Multiple Task Assignment and Path Planning of a Multiple Unmanned Surface Vehicles System Based on Improved Self-Organizing Mapping and Improved Genetic Algorithm

Guoqing Xia, Xianxin Sun* and Xiaoming Xia

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; guoqingxia@hrbeu.edu.cn (G.X.); 1611887533@hrbeu.edu.cn (X.X.)
* Correspondence: hrbsunxianxin@hrbeu.edu.cn

Abstract: This paper addresses multiple task assignment and path-planning problems for a multiple unmanned surface vehicle (USVs) system. Since it is difficult to solve multi-task allocation and path planning together, we divide them into two sub-problems, multiple task allocation and path planning, and study them separately. First, to resolve the multi-task assignment problem, an improved self-organizing mapping (ISOM) is proposed. The method can allocate all tasks in the mission area, and obtain the set of task nodes that each USV needs to access. Second, aiming at the path planning of the USV accessing the task nodes, an improved genetic algorithm (IGA) with the shortest path is proposed. To avoid USV collision during navigation, an artificial potential field function (APFF) is proposed. A multiple USV system with multi-task allocation and path planning is simulated. Simulation results verify the effectiveness of the proposed algorithms.

Keywords: artificial potential field function; improved genetic algorithm; improved self-organizing mapping; multiple tasks assignment; multiple unmanned surface vehicles; path planning

1. Introduction

Compared to manned surface ships, unmanned surface vehicles (USVs) have significant advantages when carrying out dangerous and boring tasks. However, there are some limitations when a single USV performs tasks, such as the limitation of maximum range, lack of scalability, etc. These problems can be effectively overcome using multiple USVs. A USV swarm is a typical multi-agent system. A multiple USV system, which is one of the most actively studied topics within multi-agent systems, generally aims to drive multiple agents to meet prescribed constraints on their states [1]. To research the multi-USV system more conveniently, it is divided into two sub-problems: multiple task assignment and path planning.

Recently, there has been progress regarding multiple task assignment in multi-agent systems. For example, a new intelligent multi-task allocation is proposed based upon the self-organizing map (SOM) [2]. The SOM adopts ring topology, with \( k \) SOMs for \( k \) USVs. In [3], an improved SOM is proposed for intelligent hybrid multi-task allocation of a single USV. However, for a single USV with multi-task allocation, this is equivalent to the path-planning problem, or the traveling salesman problem. In [4], an intelligent self-organized algorithm is proposed to solve a cooperative search–attack mission-planning problem for multiple unmanned aerial vehicles (UAVs). In [5], an integrated multiple autonomous underwater vehicle (AUV) dynamic task assignment and path-planning algorithm is proposed by combining the improved self-organizing map neural network with a novel velocity synthesis approach. In [6], a SOM-based neural network approach is proposed for multiple task assignment of a multi-robot system. In [7], a SOM with the shortest load-balancing constraints is proposed for AUVs. In contrast to our methods, which used distributed methods to deal with multi-task assignment and path planning, a centralized
treatment approach is adopted in [2–7]. Each SOM corresponds to an object and outputs an ordered task array. Such a solution is theoretically feasible, but not practically applicable.

Path planning is another sub-problem of a multiple USV system. The method has two parts: an exact algorithm and a heuristic algorithm. A deterministic method can provide accurate results, but it takes a long time. A heuristic algorithm improves running speed by sacrificing the calculation accuracy to obtain an approximate optimal solution. In this case, a heuristic algorithm, such as the improved quantum ant colony algorithm [8], order-first split-second approach [9], fast marching method [10], interpolation algorithm [11], virtual spring method [12], improved genetic algorithm [13], artificial intelligence [14], A* algorithm [15], self-organizing map [16], virtual target approach [17], etc., is mostly used in path planning. GA takes all individuals in a population as the object, and uses randomization technology to search an encoded parameter space efficiently.

Obstacle avoidance is a very important part of path planning. Researchers have paid extensive attention to obstacle avoidance. In [7], a hybrid path-planning approach is proposed to guide AUVs to reach their targets safely. In [15], an artificial potential field-based path-following approach with dynamic collision avoidance is proposed. In [18], a method based on a finite-control set model-predictive control is proposed. In [19], a distributed coordination mechanism is proposed for collision avoidance of multiple ships. In [20], an approach to normalize pseudo-distances is proposed. By ensuring the pseudo-distance was always larger than the safety threshold value, collisions with obstacles were avoided. In [21], a collision-avoidance potential function based on position estimation is proposed. The effectiveness of the collision-avoidance potential function assumes that the estimator has converged to stability. In actual engineering, the stability of the estimator cannot be guaranteed.

Due to the complexity of the multiple USV system, this paper divides it into two sub-problems, namely multi-task assignment and path planning. In the first stage, an improved self-organizing mapping (ISOM) for multiple task assignment of all USVs is proposed for multiple task allocation. An ISOM can be used to achieve the multiple task assignment of all USVs, with the resulting task set corresponding to USV being an unordered array. In the second stage, which aims at the path-planning problem of the USV accessing tasks, an improved genetic algorithm (IGA) is proposed. To solve the problem of USVs colliding during navigation, an artificial potential field function (APFF) based on position information measured by the sensor system is proposed. The accuracy of the information measured by the sensor system can be guaranteed in engineering. Finally, collision-free trajectories are generated for each USV.

In summary, the main contributions of this paper are summarized as follows.

1. For multiple task assignment, an improved self-organizing mapping algorithm is proposed. In this case, all tasks are distributed uniformly using the ISOM, and a set of tasks corresponds to each USV.
2. For path planning, an improved genetic algorithm with the shortest path as the cost function is proposed. By improving the genetic algorithm, we calculate the unordered array from multi-task assignment to obtain a set of ordered arrays with the shortest track.
3. Obstacle avoidance is a very important part of path planning. To avoid obstacles in USV path planning, an APFF based on position information measured by the sensor system is proposed.

This paper is organized as follows: Section 2 describes the problem of multi-USV systems, the model of USV and some necessary preliminaries. Section 3 proposes an ISOM for multiple task allocation. Section 4 designs an IGA and APFF for path planning. Section 5 gives simulation and comparison results to verify the effectiveness of the proposed method. Section 6 concludes the paper.
2. Problem Description and Notation

2.1. Problem Description

A multiple USV system is a complex system. In much of the literature, path planning is combined into SOM to provide multi-task allocation and path planning at the same time. This method is feasible in theory, but not in practice, because path planning is a very time-consuming algorithm. In this case, multiple task assignment and path planning are executed in the same program, which will run slowly. The distributed processing of a multiple USV system mainly focuses on two areas: multiple task assignment and path planning. The multiple USV system framework based on multi-task assignment and path planning is shown in Figure 1.

![Figure 1. Multiple USV system framework based on multi-task assignment and path planning.](image)

The first challenge is the multi-task allocation problem of the multiple USV system. Aiming to solve the problem of multiple task assignment, an ISOM is proposed. An ISOM can allocate all tasks for all USVs. Through ISOM, the task set accessed by each USV is obtained, and each task belongs to only one task set. Moreover, the ISOM has good extensibility and scalability. For example, the number of USVs needs to be added or reduced, and some tasks are added or modified, etc.

The second challenge is to give all USVs access to their task node sets along an optimum path. Multiple tasks can be regarded as multiple task points, and access to the set of tasks can be considered to be the traveling salesman problem (TSP). Based on traditional path planning, an IGA with the shortest path is proposed. At the same time, to avoid USV collisions during navigation, an APFF is proposed.

A real USV is in a three-dimensional space, which makes path planning problems of multiple USVs complex. The USV motion can be decoupled into two motions in a two-dimensional plane and a vertical direction, respectively. As shown in the Figure 2, the mission area is two-dimensional [4].

**Assumption 1.** All the tasks are in the mission area, and the priority of each task is the same.

**Assumption 2.** The USVs are homogeneous USVs with basic capabilities for navigation, obstacle avoidance, and location recognition [4].
2.2. Modeling of the USV

The 3 degrees of freedom (DOFs) kinematic equations of the USV can be expressed in vector form as [22]

\[ \dot{\eta} = R(\psi) \dot{\eta} = R(\psi) v \]  
\[ M \ddot{v} + D(\dot{v}) v = \tau + \tau^* \]

where \( R(\psi) \) is the rotation matrix, it is given as:

\[ R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

with the properties: \( ||R(\psi)|| = 1 \) and \( R^T(\psi)R(\psi) = I_{3 \times 3} \). \( \eta := [x, y, \psi]^T \) is the position and yaw angle in the earth-fixed frame \( X_EO_EY_E \) (see Figure 3). \( \dot{\eta} := [\dot{x}, \dot{y}, \dot{\psi}]^T \) is the position-yaw and velocity information for each vessel. \( \dot{v} := [\dot{u}, \dot{v}, \dot{r}]^T \) is the velocity vector in the body-fixed frame \( X_BO_BOY_B \), where \( u, v, \) and \( r \) represent the surge, sway, and yaw angular velocities, respectively. The system inertia matrix \( M \) is positive, definite, and constant, where \( M = M^T \in \mathbb{R}^{3 \times 3} \) and \( \dot{M} = 0 \). The damping matrix \( D \) is also symmetric and positive definite, i.e., \( D \in \mathbb{R}^{3 \times 3} \). \( \tau := [\tau_1, \tau_2, \tau_3]^T \) is the control input, which is produced by propellers. \( \tau^* := \tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{ocean}} \) is the total environment disturbance by wind, waves, and ocean currents, respectively.

Assumption 3. The position-yaw and velocity information for each vessel are available.
Remark 1. The environmental disturbances consist of a low-frequency part and high-frequency part. In this paper, only the low-frequency part is considered during the control process.

2.3. Notation

For the multi-USV system of multiple task assignment and path planning, the following notations will be used in this paper. \( T = \{ T_1, T_2, \ldots, T_i, \ldots, T_M \} \) represents the \( M \) tasks randomly distributed in the mission area. Throughout the paper, \( T_i \) represents the \( i \)th task, i.e., \( i = 1, 2, \ldots, M \). \( T_i = (t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{iK}) \) represents the \( i \)th target node is \( K \)-dimensional, i.e., \( k = 1, 2, \ldots, K \). \( U = \{ USV_1, USV_2, \ldots, USV_j, \ldots, USV_N \} \) represents the set of \( N \) USVs in the multiple USV system. Throughout the paper, \( USV_j \) represents the task set of the \( j \)th USV needs to access, i.e., \( j = 1, 2, \ldots, N \). At the same time, \( M \) is the number of neurons in the input layer, which is the same size as the number of task nodes in multiple USVs system. \( N \) is the number of cluster centers in the ISOM network, and the size is consistent with the number of USVs in multi-USV system. \( O = \{ O_1, O_2, \ldots, O_q, \ldots, O_Q \} \) represents the set of \( Q \) obstacles in the mission area, i.e., \( q = 1, 2, \ldots, Q \). \( \max(\bullet) \) means to take the maximum value of the \( \bullet \). \( (\cdot)^\top \) represents the transpose of a matrix.

3. Improved Self-Organizing Mapping

The self-organizing mapping network and multiple USV system multi-task allocation module have many similar characteristics and phenomena. First, the number of input objects in a SOM network can be changed arbitrarily. It is not affected by the design principle of the SOM network. Second, SOM adopts a mechanism of competitive learning, which gives adjacent neurons similar characteristics. In the multiple USV system, each USV can be understood as the “cluster center” in SOM. Because of these similar characteristics and phenomena, it is feasible to apply SOM in the multiple task assignment module of the multi-USV system.

3.1. The Principle of Self-Organizing Mapping

Self-organizing mapping, also known as the Kohonen network, was first proposed by Kohonen, and can realize unsupervised learning and clustering of data [23,24]. The Kohonen network is a neural network with only an input layer and a computational layer, as shown in Figure 4. SOM assumes that there are some topological structures or orders in the input objects, which can realize mapping from the input layer to the computational layer. The mapping can preserve topological features.

Suppose \( M \) tasks and \( N \) USVs are randomly distributed in the mission area. The SOM network model with an input layer and computational layer is shown in Figure 4. The first layer is the input layer. It contains \( M \) neurons, and the dimension of each neuron is \( K \)-dimensional. Therefore, the expression pattern of the \( i \)th neuron is \( T_i = (t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{iK}) \), \( i = 1, 2, \ldots, M \). \( K = 1, 2, \ldots, K \). The second layer is the output layer. It contains \( NM_1 \) neurons \( (R_{11}, \ldots, R_{1M_1}, R_{21}, \ldots, R_{2M_1}, \ldots, R_{N1}, \ldots, R_{NM_1}) \). It represents the coordinates of the \( N \) USVs and the corresponding set of \( M_1 \) task nodes \( (M_1 \leq M) \). \( R_{nm} = (w_{nm1}, w_{nym}) \) rep-
represents the connection weight of the output neurons to the input neurons, \( n = 1, 2, \ldots, N \), \( m = 1, 2, \ldots, M \), \( M_1 \leq M \). At the beginning, the SOM is initialized with the connection weights \((w_{nmx}, w_{nym})\), \( n = 1, 2, \ldots, N \), which are the initial locations of the \( N \) USVs.

### 3.2. Initialization Parameters

These initialization parameters mainly include a training set \((data_{training})\), test set \((data_{test})\), maximum value of iterations \((iter)\), learning rate \((\eta_0)\), learning rate parameter \((learn\_para)\), neighborhood radius \((neigh\_radius)\), neighborhood parameter \((neigh\_para)\), weight of connecting all neurons \((w)\), etc. In addition, we focus on one of the parameters \( w \), which meets the following condition

\[
  w = \text{rand}(N, M). \tag{4}
\]

where \( w \in \mathbb{R}^{N \times M} \) represents the \( N \times M \)-dimensional Euclidean Space, in which the elements are randomly distributed in \((0,1)\). \( N \) and \( M \) represent the numbers of USVs and tasks, respectively.

### 3.3. Winner Selection Rules

For a given task as an input object, the output neurons compete to be the winner according to a specified criterion described as \([6]\) references\( W_i = \min_i D_{inm} \), \( i = 1, 2, \ldots, M \), \( n = 1, 2, \ldots, N \), \( m = 1, 2, \ldots, M_1 \). \( W_i \) denotes that the \( i \)th neuron from the \( nth \) group of the computational layer is the winner of the \( m \)th input layer object. \( D_{inm} \) is the Euclidean distance between \( T_i \) and \( R_{nm} \), which is calculated by

\[
  D_{inm} = \| T_i - R_{nm} \| \times (1 + P), \tag{6}
\]

where

\[
  \| T_i - R_{nm} \| = \sqrt{\sum_{k=1}^{K} (t_{ik} - w_{nm})^2}, \tag{7}
\]

where \( T_i = (t_{i1}, t_{i2}, \ldots, t_{ik}, \ldots, t_{iK}), i = 1, 2, \ldots, M, k = 1, 2, \ldots, K \), \( T_i \) is the target coordinate of the \( i \)th input node. \( R_{nm} = w_{nm}, n = 1, 2, \ldots, N, m = 1, 2, \ldots, M_1, i \neq m, R_{nm} \) is the coordinate of the \( m \)th neuron from the \( nth \) group of the computational layer.

Parameter \( P \) in \((6)\) represents the equitable distribution of workload for \( j \)th USV \([7]\), and it is defined as

\[
  p_j = \frac{L_j - \overline{L}_j}{1 + \overline{L}_j}, \tag{8}
\]

where \( j = 1, 2, \ldots, N \). \( L_j \) and \( \overline{L}_j \) are the actual cruise path length and the average cruise path length of the \( j \)th USV, respectively.

Without losing generality, we need to consider the workload and energy consumption of each USV to ensure that all USVs can return to the supply point for fuel consumption. Considering different types of USV, their equipment is different, which will lead to different basic technical parameters. For example, the basic technical parameters of the US Navy “ship-class” unmanned combat surface craft are as follows: the maximum speed is between 32 knots and 35 knots, the cruise speed of towing and other mine-sweeping tasks can reach 20 to 25 knots, and the sea self-sustaining capacity can reach 48 h. The cruise speed of the remote-control multi-mission surface craft designed by the US Navy is 16 knots, and its self-sustaining power can reach 24 h during the operation. Here, we choose the cruising
speed of USV at 20 knots and cruise duration of 24 h. The track length \( L_{\text{path}} \) of USV is calculated as follows:

\[
L_{\text{path}} = v \times t = 20 \text{(knots)} \times 1.852 \text{(km/h)} \times 24 \text{(h)} = 888.96 \text{(km)}.
\] (9)

After the above calculation, the cruise path reaches 888.96 km, which can fully meet our design requirements. In this case, the influence of parameter \( P \) can be ignored, and Formula (6) can be rewritten as follows:

\[
D_{\text{inm}} = \| T_i - R_{nm} \|. \quad (10)
\]

3.4. Neighborhood Updating Rules

The next step is to calculate the distance between the other neurons and the winner neuron, which is the weight \( w \). The key is to determine the neighborhood function \( h_{ji} \). We choose the function as the neighborhood function of the winner neuron, and the specific expression is as follows:

\[
h_{ji}(t) = \begin{cases} 
e^{-d_{ji}^2/2\sigma^2(t)}, & d_{ji} \leq \text{neigh radius} \\ 0, & \text{otherwise} \end{cases}
\] (11)

where \( d_{ji} = \| T_i - T_{\text{winner}} \| \), \( j = 1, 2, \ldots, N \), \( i = 1, 2, \ldots, M \). \( d_{ji} \) represents the distance between the \( i \)-th neuron and the winner neuron in the \( j \)-th output neuron. \( \sigma(t) = \sigma_0 \times e^{-t/\tau} \) is a monotone decreasing function, and \( \sigma_0 > 0 \) is a small constant. The \( h_{ji} \) provides a percentage-like measure of how close the neurons are.

The initial neighborhood radius \( \text{neigh radius}(0) \) should be half the length of the whole output plane at least. The neighborhood radius \( \text{neigh radius}(t) \) is defined as follows

\[
\text{neigh radius}(t) = \text{neigh radius}(0) \times e^{-t/\tau_{\text{par}}},
\] (12)

where \( \text{neigh radius}(t) \) is a monotone decreasing function. \( \text{neigh radius}(0) > 0 \) is a very large constant. \( \tau_{\text{par}} \) represents neighborhood parameter. After iteration, the neighborhood radius will converge to a small normal value.

The update rule without considering the ocean current is defined as

\[
R_{nm}(t+1) = \begin{cases} T_i, & D_{\text{inm}} \leq D_{\text{min}} \\ R_{nm}(t) + \beta \times h_{ji} \times (T_i - R_{nm}), & \text{otherwise} \end{cases}
\] (13)

where \( D_{\text{min}} \) represents the range of topological neighborhood structure. After calculation, \( D_{\text{min}} = \text{neigh radius} \times \text{neigh radius} \) is a large enough number. After iteration, once \( D_{\text{inm}} \leq D_{\text{min}} \), \( R_{nm}(t+1) \) is expressed as

\[
R_{nm}(t+1) = T_{\text{winner}}(t).
\] (14)

3.5. Weight Updating Rules

The learning rate is represented by the character \( \eta(t) \), which is defined as follows

\[
\eta(t) = \eta(0) \times e^{-t/\tau_{\text{learn}}},
\] (15)

where \( \eta(t) \) is a monotone decreasing function denoting learning rate, \( \eta(0) = \eta_0, \eta_0 \in (0, 1] \) is a constant, \( \tau_{\text{learn}} \) represents learning rate parameter.

Through the above description, \( w(t+1) \) is expressed as

\[
w(t+1) = w(t) + \eta(t) \times h_{ji}(t) \times (T_i - w(t)).
\] (16)
3.6. Improved Self-Organizing Mapping

Next, the improved SOM is divided into the following two areas.

(1) The target set is divided into a training set and test set. In the training process of the ISOM algorithm, only the training set is used. When the training is completed and the classification effect is achieved, the test set is processed. This can ensure the accuracy of test-set classification.

(2) The verification of the test set is divided into two steps: storage and grouping. After testing, $M$ neurons were stored. According to the serial number of the task node in the storage data, the grouping is completed. Finally, the set of tasks is the effect of the multi-USV system’s multiple task allocation.

3.7. Pseudocode and Flowchart of Improved Self-Organizing Mapping

After the above comprehensive analysis and overall design, we can configure the ISOM algorithm. The pseudocode is shown in Algorithm 1 and the flowchart of the ISOM, as shown in Figure 5.

Figure 5. The improved self-organizing mapping flowchart.
Algorithm 1 ISOM algorithm-based multiple USVs system multi-task assignment module.

Input: Parameter Initialization. training set (data\textsubscript{training}), test set (data\textsubscript{test}), number of targets in the test set (N), number of neuron nodes (M), maximum number of iterations (iter), learning rate ($\eta_0$), learning rate parameter (learn\_para), neighborhood radius (neighbor\_radius), neighborhood parameter (neighbor\_para), weight of connecting all neurons ($w$), etc.

Output: ISOM\textsubscript{num} = cell(1, N); Grouping: $X_{\text{group}1}$, $X_{\text{group}2}$, ..., $X_{\text{group}N}$

// training set
for $i = 1$: iter\_max do
  for $j = 1$: M\textsubscript{training} do
    Through the calculation of Formulas (2)–(7), the winner neuron is found $W_i$;
    Through Formulas (8)–(11), calculate the neighborhood function $h_{ji}$;
    for $i = 1$: N do
      Through the calculation of Formula (13), update the weight of connecting all neurons $w$;
    end for
  end for
  Through the calculation of Formula (9), update the neighborhood radius neighbor\_radius;
  Through the calculation of Formula (12), update the learning rate $\eta$;
end for

// test set
ISOM\textsubscript{num} = cell(1, size(w, 1));
for $i = 1$: size(w, 1) do
  Storage: ISOM\textsubscript{num} = cell(1, N);
end for
for $i = 1$: M\textsubscript{test} do
  Grouping: $X_{\text{group}1}$, $X_{\text{group}2}$, ..., $X_{\text{group}N}$;
end for

4. Improved Genetic Algorithm
4.1. Genetic Algorithm

A genetic algorithm is a computational model to simulate the Darwinian evolution process of natural selection and genetic stimulation. It finds the best solution by simulating the process of biological evolution.

The genetic algorithm (GA) is a relatively mature heuristic algorithm. The IGA takes the shortest track as the fitness function. First, the IGA starts with a population of random solutions. Second, a fitness function is formed to evaluate the adaptability of the solution. Third, it determines whether the individuals in this generation are optimal solutions. If they are, the optimum solution is presented, and the IGA is ended. If not, then it interactively generates a new generation of individuals through genetic operations, including selection, crossover, and mutation to perform corresponding genetic operations. The IGA continues to judge whether the new generation of individuals is the optimal solution. The approximate optimum solution is found through continuous loops, and then gradually converges to the optimum solution.

Through the above comprehensive analysis, we can configure the IGA. The pseudocode is shown in Algorithm 2. The flowchart of the IGA is shown in Figure 6.
Algorithm 2 IGA-based multiple USVs system path-planning module.

**Input:** Parameters Initialization. initial population size ($M_{\text{popu}}$), maximum number of iterations ($M_{\text{iter}}$), crossover probability ($p_c$), mutation probability ($p_m$), etc.

**Output:** $L_{\text{path}}$. The best individual of each generation.

```plaintext
for k1 = 1: $M_{\text{popu}}$ do
    The initial population is produced
end for

while iter < $M_{\text{iter}} + 1$ do
    for k1 = 1: $M_{\text{popu}}$ do
        Genetic operation was carried out in turn
        Call their respective functions to selection, crossover, and mutation
    end for
    The population is updated, and a new generation of individuals is generated
    Calculate the fitness function
    Record the contemporary optimal fitness and best individual
    iter ← iter+1;
end while
```

Figure 6. The improved genetic algorithm flowchart.

4.2. **Artificial Potential Field Function**

The basic principle of artificial potential field (APF) is a virtual force field, assuming that USV as a point moves in a virtual force field. The repulsion field is composed of the repulsive force of all obstacles for the USV, and the gravitational field is generated by the attraction of the task node for the USV. The artificial potential field function (APFF) is equal to the sum of repulsion and gravitation. The design of the APFF is based on the information
measured by the sensor system. This information includes the actual position of multiple missions, multiple USVs, and obstacles in the mission area. Moreover, the accuracy of this information can be guaranteed. Therefore, APFF is feasible.

In APFF, we defined the following concepts. Repulsion is defined as follows

$$F_{o,j} = \omega_1 \times \max(e^{-a \times (X_j - O_q)^\top \times (X_j - O_q)}),$$

(17)

where $F_{o,j}$ represents the repulsive force of obstacles for the $j$th USV, $\omega_1$ represents the repulsion gain coefficient, which is a constant value. $a$ represents the threat level coefficient, which is a constant value. $X_j(x_j, y_j)$ and $O_q(x_q, y_q)$ represents the coordinates of the $j$th USV and $q$th obstacle, respectively $j = 1, 2, \ldots, N$ and $q = 1, 2, \ldots, Q$.

The gravitation is defined as follows

$$F_{t,j} = \omega_2 \times (X_j - T_l)^\top \times (X_j - T_l),$$

(18)

where $F_{t,j}$ represents the gravitation force of the $i$th task for the $j$th USV. $\omega_2$ represents the gravitation gain coefficient, which is a very small constant. $X_j(x_j, y_j)$ represents the coordinate of the $j$th USV, $j = 1, 2, \ldots, N$. $T_l = (x_l, y_l)$ represents the coordinates of the $l$th task corresponding to the $j$th USV, $l = 1, 2, \ldots, M_1$, $M_1 \leq M$.

The APFF is equal to the sum of repulsion and gravitation, which is as follows:

$$F_j = F_{o,j} + F_{t,j}$$

(19)

where $F_j$ represents the resultant force of the $j$th USV in the artificial potential field, $j = 1, 2, \ldots, N$.

Substituting (17) and (18) into (19), we obtain the force of the USV in the APFF. According to the gradient descent method, it can be deduced that the moving direction of the USV is the descent direction of the APFF.

5. Simulation and Results

Simulations were carried out with Matlab 2018a. The simulations were run on a PC with a dual-core 2.30 GHz Intel(R) Core(TM) i5-8300H CPU and 8 GB of RAM.

To verify the proposed algorithm, two different simulations were set up in this section, namely three experimental groups and one comparison group.

In simulations, the model of surface ship Cybership II was used [25]. The time-varying environmental disturbances were modeled as first-order Markov processes [26].

In the simulation results, the blue area represents high potential value, and the white area means a low potential value, which is approximately zero. When the mission area is a grid, the black area represents an obstacle, and the white area indicates an allowed area.

The simulation results are shown in Figures 7–10, where $USVi (i = 1, 2, \ldots, N)$ represents the $i$th USV. The * indicates the location of the tasks.

5.1. Experimental Group of Proposed Algorithms

In this subsection, the simulation results show the effectiveness of the proposed algorithm. As an experimental group, the algorithm proposed in this paper is used. The ISOM is used for multiple task allocation. The parameters are as follows: $\text{neigh}_{\text{radius}}(0) = 3$, $\text{neigh}_{\text{para}} = 10,000$, $\eta(0) = 1$, $\text{learn}_{\text{para}} = 10,000$. The IGA is used for path planning. The APFF is used to realize USV obstacle avoidance. The parameters are as follows: $\text{M}_{\text{popu}} = 100$, $\text{M}_{\text{iter}} = 200$, $p_c = 0.8$, $p_m = 0.08$, $\omega_1 = 1$, $\omega_2 = 0.0001$.

The experimental group includes three groups of experiments, which are experimental group 1, experimental group 2, and experimental group 3. The purpose of experimental group 1 is to test the effectiveness of the proposed method. Experimental group 2 aims to verify the stability and scalability of the ISOM. The purpose of experimental group 3 is to validate the effective of the APFF.
5.1.1. Experimental Group 1

According to the proposed algorithms, the number of tasks and the number of USVs are selected to be $M = 20$, $N = 2$ and $M = 20$, $N = 4$, respectively.

The multiple USV system simulation results of experimental group 1 are shown in Figure 7. Figure 7a shows the simulation effect of 20 tasks and 2 USVs. Figure 7b shows the simulation of 20 missions and 4 USVs. Through Figure 7, it can be verified that the proposed algorithm is effective.

Figure 7. Multiple USV system simulation results of experimental group 1 ($a = 0.8$). (a) $M = 20, N = 2$. (b) $M = 20, N = 4$.

Figure 8. Multiple USV system simulation results of experimental group 2 ($a = 0.8$). (a) $M = 15, N = 2$. (b) $M = 15, N = 4$. 
5.1.2. Experimental Group 2

Compared with experimental group 1, and with other parameters unchanged, only the number of tasks and the number of USVs are selected as $M = 15$, $N = 2$ and $M = 15$, $N = 4$, respectively.

The multi-USV system simulation results of experimental group 2 are shown in Figure 8. Figure 8a shows the simulation of 15 tasks and 2 USVs. Figure 8b shows the simulation effect of 15 missions and 4 USVs. From Figure 8, it also can be verified that the proposed algorithm can achieve multiple task allocation and path planning. By comparing Figures 7 and 8, it can be validated that the ISOM has good scalability, increasing or reducing the number of USV, and adding or deleting tasks, etc.

5.1.3. Experimental Group 3

In this subsection, the effectiveness of the proposed APFF has been verified. Only one parameter in the APFF was changed: the threat level coefficient ($a$). Other simulation conditions are the same as those in experimental group 1. $a = 0.8$ and $a = 1.6$ are the parameters of experimental group 1 and experimental group 3, respectively.

Simulation results are shown in Figure 9. Figure 9a shows the simulation of 20 tasks and 2 USVs. Figure 9b shows the simulation of 20 missions and 4 USVs. Figures 7 and 9 show the avoidance obstacle effect of USV when $a = 0.8$ and $a = 1.6$, respectively. When
$M = 20, N = 2$, the track lengths of USV1 and USV2 are shown in Table 1. When
$M = 20, N = 4$, the track lengths of USV1, USV2, USV3, and USV4 are shown in Table 2.

5.2. Comparison and Analysis

In this subsection, traditional path planning is used in the simulation process, but the
obstacles in the task area were not considered in [6]. In this paper, the proposed method
considers the improved genetic algorithm and the threat level coefficient ($a$) regarding
obstacles in the path planning of a multiple USV system.

The multi-USV system simulation results of comparison group 1 are shown in Figure 10.
Figures 10a,b show the multi-USV system simulation effect of each USV when $M = 20, N = 2$
and $M = 20, N = 4$, respectively. Traditional SOM can realize multi-task allocation of a
multiple USV system. However, in the process of path planning, the existence of obstacles
in the mission area was not considered. Comparing Figures 7 and 10, the trajectories of
multiple USVs will pass through the high-risk area, and some of them will even pass
directly through obstacles.

The route lengths of each USV when $M = 20, N = 2$ and $M = 20, N = 4$ are shown in
Tables 1 and 2. Through a comprehensive comparison of the contents in Tables 1 and 2, it
can be found that when the threat level of the obstacle decreases, the length of the route is
approximately equal to the straight route.

| Table 1. The track length of each USV in $M = 20, N = 2$. |
|---------------------------------------------------------|
| USV1 (m) | USV2 (m) |
| experimental group 1 ($a = 0.8$) | 39.6 | 81.7 |
| experimental group 3 ($a = 1.6$) | 37.5 | 78.7 |
| comparison group 1 | 36.9 | 78.1 |

| Table 2. The track length of each USV in $M = 20, N = 4$. |
|---------------------------------------------------------|
| USV1 (m) | USV2 (m) | USV3 (m) | USV4 (m) |
| experimental group 1 ($a = 0.8$) | 34.7 | 38.8 | 37.9 | 44.3 |
| experimental group 3 ($a = 1.6$) | 31.8 | 36.0 | 37.2 | 42.8 |
| comparison group 1 | 31.8 | 36.0 | 36.8 | 42.1 |

By comparing Figures 7 and 10, when changing the threat level coefficient ($a$) in the
APFF, the trajectory of the USV has already violated obstacles in the mission area. However,
each USV has a different track length. As shown in Table 1, the trajectory length of USV1
for $a = 0.8$ is 39.6, which is 5.6% above that of $a = 1.6$. The trajectory length of USV2 for
$a = 0.8$ is 81.7, which is 3.8% above that of $a = 1.6$. The same conclusion can be drawn from
the comparison of USV sailing length in Table 2. It can be concluded that the trajectory is
safe at the expense of the total length of the route. The smaller the value of $a$, the higher
the threat degree of obstacles. At the same time, the route is far from obstacles, safety is
improved, and the path length is increased.

6. Conclusions

In this paper, we studied a multiple USV system for multi-task assignment and
path planning. Multi-USV systems are divided into two: multiple task allocation and
path planning. Some algorithms have been proposed for multi-task assignment and path
planning. To obtain the task set of each USV, the ISOM is used to deal with multiple task
allocation in the mission area. To realize USV parallel path planning, an IGA is proposed
to obtain the ordered task sequence. To avoid a collision of USV trajectories, an APFF is
designed. Simulation results show the effectiveness of the proposed algorithms. However,
this paper does not consider the problem of dynamic obstacles and USV communication. These problems will be the focus of the future research.

**Author Contributions:** Conceptualization, G.X. and X.S.; methodology, X.S.; software, X.S. and X.X.; formal analysis, X.S. and X.X.; investigation, X.S.; supervision, G.X. and X.X.; funding acquisition, G.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the 7th Generation Ultra Deep-Water Drilling Unit Innovation Project.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Abbreviations**

The following abbreviations are used in this manuscript:

- USV unmanned surface vehicle
- ISOM improved self-organizing mapping
- IGA improved genetic algorithm
- APFF artificial potential field function

**References**

1. Oh, K.-K.; Park, M.-C.; Ahn, H.-S. A survey of multi-agent formation control. *Automatica* 2015, 53, 424–440. [CrossRef]
2. Liu, Y.; Song, R.; Bucknall, R.; Zhang, X. Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (USVs) using self-organising maps and fast marching method. *Inf. Sci.* 2019, 496, 180–197. [CrossRef]
3. Liu, Y.; Bucknall, R. Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing* 2018, 275, 1550–1566. [CrossRef]
4. Zhen, Z.; Xing, D.; Gao, C. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. *Aerosp. Sci. Technol.* 2019, 496, 180–197. [CrossRef]
5. Zhu, D.; Huang, H.; Yang, S.X. Dynamic Task Assignment and Path Planning of Multi-AUV System Based on an Improved Self-Organizing Map and Velocity Synthesis Method in Three-Dimensional Underwater Workspace. *Trans Cybern.* 2013, 43, 504–514.
6. Zhu, A.; Yang, S.X. A neural network approach to dynamic task assignment of multirobots. *IEEE Trans. Neural Netw.* 2006, 17, 1278–1287.
7. Chen, M.; Zhu, D. A Workload Balanced Algorithm for Task Assignment and Path Planning of Inhomogeneous Autonomous Underwater Vehicle System. *IEEE Trans. Cogn. Dev. Syst.* 2019, 11, 483–493. [CrossRef]
8. Xia, G.; Han, Z.; Zhao, B.; Liu, C.; Wang, X. Global Path Planning for Unmanned Surface Vehicle Based on Improved Quantum Ant Colony Algorithm. *Math. Probl. Eng.* 2019, 2019, 1–10. [CrossRef]
9. Palacios-Gasos, J.M.; Montijano, E.; Sagues, C.; Llorente, S. Cooperative Periodic Coverage With Collision Avoidance. *Trans. Control Syst. Technol.* 2019, 27, 1411–1422. [CrossRef]
10. Liu, Y.; Bucknall, R. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Eng.* 2015, 97, 126–144. [CrossRef]
11. Liang, Z.; Zhu, S.; Fang, F. A Twofold-interpolation-based Path Planning Algorithm and Its Path Following Based on Improved Virtual Vehicle Method. *Int. J. Control. Autom. Syst.* 2012, 10, 186–191. [CrossRef]
12. Pan, Z.; Wang, D.; Deng, H.; Li, K. A Virtual Spring Method for the Multi-robot Path Planning and Formation Control. *Int. J. Control. Autom. Syst.* 2019, 17, 1272–1282. [CrossRef]
13. Peng, Y.; Luo, X.; Wei, W. A New Fuzzy Adaptive Simulated Annealing Genetic Algorithm and Its Convergence Analysis and Convergence Rate Estimation. *Int. J. Control. Autom. Syst.* 2014, 12, 670–679. [CrossRef]
14. Sands, T. Development of Deterministic Artificial Intelligence for Unmanned Underwater Vehicles (UUV). *J. Mar. Sci. Eng.* 2020, 43, 59–69. [CrossRef]
15. Mina, T.; Singh, Y.; Min, B.-C. Maneuvering Ability-Based Weighted Potential Field Framework for Multi-USV Navigation, Guidance, and Control. *Mar. Technol. Soc. J.* 2020, 19, 40–58. [CrossRef]
16. Luo, S.; Singh, Y.; Yang, H.; Bae, J.H.; Dietz, J.E.; Diao, X.; Min, B.C. Image Processing and Model-Based Spill Coverage Path Planning for Unmanned Surface Vehicles. In Proceedings of the OCEANS 2019 MTS/IEEE SEATTLE, Seattle, WA, USA, 27–31 October 2019.
17. Singh, Y.; Bibuli, M.; Zereik, E.; Sharma, S.; Khan, A.; Sutton, R. A Novel Double Layered Hybrid Multi-Robot Framework for Guidance and Navigation of Unmanned Surface Vehicles in a Practical Maritime Environment. *J. Mar. Sci. Eng.* 2020, 9, 624. [CrossRef]

18. Sun, X.; Wang, G.; Fan, Y.; Mu, D.; Qiu, B. A Formation Collision Avoidance System for Unmanned Surface Vehicles With Leader-Follower Structure. *IEEE Access* 2019, 7, 24691–24702. [CrossRef]

19. Li, S.; Liu, J.; Negenborn, R.R. Distributed coordination for collision avoidance of multiple ships considering ship maneuverability. *Ocean Eng.* 2019, 181, 212–226. [CrossRef]

20. Mu, Z.; Xu, W.; Liang, B. Avoidance of Multiple Moving Obstacles during Active Debris Removal Using a Redundant Space Manipulator. *Int. J. Control. Autom. Syst.* 2017, 15, 815–826. [CrossRef]

21. Xia, Y.; Na, X.; Sun, Z.; Chen, J. Formation control and collision avoidance for multi-agent systems based on position estimation. *ISA Trans.* 2016, 61, 287–296. [CrossRef] [PubMed]

22. Vu, M.T.; Van, M.; Bui, D.H.P.; Do, Q.T.; Huynh, T.-T.; Lee, S.-D.; Choi, H.-S. Study on Dynamic Behavior of Unmanned Surface Vehicle-Linked Unmanned Underwater Vehicle System for Underwater Exploration. *Sensors* 2020, 20, 1329. [CrossRef] [PubMed]

23. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 1982, 43, 59–69. [CrossRef] [PubMed]

24. Kohonen, T.; Kaski, S.; Lagus, K.; Salojarvi, J.; Honkela, J.; Paatero, V.; Saarela, A. Self-organization of a massive document collection. *IEEE Trans. Neural Netw.* 2000, 11, 574–585. [CrossRef] [PubMed]

25. Skjetne, R.; Fossen, T.I.; Kokotović, P.V. Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory. *Automatica* 2005, 41, 289–298. [CrossRef]

26. Fossen, T.I. How to incorporate wind, waves and ocean currents in the marine craft equations of motion. In Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano, Italy, 19–21 September 2012; pp. 126–131.