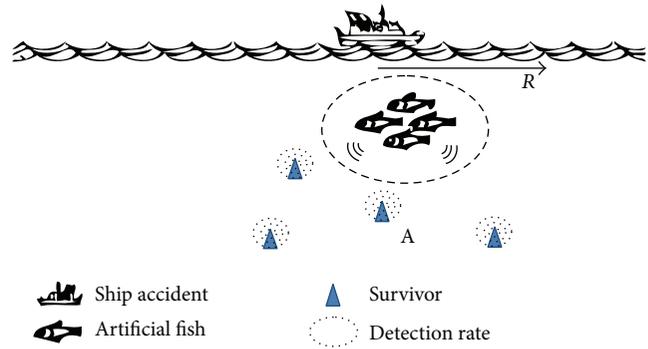


FIGURE 1: Task scenario description.

toward a certain direction; when they reach the task boundary, they randomly choose another direction and continue swimming, but the whole swimming track has to be within the task Area A . Here, we will discuss the second scheme, and in the experiment part of this paper, the effects of the two schemes will be compared. Assume the trajectory equation of fish swarm's cruise is $F(r, \theta, z, t)$, which represents that at any moment t , the spatial position of fish swarm is (r, θ, z) , and Figure 2 has listed the corresponding helixes of several alternative helix equations. Of course, this is only an assumption, any equation will work, and the only difference is the effect.

In accordance with Figure 2, we can see that none of these helix curves satisfies the searching requirement; that is, there is a blind spot in exploration. An ideal track of fish swarm should satisfy the following several conditions. Firstly, it should cover the whole Area A . Secondly, it only needs to search any position in the area once; that is, there is repeated area, and in this way, it can ensure the shortest searching time. Thirdly, the trajectory equation should not be too complicated, which will sabotage the mechanical realization of robot fish's route. Therefore, in accordance with the above conditions, this paper constructs an applicable helix equation, which can be expressed as the following in a polar coordinate system:

$$F(r, \theta, z, t) = \begin{cases} r = \frac{R}{T}(t - 2nT), \\ 2nT \leq t < (2n + 1)T, \quad n = 0, 1, 2, \dots, \\ r = -\frac{R}{T}(t - 2nT - T) + R, \\ (2n + 1)T \leq t < (2n + 2)T, \quad n = 0, 1, \dots, \\ \theta = \theta_0 + \frac{R}{2R_s}2\pi t, \\ z = z_0 - 2R_s m, \\ mT \leq t \leq (m + 1)T, \quad m = 0, 1, 2, \dots \end{cases} \quad (1)$$

In Formula (1), R refers the search radius of Area A , T is a search cycle, the whole search process consists of multiple search cycles T , R_s is the perceived radius of node, θ_0 and z_0 represent the angle and the initial position of axis z , respectively, and m and n represent integers. In accordance

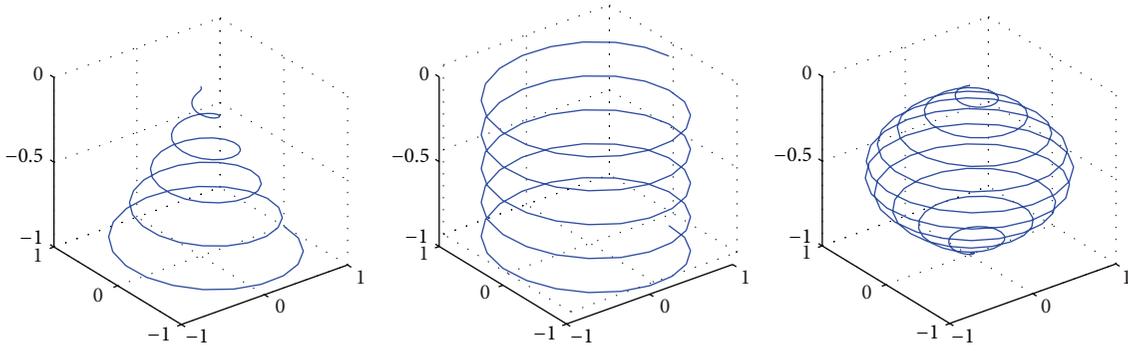


FIGURE 2: Helix curve.

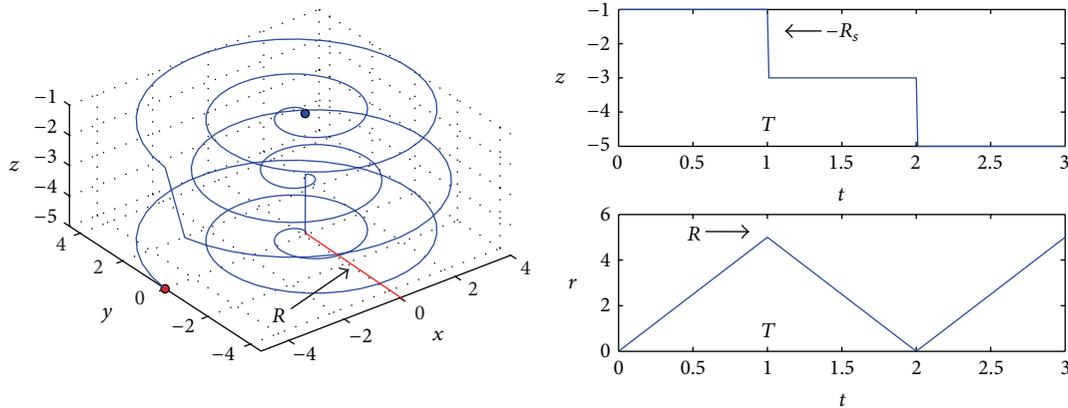


FIGURE 3: Helix constructed in this paper.

with the left diagram in Figure 3, the node moves from the initial position (blue dot) to the final position (red dot) through the helix, during which, 3 cycles (T) of search have been conducted, the search of each cycle represents a plane on axis z , and it gets deeper and deeper. Therefore, the trajectory can cover the whole Area A , and there is no repeated path.

In accordance with the above right diagram in Figure 3, this is the curve of the search depth of fish swarm changing with time, the initial altitude of search is decided by z_0 , and here, $z_0 = -R_s$, which can just cover the sea surface. After a search cycle T is completed, the fish swarm needs to search in a deeper position; that is, $z_0 - 2R_s$, and in this way, it can ensure that there will not be any omission or repeated search area. We call the search plane at the same altitude completed in each cycle T a search layer, and the whole search process consists of multiple such search layers. The lower part of the right diagram in Figure 3 is the curve of search radius r changing with time. The initial position is at the accident center, so $r = 0$; as the fish swarm diffuses to the surrounding area, r reaches the maximum search radius R in a search layer; then, it will conduct backward search in the next search later, and r gradually reduces from R to 0. The benefits of trajectory equation with such design include full coverage of Area A and there is no area with repeated search, and in the meantime, the fish swarm can smoothly transit between various search layers.

3. Optimization Formation with Fish-Swarm Algorithm

If only one artificial fish participates into the search task, there is no need for formation optimization. Of course, during the actual process, in order to increase the searching scale and shorten the searching time, it is more reasonable to adopt collaborative search. Therefore, it will generate the formation optimization problem of fish swarm. Apparently, the formation of searching fish swarm should satisfy the following two requirements: firstly, the search section of fish swarm should be as big as possible; secondly, the fish swarm should maintain connection; that is, they should be within the communication range. Naturally, it is appropriate for us to use the artificial fish swarm optimization algorithm (AFSA) [21, 22] to optimize the formation of fish swarm.

3.1. Network Coverage. First of all, let us discuss the calculation of network coverage. As an important index to measure the strategy of sensor network deployment, the network coverage is generally defined as the ratio between the whole area that can be covered by nodes in the monitored area and the total monitored area. Considering the complication of monitoring environment in actual application, this paper has adopted the probability measurement model in the literature

[23] to calculate the network coverage. Assume the total number of nodes is N , s_i representing the i th node in the network; then, the corresponding node set is $S = \{s_i \mid i = 1, 2, \dots, N\}$. Assume $p(x, y, z)$ is a random point in Area A , $p \in A$; then the distance between node $s_i(x_i, y_i, z_i)$ and point P is

$$D(s_i, p) = \|s_i - p\| = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}. \quad (2)$$

By adopting the probability measurement model in the literature [23], the detection probability of node s_i to point p is

$$P_p(s_i) = \begin{cases} 0, & R_s + R_e \leq D(s_i, p) \\ e^{-\alpha\lambda^\beta}, & R_s - R_e < D(s_i, p) < R_s + R_e \\ 1, & R_s - R_e \geq D(s_i, p), \end{cases} \quad (3)$$

in which R_s refers to the perceived radius of various nodes in the network, R_e refers to the uncertain factors within the measurement range of nodes, and $0 < R_e < R_s$; α and β refer to the measured parameters related to the physical device; λ is the input parameter, which is defined as

$$\lambda = D(s_i, p) - (R_s - R_e). \quad (4)$$

Therefore, we can obtain the joint detection probability of multiple sensor nodes simultaneously conducting measurement to the target point p as

$$P_p(S) = 1 - \prod_{s_i \in S} (1 - P_p(s_i)), \quad (5)$$

in which S refers to the sensor node set of the measured target point. Therefore, in order to realize full coverage of the target area, it requires satisfying the following condition:

$$\min \{P_p(S)\} \geq c_{th}, \quad (6)$$

in which c_{th} ($0 < c_{th} < 1$) refers to the threshold value of detection probability set in accordance with different application requirements. In order to calculate the coverage of sensor network, we need to conduct grid processing of the monitored area (the bigger the distance between adjacent grid points is, the higher the calculation accuracy is), and then, the joint detection probability at each grid point is solved. The percentage of grids which meet $P \geq c_{th}$ is called network coverage.

As shown in Figure 4, Area A with grids is the monitored area, and the shadow areas $U1$ and $U2$ represent the perceived areas of s_1 and s_2 , respectively. Assume Area A is divided into N grids, and p refers to a random grid; then its network coverage is

$$\frac{N \{ \min \{ P_{p \in (A \cap (U1 \cup U2))} (s_1, s_2) \} \geq c_{th} \}}{N}. \quad (7)$$

In Formula (7), the numerator refers to the number of grids that satisfy the threshold value of detection probability c_{th} , while the denominator refers to the total number of grids.

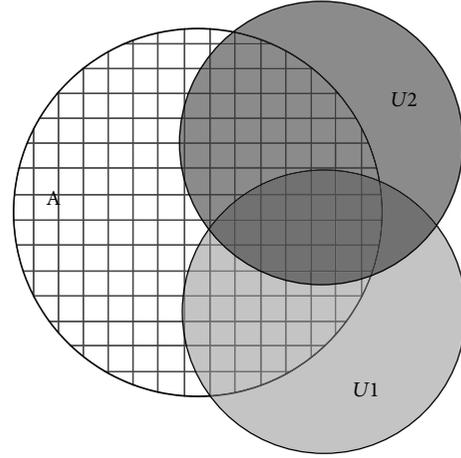


FIGURE 4: Schematic diagram of network coverage.

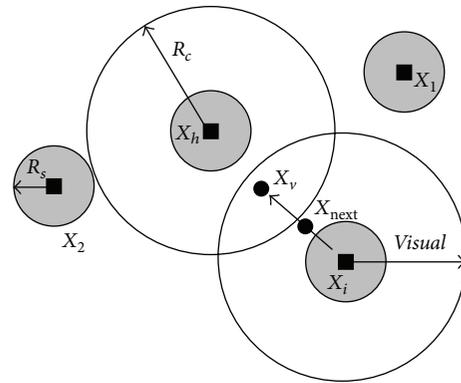


FIGURE 5: Vision concept of the artificial fish.

Because our task model is in a 3-dimensional space, the calculation process is conducted in accordance with the 3D grids, and Figure 5 only shows an example.

Assume under current state, the position vector of all sensor nodes in the network is $U = [X_n \ Y_n \ Z_n]$, and the function to calculate the network coverage with U as the input variable is $f(U)$, in which $X_n = \{x_1, x_2, \dots, x_n\}$, $Y_n = \{y_1, y_2, \dots, y_n\}$, and $Z_n = \{z_1, z_2, \dots, z_n\}$ are the node coordinate vectors. We abstract the optimization of network layout into solving the optimization problem with $f(U)$ as the objective function.

3.2. Optimization with Fish-Swarm Algorithm. The artificial fish-swarm optimization algorithm (AFSA) is an optimization algorithm which simulates the behavior of fish swarm, and it uses the preying, gathering, and chasing behavior of fish swarm to find fast and global optimum solution. In this paper, the fish swarm consisting of mobile nodes can also be regarded as a cluster of nodes, in which the cluster head node (the leader of fish swarm) conducts cruise in accordance with the trajectory equation $F(r, \theta, z, t)$ mentioned in the last section; other nodes are processed in accordance with the fish swarm algorithm, and in this way, it can ensure that the

whole fish swarm maintains reliable communication and a maximum search scale.

Assume in an n -dimensional target search space, there is a fish swarm consisting of N artificial fish, and every day, the state of individual artificial fish can be expressed as vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$, in which x_i ($i = 1, \dots, n$) is the variable that needs to be optimized: the food concentration of current location where the artificial fish are can be expressed as $Y = f(\mathbf{X})$, in which Y is the objective function; the distance between individual artificial fish can be expressed as $d_{ij} = \|\mathbf{X}_i - \mathbf{X}_j\|$; visual refers to the perceived range of artificial fish, step is the moving step length of artificial fish, and δ is the crowding degree factor; try_number represents the maximum trying times each time the artificial fish go preying.

The AF realizes external perception by its vision shown in Figure 5. X_i is the current position of an AF, X_h is the cluster head's position, $Visual$ (R_c , communication radius) is the visual distance, and X_v is the visual position at some moment. If the position at the visual position is better than the current position, it goes forward a step in this direction, and arrives at the X_{next} position; otherwise, it continues an inspecting tour in the vision.

In the sensor network, the process of mobile nodes exploring toward bigger network coverage is similar to the chasing and preying behavior by individual artificial fish, and the food concentration of current location where the artificial fish are can be regarded as the network coverage under current state. Fish usually stay in the place with a lot of food, so we simulate the behaviors of fish based on this characteristic to find the global optimum, which is the basic idea of the AFSA. The basic behaviors of AF are defined as follows.

(1) *AF_Prey*. This is a basic biological behavior heading for the food; generally the fish perceives the concentration of food in water to determine the movement by vision or sense and then chooses the tendency.

Behavior description: let X_i be the AF current state and select a state X_j randomly in its visual distance, Y is the food concentration (objective function value), the greater the $Visual$ is, the more easily the AF finds the global extreme value and converges:

$$\mathbf{X}_j = \mathbf{X}_i + Visual.Rand(). \quad (8)$$

If $Y_i < Y_j$ in the maximum problem, it goes forward a step in this direction; otherwise, select a state X_j randomly again and judge whether it satisfies the forward condition. If it cannot satisfy after try_number times, it moves a step randomly. When the try_number is small in AF_Prey, the AF can swim randomly, which makes it flee from the local extreme value field:

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + Visual.Rand(). \quad (9)$$

(2) *AF_Swarm*. The fish will assemble in groups naturally in the moving process, which is a kind of living habits in order to guarantee the existence of the colony and avoid dangers. Behavior description: let X_i be the AF current state, X_c the center position, and nf the number of its companions in the current neighborhood ($d_{ij} < Visual$), N is the number of

total fishes. If $Y_c > Y_i$ and $nf/N < \delta$, which means that the companion center has more food (higher fitness function value) and is not very crowded, it goes forward a step to the companion center:

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + \frac{\mathbf{X}_c - \mathbf{X}_i^{(t)}}{\|\mathbf{X}_c - \mathbf{X}_i^{(t)}\|} Step.Rand(). \quad (10)$$

Otherwise, it executes the preying behavior. The crowd factor limits the scale of swarms, and more AF only cluster at the optimal area, which ensures that AF moves to optimum in a wide field.

(3) *AF_Follow*. In the moving process of the fish swarm, when a single fish or several ones find food, the neighborhood partners will trail and reach the food quickly. Behavior description: let X_i be the AF current state, and it explores the companion X_j in the neighborhood ($d_{ij} < Visual$), which has the greatest Y_j . If $Y_j > Y_i$ and $nf/N < \delta$, which means that the companion X_j state has higher food concentration (higher fitness function value) and the surroundings is not very crowded, it goes forward a step to the companion X_j :

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + \frac{\mathbf{X}_j - \mathbf{X}_i^{(t)}}{\|\mathbf{X}_j - \mathbf{X}_i^{(t)}\|} Step.Rand(). \quad (11)$$

Otherwise, it executes the preying behavior.

(4) *AF_Move*. Fish swim randomly in water; in fact, they are seeking food or companions in larger ranges. Behavior description: it chooses a state at random in the vision; then it moves towards this state; in fact, it is a default behavior of AF_Prey:

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + Visual.Rand(). \quad (12)$$

Therefore, by using the gathering and chasing behavior of fish swarm, it can draw the nodes close to the cluster head. In the meantime, in order to realize maximum search range, the fish swarm should maintain a good formation, which could be controlled through the crowding factor δ and the distance between fishes. The formation optimization process based on artificial fish swarm consists of the following specific steps.

- (a) Initialize the wireless sensor network. Ensure the scale of artificial fish swarm N in accordance with the application requirement, the maximum moving step length of artificial fish is $step$, the visible range of artificial fish is $visual$, the maximum iteration times are k , and the crowding degree factor is δ .
- (b) Initialize the fish swarm X , randomly generate N individual artificial fish within the task space, and in the meantime, set the initial iteration times as $k = 0$.
- (c) Calculate the food concentration of current location where the initial individual fish in the swarm is Y_i (i.e., the network coverage Y_i); then put them in sequence, and select the individual artificial fish with the biggest value of Y_i to enter the billboard T .

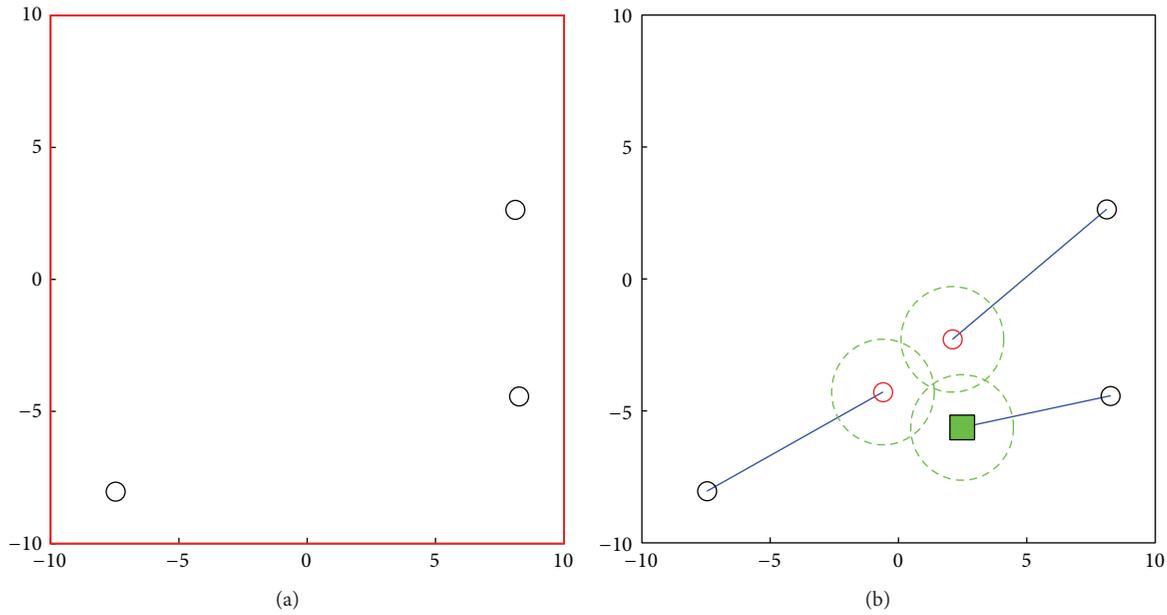


FIGURE 6: Example of 3 fishes forming a swarm.

- (d) The artificial fish simulate the gathering and chasing behavior of fish swarm, and the fish with big Y value is selected to conduct preying behavior.
- (e) After each action of various artificial fish, the food concentration of current location Y will be compared to the Y_T of artificial fish on the billboard, and if it is bigger than the value of Y_T on the Billboard, this artificial fish will replace the billboard fish and enter billboard.
- (f) Determine the end condition. If it has reached the maximum iteration times, the Y value of billboard will be output, that is, the optimum formation solution; otherwise, $k = k + 1$, go to (d).

3.3. *Example for Optimization Formation.* How to make the fish swarm search in a bigger scale? An easy approach is to make all fish in the fish swarm stay on the same plane, this plane is vertical to the direction which this fish swarm moves toward, and this could realize maximization of the cross-section area. In the meantime, efforts should be made to avoid any gap in the middle of fish swarm because the gap might cause blind spot during the search. Take a search fish swarm of 3 fishes for example, and after the 3 fishes reach the accident center, respectively, the formation process will successively begin. The initial positions of the 3 fishes could be random, they are optimized in accordance with the optimization algorithm in Section 3.2, they come closer to form a fish swarm, in the meantime, the distance between individuals is controlled to avoid any gap, and Figure 6 has shown this process.

The small circles on the left diagram represent the initial positions of 3 fishes, the red small circles on the right diagram represent the final positions after forming a swarm, the blue solid line refers to the trace of moving from the initial position

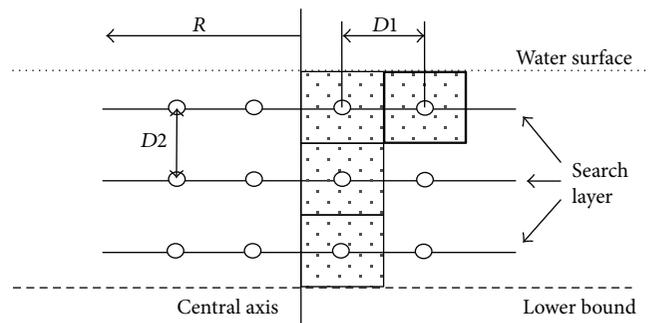


FIGURE 7: Diagrammatic sketch of search cross-section.

to the final position, and the green block represents the leader of fish swarm after formation (cluster head node).

Apparently, if the search is conducted in accordance with the formation in Figure 6, it might cause blind spot during search because it is very difficult to ensure that there is no gap between various search layers. If there is a fish swarm consisting of N fish, for the convenience of not leaving any gap between various search layers or between pitches, as shown in Figure 7, we need to calculate the maximum rectangular cross-sectional area for searching the fish swarm.

In Figure 7, R represents the radius of search area A , $D1$ refers to the space between helixes on the same search layer, and $D2$ refers to the space between various search layers. The shadow block area in bold represents the search cross-section, and we can see the size of cross-section is $D1 \times D2$. When $N = 1, 4, 9$, $D1$ and $D2$ are as shown in Figure 8.

4. Simulation and Results

We consider mobile nodes to be operating in a three-dimensional underwater mission space. Assume that the

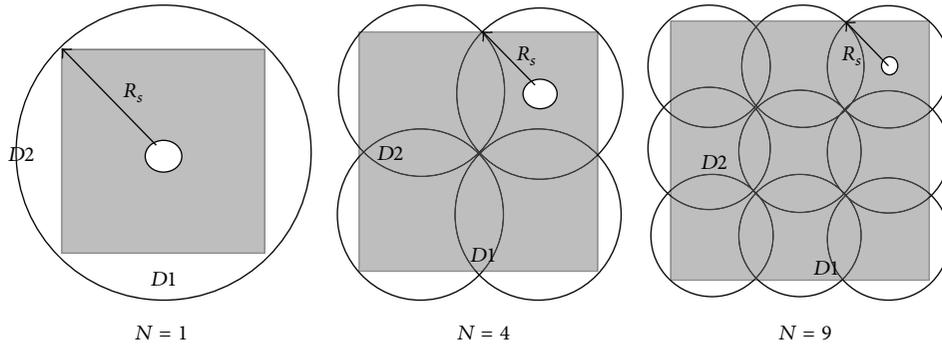


FIGURE 8: Diagrammatic sketches of D1 and D2 when $N = 1, 4, 9$.

mission is to find survivors from M targets using N nodes. Let set G denote the M targets $G = \{m_j \mid j = 1, 2, \dots, M\}$ and let set B denote N nodes $B = \{n_i \mid i = 1, 2, \dots, N\}$. Note that a target may change its location during operation of the system, or new targets may show up; hence M and y_j may not be constant. At the same time, a node may malfunction, and thus N may also change in time. Associated with the j th target is a life value L_j , if $L_j \leq 0$, this means target j th fails, and in reality, it might mean that the survivor is dead. The mission's objective is to maximize the total life value collected by visiting target points in the set G within a given mission time T . Target life value may be time dependent, typically decreasing in time. The exact location of targets may not always be known in advance and there may be obstacles in the mission space, which constrain the feasible trajectories of nodes.

We model the mission space as a cylinder $A \subset \mathbf{R}^3$, over which there is an event density function $\rho(x)$, $x \in A$, that captures the frequency or density that a specific event takes place (in Hz/m^3). $\rho(x)$ satisfies $\rho(x) \geq 0$ for all $x \in A$. In this paper, $\rho(x)$ may be the frequency that a survivor appears at target point. When an event occurs at point x , it emits a signal and this signal is observed by a sensor at that location nearby.

To distinguish the relative importance of targets at time t , each target has an associated life function denoted by $L_j \Phi_j(t)$, where L_j is the maximal life value and $\Phi_j(t) \in [0, 1]$ is a discounting function which describes the life value change over time. When a deadline is associated with a particular target point, we can use

$$\Phi_j(t) = \begin{cases} 1 - \frac{\mu_j}{H_j}t, & t \leq H_j, \\ (1 - \varepsilon_j)e^{-\eta_j(t-H_j)}, & t > H_j, \end{cases} \quad (13)$$

where H_j is a deadline assigned to target point j and $\varepsilon_j \in (0, 1]$, $\eta_j > 0$ are parameters which may be target specific and are chosen to reflect different cases of interest.

The optimal searching problem can be formulated as an optimization problem to maximize the expected life value collected by the sensors over the mission space A :

$$\text{MAX} \int_A L_j \Phi_j(A) \rho(A) P_A(B) F(A) dA. \quad (14)$$

By referring to the above definition, $L_j \Phi_j(A)$ in Formula (14) refers to the fact that the life value represented by the

target points in area A reduces with the discounting function, $\rho(A)P_A(B)$ refers to the joint detection probability of the target points in area A by all nodes in set B , and $F(A)$ refers to the node trajectory equation in area A . Therefore, the meaning of Formula (14) is to maximize the expected life value collected by the sensors over the mission space A .

Table 1 shows the parameters of simulation experiment, and in order to make the experiment as close to the actual situation as possible, we have set the experiment parameters.

4.1. Total Search Time versus Nodes Number. Figure 9 shows the relational diagram between the overall searching time and the node number under two schemes. In the first scheme, the fish swarm conducts cruise in accordance with a helix equation (blue bar graph), and in the second scheme, random walk is adopted. In accordance with the diagram, we can see that in both schemes, the overall searching time reduces with the increase of node number ($N = 1, 4, 9$), but the random walk scheme takes significant more overall searching time than the helix scheme. When the node number is $N = 1, 4, 9$, the time taken by the helix scheme is, respectively, 21.9%, 22.2%, and 22.6% of the time taken by the random walk scheme. When the node number is $N = 1$, the time taken by the helix scheme is 2.99 times of that when $N = 4$ and 8.96 times of that when $N = 9$. It can significantly reduce the searching time by increasing the number of fishes in the fish swarm, but of course, this will also increase the communication consumption and complexity. A feasible method is to increase the number of fish swarms, which can also achieve the effect of reducing the overall searching time.

4.2. Average Rescue Time versus Nodes Number. Figure 10 shows the relational diagram between the average rescuing time and the node number under two schemes. The average rescuing time refers to the average time to find 5 target points. In accordance with the diagram, we can see that in both schemes, the average rescuing time reduces with the increase of node number, but the average rescuing time taken by the random walk scheme is longer than the helix scheme. When the node number is $N = 1, 4, 9$, the time taken by the helix scheme is, respectively, 41.6%, 38.2%, and 55.4% of the time taken by the random walk scheme. In accordance with the set parameters in Table 1, when the node number is $N = 1$, the average rescuing time taken by the helix

TABLE 1: Experiment parameters.

| Network configuration | Node | Artificial fish | Others |
|-----------------------|-----------------------------|------------------------------|-------------------------|
| Radius $R = 1$ km | $R_s = 50$ m | Swimming speed $V = 1.5$ m/s | Life value = 10 minutes |
| Depth $d = 200$ m | $R_c = 100$ m | $Visual = R_c$ | |
| $c_{th} = 0.9$ | $R_e = 0.5 R_s$ | $Step = R_s$ | |
| $N = \{1, 4, 9\}$ | $\alpha = 0.2; \beta = 2.0$ | | |
| Targets: $M = 5$ | | | |

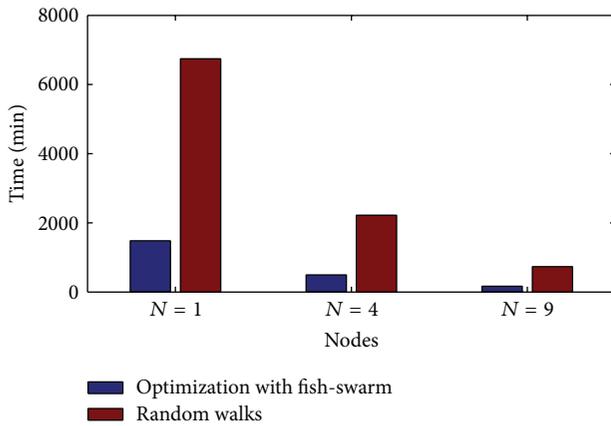


FIGURE 9: Overall searching time versus node number.

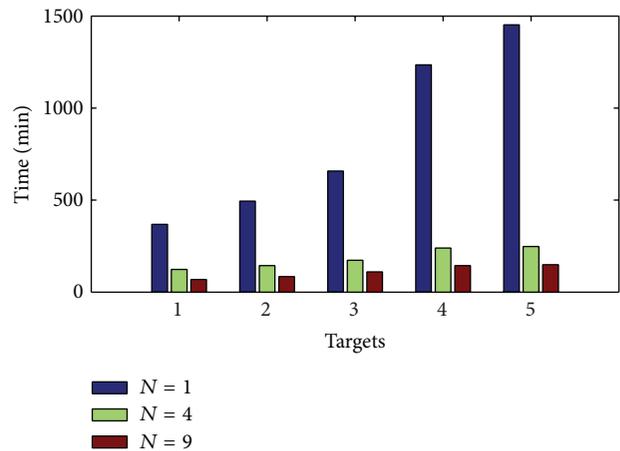


FIGURE 11: Target discovery time.

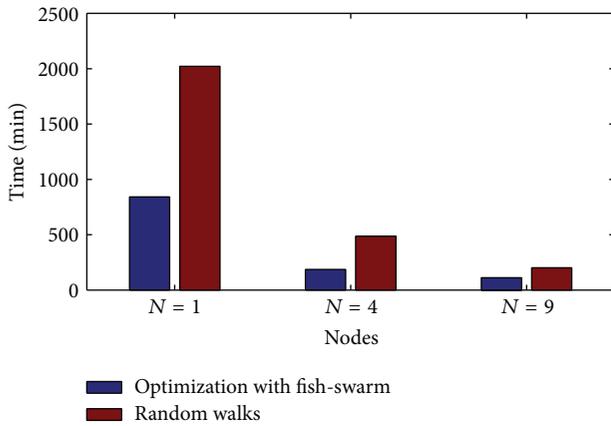


FIGURE 10: Average rescuing time versus node number.

scheme is 841.5 minutes, and when $N = 4$ and $N = 9$, it takes 185.5 minutes and 111.2 minutes, respectively. This is far from the 10-minute threshold value of life value we expect, and the solutions include increasing the swimming speed of fish swarm, increasing the number of fish swarms, or increasing the number of fishes in the fish swarm. Increasing the swimming speed of fish requires consideration on the aspect of mechanics, which exceeds the discussion in this paper; increasing the number of fishes in the fish swarm is restricted because excessive nodes in the swarm will cause communication delay, which will also increase the control overhead. A feasible method is to increase the number of fish swarms while at the time ensuring that there will not

be excessive fishes in the fish swarm, which will significantly reduce the average rescuing time.

4.3. Target Discovery Time. Figure 11 shows the target discovery time by the fish swarm in accordance with the helix equation under the cruise scheme, and 5 total target points are found, which are randomly distributed in the whole task Area A. Record the 5 time points where the fish swarm find them in accordance with the cruise route. In accordance with the diagram, we can see that the discovery time of 5 targets presents gradual progressive increase, which is easy to understand. In reality, the sooner the target is discovered, the more beneficial it is to the rescue work, which requires the target discovery time to be as short as possible. In the diagram, we can see that it can significantly reduce the target discovery time by increasing the node number. For example, when $N = 9$, the time to discover the first target is 244.9 minutes shorter than that when $N = 1$, and the time to discover the fifth target is 1204.2 minutes shorter. In the meantime, in order to further reduce the target discovery time, the number of fish swarms can also be increased, so that the fish swarms can conduct searching at different search layers, which can significantly reduce the target discovery time and increase the rescuing efficiency.

4.4. Optimized Formation versus No Optimization. In accordance with the above result, we can see that the optimized formation can increase the search cross-sectional area of fish swarm, and Figure 12 shows the comparison between the optimized formation and the formation with no optimization

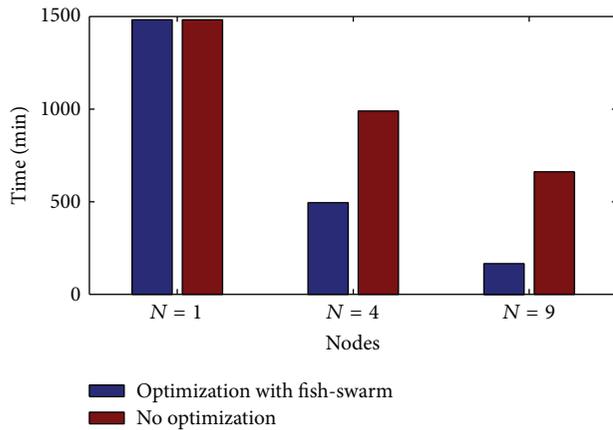


FIGURE 12: Comparison between the optimized formation and the formation with no optimization.

when the node number is $N = 1, 4, 9$. In accordance with the diagram, we can see that optimized formation to the fish swarm can significantly reduce the searching time. When $N = 1$, because there is only one fish, there is no so-called formation optimization; when $N = 4$, the efficiency has increased by 51.3% after optimization; when $N = 9$, the efficiency has increased by 74.6% after optimization. Apparently, the more nodes there are, the more significant the optimization result is. However, during the actual application, there should not be too many fishes in the fish swarm because this will cause a high cost of optimization.

5. Conclusions

Under the background of sea rescue, this paper proposes a search scheme based on the fish-swarm optimization algorithm, and the sensor nodes moving underwater are considered as the artificial fishes. In this paper, a trajectory model of fish swarm cruise is built in accordance with the helix equation, the fish-swarm optimization algorithm is used to optimize the formation of nodes, and the fish swarm behavior, such as gathering, chasing, and preying, is used to control the movement of nodes, so that maximum cross-sectional area of search by the fish swarm can be achieved. The experiment result shows that compared to the random walk model, it can significantly reduce the searching time in accordance with the preset trajectory cruise, which has reference value for practical application; in the meantime, it can also help reduce the searching time through formation optimization of fish swarm. Sea rescue is a task that has high requirement of saving time, the next step of work should be further in-depth research on the issues discussed in this paper, and how to adapt to underwater barriers will also be discussed.

Conflict of Interests

The authors declare that they have no conflict of interests.

Acknowledgment

This paper is a part of the work supported by the National Natural Science Foundation of China (no. 61301108).

References

- [1] M. Kemp, B. Hobson, J. Meyer, R. Moody, H. Pinnix, and B. Schulz, "MASA: a multi-AUV underwater search and data acquisition system," in *Proceedings of the MTS/IEEE (OCEANS '02)*, vol. 1, pp. 311–315, October 2002.
- [2] B. Anderson and J. Crowell, "Workhorse AUV—a cost-sensible new autonomous underwater vehicle for surveys/soundings, search & rescue, and research," in *Proceedings of the MTS/IEEE (OCEANS '05)*, pp. 1–6, September 2005.
- [3] A. Jacoff, *Search and Rescue Robotics*, Springer, 2008.
- [4] E. P. Gels, *Electroactive Polymer (EAP) Actuators As Artificial Muscles: Reality, Potential, and Challenges*, 2004.
- [5] J. Yu, S. Wang, and M. Tan, "A simplified propulsive model of bio-mimetic robot fish and its realization," *Robotica*, vol. 23, no. 1, pp. 101–107, 2005.
- [6] Z. Wang, Y. Wang, J. Li, and G. Hang, "A micro biomimetic manta ray robot fish actuated by SMA," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 1809–1813, December 2009.
- [7] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of the 20th IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, April 2001.
- [8] J. Cortés, S. Martínez, T. Karataş, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [9] W. Li and C. G. Cassandras, "A minimum-power wireless sensor network self-deployment scheme," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 3, pp. 1897–1902, March 2005.
- [10] L. Mihaylova, T. Lefebvre, H. Bruyninckx, and K. Gadeyne, "Active sensing for robotics—a survey," in *Proceedings of the 5th International Conference on Numerical Methods and Applications*, pp. 316–324, Borovets, Bulgaria, 2002.
- [11] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems*, vol. 5, pp. 299–308, Springer, Tokyo, Japan, 2002.
- [12] D. O. Popa, H. E. Stephanou, C. Helm, and A. C. Sanderson, "Robotic deployment of sensor networks using potential fields," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 1, pp. 642–647, May 2004.
- [13] G. Tan, S. A. Jarvis, and A.-M. Kermarrec, "Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 6, pp. 836–848, 2009.
- [14] A. Casteigts, J. Albert, S. Chaumette, A. Nayak, and I. Stojmenovic, "Biconnecting a network of mobile robots using virtual angular forces," *Computer Communications*, vol. 35, no. 9, pp. 1038–1046, 2012.
- [15] Y. Song, B. Wang, Z. Shi, K. Pattipati, and S. Gupta, *Distributed Algorithms for Energy-Efficient Even Self-Deployment in Mobile Sensor Networks*, 2013.

- [16] C. Nowzari and J. Cortés, “Self-triggered coordination of robotic networks for optimal deployment,” *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.
- [17] J. Cortes, “Coverage optimization and spatial load balancing by robotic sensor networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 749–754, 2010.
- [18] C. Yao, X. C. Ding, and C. G. Cassandras, “Cooperative receding horizon control for multi-agent rendezvous problems in uncertain environments,” in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 4511–4516, December 2010.
- [19] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Distributed coverage control and data with mobile sensor networks,” *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2445–2455, 2011.
- [20] X. C. Ding, C. Belta, and C. G. Cassandras, “Receding horizon surveillance with temporal logic specifications,” in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 256–261, December 2010.
- [21] X. L. Li and J. X. Qian, “Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques,” *Journal of Circuits and Systems*, vol. 1, pp. 1–6, 2003.
- [22] H.-C. Tsai and Y.-H. Lin, “Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior,” *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5367–5374, 2011.
- [23] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1293–1303, April 2003.

