applied sciences

Article

Autonomous Path Planning of AUV in Large-Scale Complex Marine Environment Based on Swarm Hyper-Heuristic Algorithm

Dunwen Wei *, Feiran Wang and Hongjiao Ma

School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China

* Correspondence: weidunwen@uestc.edu.cn

Received: 30 May 2019; Accepted: 28 June 2019; Published: 29 June 2019

Featured Application: This paper presents a new path planning model that combines the global path planning and the local path planning for the large-scale complex marine environment. Meanwhile, the online learning swarm hyper-heuristic algorithm (SHH) is proposed to solve this model with real-time performance and stability.

Abstract: Autonomous underwater vehicles (AUVs) as an efficient underwater exploration means have been used to perform various marine missions. However, limited by the technologies of underwater acoustic communications and intelligent autonomy, the most current and advanced AUVs only perform a limited number of tasks in the small-scale area and the known underwater environment. Therefore, in this paper, a one path planning model was proposed combining the global path planning and the local path planning for the large-scale complex marine environment. More specifically, the B-spline curve was used to represent the smooth path for the requirement of kinematic constraints of AUVs. After considering the various constraints, such as the energy/time consumption, the turning radius limitation, the marine environment, and the ocean current, the path planning was abstractly modeled as a multi-objective optimization model with the time cost, the curvature cost, the map cost, and the ocean current cost. The swarm hyper-heuristic algorithm (SHH) with the online learning ability was proposed to solve this model with real-time performance and stability. The results showed that the proposed online learning SHH algorithm had obvious advantages in terms of time efficiency, stability, and optimal performance compared with the results of two traditional heuristic algorithms, both particle swarm optimization (PSO) and firefly algorithm (FFA). The time efficiency of the online learning SHH algorithm improved at least 20% compared with PSO and FFA.

Keywords: AUV; path planning; swarm intelligence; swarm hyper-heuristic algorithm; online learning

1. Introduction

In recent years, many scholars and engineers are committed to developing various autonomous underwater vehicles (AUVs) because of the increasing application demands [1], such as marine environment exploration [2], marine resource monitoring [3], pelagic survey [4], seabed resource exploitation [5,6], military mission [7], and so forth. With the rapid development of more available advanced technologies, such as processing capabilities and high-density power supplies, more and more AUVs are now indeed being used for executing those high risk and repetitive tasks instead of people [8]. With the roles and missions constantly evolving, AUVs, instead of people performing missions in the unknown large-scale complex underwater environment, are an inevitable trend because of AUVs' inherent advantages. However, until recently, limited by its autonomy and acoustic communication abilities, AUVs can only conduct and implement a limited number of tasks in the offshore area in a relatively known environment. AUVs are needed for higher performance on autonomy and intelligence, such as decision-making planning, situational awareness, path planning, and simultaneous localization and mapping (SLAM), etc [9]. Path planning technology of AUVs is a significant technology that improves the AUVs' autonomy and promotes their applications in the large-scale complex marine environment [10].

It is desirable for a path planner to have an effective path planning model and an online learning algorithm to adapt the model's complexity. On the one hand, it is still a challenge to build this effective path planning model that satisfies the various constraints, such as the energy/time consumption, the turning radius limitation, the marine environment, and the ocean current. On the other hand, real-time performance and stability of planning the path is also an important issue. In fact, the autonomous path planning problem can be recognized as a search problem under multiple constraints, which principally includes two sub-questions: How to represent and model the path, and what algorithms efficiently find the optimal path. The following part first summarizes the commonly used path representation methods and then summarizes the various path planning algorithms.

There are many ways to represent the motion path of AUVs in the underwater environment. Common representation methods are straight lines, Dubin-curves, polynomial curves, and spline curves. The straight-line representation [11] treats the path as a series of segments that are connected end to end. This representation is simple, but the line segments need to be smoothed for the kinematic constraints and controllability of AUVs. The Dubin-curve representation [12] uses a circle for smoothing straight-lines at the line segment connection, which is widely used in the field of mobile robots [13], but the Dubin-curve representation is computationally intensive and is not suitable for real-time path planning [14]. The polynomial and spline curves can represent more complex and smooth paths [15,16] using fewer parameters (or control points), which have been increasingly used in robot path representations in recent years. In this paper, the B-spline curve is proposed to represent the smooth path for the requirement of kinematic constraints of AUVs.

The path planning algorithm can be simply divided into two categories according to the optimality of the solution result. One class emphasizes the optimality of the solution results. These algorithms are represented by the graph search algorithms, while the other class emphasizes that the optimal feasible solutions are obtained with the greatest possibility under reasonable computational costs, such as sample-based algorithms, artificial potential field (APF), bioinspired meta-heuristics [17], and artificial intelligence based (AI-based) algorithms. Li et al. [18] also summarized AUV path planning methods based on the geometric model search algorithms, probabilistic sampling algorithms, artificial potential field algorithms, and intelligent algorithms.

The graph search algorithm is a classic path planning method in the robotic field. The graph search algorithm uses the D or A* algorithm and their various variants to search for the optimal path between the start point and the goal point by representing the environment as a raster map or a topological map with a smaller amount of data. Arinaga et al. [19] applied the D algorithm to the global path planning problem of AUVs in the underwater environment, and the algorithm generated a path without collision. Carroll et al. [20] represented the marine environment as a quadtree map and used the A* algorithm to search for the optimal path in the map. Carsten et al. [21] applied the 3D Field D* algorithm to solve the three-dimensional space path planning problems.

The sampling-based method mainly includes the probabilistic roadmap algorithm (PRM) and random trees algorithm (RRT) [22] and its various variants, which are used to quickly obtain feasible solutions for complex path planning problems. The PRM algorithm first generates a network graph by a series of random position points, and then uses the A* algorithm to search for the path from the start point to the goal point. McMahon et al. [23] successfully applied the PRM algorithm to the path planning problem under the complex constraints of AUVs in a 3D underwater environment and carried out experimental verification. RRT uses a special incremental sampling search method, which is more efficient than PRM in single-request planning and has a good performance in high-dimensional space planning problems. Carreras et al. [24] used the RRT* algorithm in the path planning problem of AUV in a real-time two-dimensional environment. The simulation results showed that the method had excellent adaptability in the real marine environment.

The artificial potential field method mainly includes APF and BUG algorithms, and this kind of algorithm is suitable for real-time obstacle avoidance. Kruger et al. [25] modeled the AUV path planning problem as an optimization problem under multiple constraints and solved the model by APF algorithm. Putra et al. [26] used the BUG algorithm for AUVs local emergency obstacle avoidance and performed the simulation verification.

AI-based algorithms (or biological heuristic algorithms) mainly include particle swarm optimization (PSO) [27,28], ant colony algorithms (ACO) [29], genetic algorithms (GA), tabu search (TS) algorithms, or bioinspired self-organizing maps [30]. In recent years, such algorithms have drawn increasing attention to the solution of AUVs path planning problems [31]. Sun et al. [32] applied the PSO algorithm to the AUV path planning problem. The simulation results showed that the algorithm was simple and easy to implement, with high robustness and fast convergence. Wang et al. [33] proposed an adaptive ACO algorithm to solve the AUV path planning problem, which solved the shortcomings of slow convergence and easy to fall into the local optimal solution. Alvarez et al. [34] applied a genetic algorithm (GA) to the AUV path planning problem to minimize the energy consumption of the path considering ocean current information.

Differing from these omnidirectional ground robots and quadrotors, most of the AUVs are typical underactuated systems, which only rely on the propeller of the tail to generate power. The search method for discrete paths by rasterizing environmental maps is difficult to apply in underactuated AUVs. The AUV path planning method puts forward the higher requirements on the smoothness of planned paths. The path not only needs to be smooth, but also cannot exceed its maximum turning radius of the kinematic constraints and controllability of AUVs. The search-based algorithm has a good performance in the case of fewer constraints and a small-size problem. However, once the problem constraints increase or the problem size becomes larger, the search efficiency will drop sharply, which is not conducive to the practical application. Although the artificial potential field method can quickly solve the optimization problem under multi-constraint conditions, it is easy to fall into the local optimal solution and sacrifice the stability. AI-based algorithm solves the path planning problem, multiple constraints can be considered in the optimization model, and the better feasible solutions satisfying the constraints are obtained in an acceptable time. However, real-time performance and robustness of such intelligent algorithms still seriously restrict the extension and utilization of AI-based algorithms.

In this paper, a path planning model was proposed to implement the application in the large-scale complex marine environment, and the online learning swarm hyper-heuristic algorithm was proposed to meet the requirements of the real-time performance and robustness. There are two main contributions in this paper. Firstly, the proposed path planning model which combines the global path planning with the local path planning is suitable for the large-scale complex marine environment. After considering the various constraints, such as the energy/time consumption, the turning radius limitation, the marine environment, and the ocean current, the path planning was abstractly modeled as a multi-objective optimization model with the time cost, the curvature cost, the map cost, and the ocean current cost. Secondly, the swarm hyper-heuristic algorithm (SHH) with the online learning ability was proposed to solve the proposed path planning model with real-time performance and stability.

The structure of this article is as follows. The autonomous decision-making system is described in Section 2. Then, in the following Section 3, the modeling method of AUV path planning is introduced. Following this, the swarm hyper-heuristic algorithm is proposed to solve this problem in Section 4. Finally, the results of different algorithms are compared and analyzed to verify the correctness and effectiveness of the model.

2. Overview of Autonomous Decision-making System

With the aim of improving the AUVs' autonomous ability, this study proposed and developed the autonomous decision-making system (ADS) of AUVs. As shown in Figure 1, the proposed ADS comprises five closely related components, which includes the decision layer, the navigation layer, the control layer, the monitoring unit, and the communication unit.

The decision layer is responsible for advanced decision-making and task management, such as task scheduling, emergency control, and permission management. The navigation layer provides path planning, environment mapping, and situation awareness to support autonomous navigation. In the navigation layer, the AUV uses the electronic chart data to conduct the online path planning under the multi-constraints, and according to the real-time data obtained by various sensors to construct the environment map and re-plan the path. The control layer is used to generate control commands that control the AUV traveling along the planned path or uses reactive behavior to obtain real-time responsiveness of the AUV system. The communication unit is a bridge for data exchange between the AUV and the command center system.

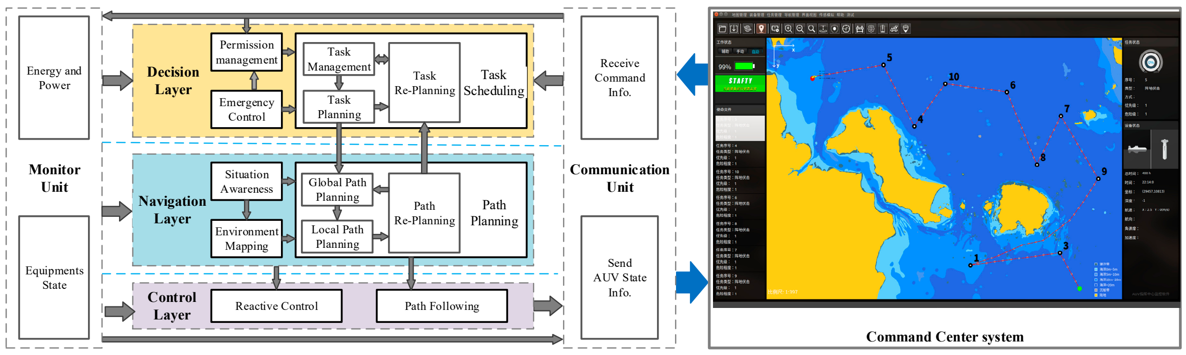

Figure 1. The system architecture of autonomous decision-making system (ADS).

The navigation layer can plan the voyage paths according to the AUV's missions and the environmental information. At present, many available AUVs still need to download the workflow for a certain task to the control computer with a pre-programmed form before the start of the task. At the stage of task execution, human intervention is required at any time to ensure the security of the AUV and the reliability of the task execution. This traditional operation procedure cannot handle large-scale, highly dynamic and complex marine environments, and has low adaptability to new environments, which can easily lead to the loss of the AUV or the failure of mission's execution. Advanced AUVs are often equipped with expensive sensors so that the price and maintenance costs of AUVs are relatively high, and the loss or failure to complete tasks is unacceptable.

Therefore, AUVs must have the ability to adapt to the large-scale, highly dynamic marine environments to ensure their own safety and the accuracy of mission's completion. Planning is an effective way to deal with complex environments. Path planning and path re-planning generate the trajectory of the AUV navigation to ensure collision-free, safe and energy-saving and to adapt to dynamic and complex marine environments. The path planning achieves the autonomy of the AUV, ensuring the security of the AUV and the reliability of task execution. The following is an analysis of the constraints that need to be considered for AUV navigation path planning.

Trajectory dynamics constraints. When the actual AUV travels, there are dynamic constraints, such as the maximum radius of gyration, the maximum speed, the maximum acceleration, etc. The planned path should satisfy these constraints, which requires adding corresponding constraints in the algorithm to limit the generation space of the track. The path generated by the local path planning module should satisfy the dynamic constraints, and there should be no turning radius which is too small or other conditions that do not satisfy the dynamic constraints.

Obstacle and current constraints. Obviously, the path planned by the path planning algorithm needs to meet the collision-free constraint. In addition, the motion of the AUV should also take into account the influence of the water flow. The tangential water flow will cause the AUV to deviate from the original path and result in motion errors. The reverse flow will increase energy loss. Therefore, the path planning algorithm should plan as far as possible to avoid the collision and meet the path of certain water flow constraints.

3. Modeling of Path Planning

The conventional path planning method mainly relies on discretizing the environment and using the A* and other search algorithms to obtain the path between the start point and the end point. This obtained path is the absolute shortest path at the high resolution of the environment map. Although the path obtained by the A* and other search algorithms is the absolute shortest path, either it is not necessarily a smooth path due to the low-resolution environment map, or those methods would consume much time to search in the high-resolution map. It is difficult to track the unsmooth path for AUVs due to its ontology structure design. In recent years, many scholars have begun to study the path planning method that satisfies the requirements of the smoothness, and the shortest path simultaneously. In this section, the A* algorithm was first used to find the global path, and then modeled the local path planning based on the global path planning, considering the time, turning radius, obstacles, and ocean current constraints. The authors provided a detailed description of the problem mathematical definition and the optimization criteria for the local path planning.

3.1. Global Path Planning

The global path planning in the navigation layer generates a global network node path for AUV navigation based on the task execution sequence generated by the decision layer. The task execution sequence generated by the decision layer has three disadvantages for the following path planning. Firstly, the generation procedure of task sequence does not consider the influence of environment in detail. This will result in the length of the planned path being larger than the distance of two task points. Secondly, the distances between two task points are usually too long, especially for the scattered tasks in the large-scale marine environment. This characteristic will lead to the long computing time of calculating the optimal path and go against real-time planning and re-planning. Thirdly, the distances between two task points largely differ so that it brings difficulties to the real-time path planning and path re-planning.

Therefore, in the global path planning stage, this study expressed the task execution sequence as one task network graph $G^T = (V^T, E^T)$. V^T is the task node set of the task network graph (including one start point, one goal point, and lots of task points). E^T indicates the edge set of the task network graph. Some discrete intermediate points with roughly uniform length are randomly added to the task node set V^T. Combined with the task node set V^T, one path node set V^P and one path network graph $G^P = (V^P, E^P)$ can be generated. The new edge set E^P can be easily obtained by trying to connect all the added nodes by straight lines. If the straight line connecting two nodes is passed through the land or danger area, the A* algorithm is used to search for the shortest path in the path network graph. The A* algorithm should not take much time to explore the shortest path in this path network graph since it does not concern itself about other constraints, except for the environment. However, it can obtain a discrete path with the shortest distance. The distance between two arbitrary path points is approximately the same that will benefit the real-time path planning and path re-planning. By this way, this global path planning overcomes the above three disadvantages of the task execution sequence generated by the decision layer. As shown in Figure 2, the left picture is the task execution sequence generated by the task planning algorithm, and the right picture is the refined global planning path generated by the discrete path points with uniform spacing distances.

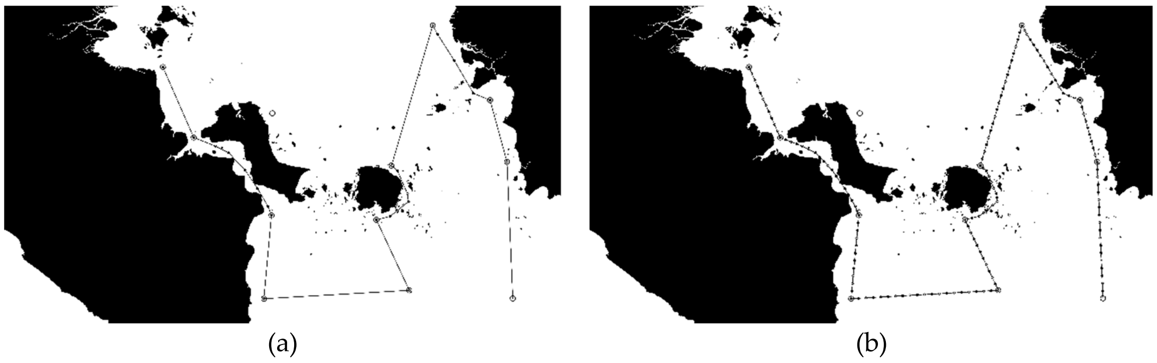

Figure 2. The global path planning results (**a**) the task execution sequence after task planning, and (**b**) the refined global path after global path planning.

3.2. Mathematical Definition of the Problem

The sailing of the AUV is easily affected by ocean currents due to its small inertia. On the one hand, the ocean current will interfere with the navigation of AUVs, causing its navigation drift. On the other hand, the rational use of ocean currents can assist AUVs to reduce energy consumption. Considering two extreme cases, if the AUV's navigation direction is consistent with the ocean current direction, the ocean currents will provide the positive work for the AUV's navigation. In this case, the ocean current assists the AUV to sail. When the AUV's navigation direction is opposite to the ocean current, in such case the ocean currents do negative work to the AUV, and the currents hinder the AUV's sailing. Therefore, if the ocean current information of the known environment is considered in the local path planning stage, the influence of ocean currents on the AUV navigation can be utilized to some extent to assist the AUV navigation.

The ocean current information is modeled as a two-dimensional ocean map, which represents the seawater flow information. This simplification is reasonable because the variation in the vertical direction is small relative to the horizontal size scale, and the movement of the water flow in the vertical direction is much smaller than the movement in the horizontal direction (because of the rotation of the earth). Under these two assumptions, the ocean dynamics model in the horizontal plane is described by the two-dimensional Navier- Stokes equation [35]:

$$\frac{\partial \omega}{\partial t} + \left(\vec{V}\nabla\right)\omega = \nu \Delta \omega \tag{1}$$

where $\vec{V} = (V_x, V_y)$ is the velocity field, ν is the fluid viscosity, $\omega = \frac{\partial V_y}{\partial x} - \frac{\partial V_x}{\partial y}$ is the vorticity, ∇ and Δ are the gradient operator and Laplace operator respectively. However, this model is computationally intensive and cannot meet real-time requirements. In practice, Navier- Stokes can be expressed as a superposition of single-point vortices, namely the viscous Lamb vortex. The mathematical expression of the Lamb vortex is shown as:

$$\begin{aligned}
V_x(\vec{r}) &= -\Gamma \frac{y-y_0}{2\pi(\vec{r}-\vec{r}_0)^2}\left[1-e^{-\left(\frac{(\vec{r}-\vec{r}_0)^2}{\delta^2}\right)}\right] \\
V_y(\vec{r}) &= \Gamma \frac{x-x_0}{2\pi(\vec{r}-\vec{r}_0)^2}\left[1-e^{-\left(\frac{(\vec{r}-\vec{r}_0)^2}{\delta^2}\right)}\right] \\
\omega(\vec{r}) &= \Gamma \frac{1}{\pi\delta^2} e^{-\frac{(\vec{r}-\vec{r}_0)^2}{\delta^2}}
\end{aligned} \tag{2}$$

where $\vec{r}_0$ is the center position vector of the vortex; Γ and δ are the intensity and radius parameters of the vortex. The physical model represented by the Equation (2) is used to analyze the velocity field of the surrounding water flow environment [35].

The position, radius, and velocity of the Lamb vortex in practice are analyzed using data obtained by the horizontal acoustic doppler current profiler (H-ADCP). The number of Lamb vortices and the center of the vortex are estimated using the following method. Assuming that the velocity measured by H-ADCP is the tangential velocity produced by the nearest vortex, the vertical direction of this velocity is the radial direction of the vortex, using adjacent velocity information can obtain the center of the vortex. Figure 3 shows a schematic diagram of a water flow Lamb vortex.

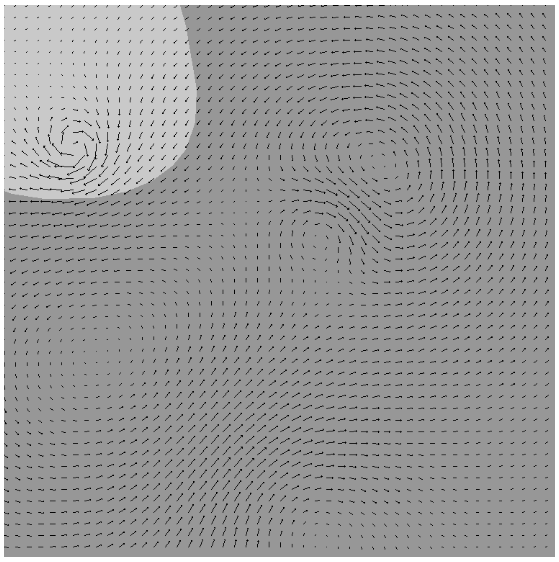

Figure 3. The Lamb vortex diagram.

Since the AUV rarely changes its depth when navigating in large-scale marine, in this paper, the authors only considered the two-dimensional path of AUV. The B-spline curve was used to represent the two-dimensional local path of AUV. Compared with the Bezier curve, the number of control points of the B-spline curve is independent of the order of the curve. The local characteristics of the curve can be adjusted by adjusting the position of the local control point. This explains why it is more and more applied in the robot path representation. The position of the control points can be defined as $C = \left\{c_i = (\vartheta_x^i, \vartheta_y^i)\middle|c_i \in \mathrm{R}^2, i = 1, 2\ldots, n\right\}$. The curve order is K and $B_{i,K}(t)$ is the basic function of the B-spline curve, then an AUV path can be expressed in the form of the equation:

$$\begin{cases} X(t) = \sum\limits_{i=1}^{n} \vartheta_x^i \times B_{i,K}(t) \\ Y(t) = \sum\limits_{i=1}^{n} \vartheta_y^i \times B_{i,K}(t) \end{cases} \tag{3}$$

Therefore, the local path planning problem of AUV can be described as finding the control points of the B-spline curve, so that the obtained curve path satisfies the non-collision, the whole path satisfies the minimum turning radius constraint, and the path direction follows the direction of the ocean current as much as possible. This problem can be modeled as an optimization problem under multiple constraints, using intelligent algorithms.

3.3. Optimization Criteria

The optimization cost functions are the functions related to the B-spline path. Given a curve path, the cost function can be used to calculate a comprehensive cost value that combines the time cost, the path curvature, the environment map, and the ocean current. A discrete B-spline path $\wp$ can be expressed as the set comprised by the discrete points shown in the formula (4).

$$\wp = \left\{\wp^1, \ldots \wp^i, \ldots \wp^h \middle| 1 \leq i \leq h, \wp^i = \left(X^i, Y^i, \psi^i\right), \wp^i \in \mathbf{R}^3\right\} \tag{4}$$

where h is the number of discrete points on the path; X^i, Y^i are the position of the AUV at the i discrete point; ψ^i is the direction of the AUV at the i discrete point, also named as the yaw angle, which can be calculated using the following equation:

$$\psi^i = \tan^{-1}\left(\frac{\left|Y^{i+1} - Y^i\right|}{\left|X^{i+1} - X^i\right|}\right) \tag{5}$$

The time cost is one optimization criteria that requires the shortest sailing time, and usually is proportional to energy consumption. The time cost is obtained by dividing the total length of the accumulated path by the average velocity, as shown in the Equation (6).

$$T_{\cos t} = \sum_{i=1}^{h-1} \frac{\left|\wp_{x,y}^{i+1} - \wp_{x,y}^{i}\right|}{|v|} \tag{6}$$

where h is the total number of points in the discrete path, and v is the average velocity. The time cost is used to constrain the traveling time. Minimizing the time cost under certain conditions can result in a path that has the shortest distance and satisfies certain conditions.

The curvature cost is obtained by accumulating the approximate curvature at each discrete point on the path, as shown in the Equation (7).

$$\rho_{\cos t} = \sum_{i=1}^{h-1} \frac{\left|\wp_{\psi}^{i+1} - \wp_{\psi}^{i}\right|}{\left|\wp_{x,y}^{i+1} - \wp_{x,y}^{i}\right|} \tag{7}$$

The curvature cost is used to constrain the spatial curvature of the path, that is, the turning radius (curvature radius) on the path. The curvature and the turning radius are inversely proportional. A legal path should not have a turning radius less than the minimum turning radius, and the path that satisfies the AUV turning radius constraint can be obtained by minimizing the curvature cost.

The map cost is obtained by the map information at each discrete point on the path, as shown in the Equation (8). If the point is on a map danger area (land, island, etc.), the map cost for that point is 1, otherwise 0. Minimizing the cost of the map is used to constrain the path from colliding with the danger area of the map and ensure the non-collision of the path.

$$M_{\cos t} = \sum_{i=1}^{h} \begin{cases} 1 & \wp_{x,y}^{i} \cap DangerArea \\ 0 & otherwise \end{cases} \tag{8}$$

The ocean current cost is one optimization criteria obtained by accumulating the dot product of the ocean current vector and the path direction vector at each discrete point on the path, as shown in Equation (9)

$$O_{\cos t} = \sum_{i=1}^{h} \left|\vec{v}_{\wp_{x,y}^{i}}\right| \cdot \theta\left(\vec{v}_{\wp_{x,y}^{i}}, \vec{\tau}_{\wp_{x,y}^{i}}\right) \tag{9}$$

where $\vec{v}_{\wp_{x,y}^{i}}$ is the ocean current vector at the point $\wp_{x,y}^{i}$; $\vec{\tau}_{\wp_{x,y}^{i}}$ represents the path unit direction vector at the point $\wp_{x,y}^{i}$; $\theta(\cdot)$ is the angle between the two vectors $\vec{v}_{\wp_{x,y}^{i}}$ and $\vec{\tau}_{\wp_{x,y}^{i}}$.

The path planning is a multi-objective optimization problem. The optimal path evaluation criterion includes the shortest path time T_{cost}, the smallest cumulative curvature ρ_{cost}, the map cost

M_{cost}, and the ocean current cost O_{cost}. The multi-objective optimization problem can be converted into a single-objective optimization problem by summing the weighted four cost value, as shown in the following equation:

$$\min \wp_{\mathrm{cost}} = \omega_1 \cdot T_{\mathrm{cost}} + \omega_2 \cdot \rho_{\mathrm{cost}} + \omega_3 \cdot M_{\mathrm{cost}} + \omega_4 \cdot O_{\mathrm{cost}} \tag{10}$$

where $\omega_1,\ \omega_2,\ \omega_3,\ \omega_4 \in \mathbf{R}$ are the weights of different cost values respectively. The values of weights are determined by the expert system and application demands.

4. Swarm Hyper-heuristic Algorithm

4.1. Swarm Hyper-heuristic Algorithm

As shown in Figure 4, the hyper-heuristic algorithm is an advanced type of intelligent computing method appearing in the early 2000 s, which provides a high-level strategy (HLS) that manipulates or manages a set of low-level heuristics (LLH) to improve the searching efficiency and performance. The hyper-heuristic algorithm works in a higher abstraction layer than the meta-heuristic algorithm, and the hyper-heuristic algorithm selects which low-level heuristic method to use at a particular moment through state of the problem solving [36]. Swarm hyper-heuristics algorithm searches a kind of good solution to the problem rather than the best solution of the problem directly, which is independent of the specific problem domain or background issues.

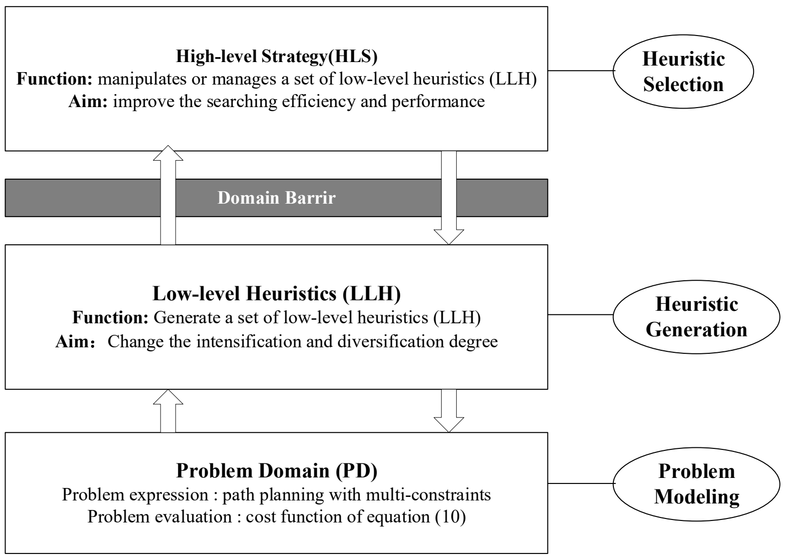
High-level Strategy(HLS)
Function: manipulates or manages a set of low-level heuristics (LLH)
Aim: improve the searching efficiency and performance
Heuristic Selection
Domain Barrir
Low-level Heuristics (LLH)
Function: Generate a set of low-level heuristics (LLH)
Aim: Change the intensification and diversification degree
Heuristic Generation
Problem Domain (PD)
Problem expression : path planning with multi-constraints
Problem evaluation : cost function of equation (10)
Problem Modeling

Figure 4. Hyper-heuristic framework and composition.

The swarm intelligence, derived from the observation of insect swarms in nature, is the behavior characteristics exhibited by swarm organisms through cooperation. Swarm intelligence is an increasing focus in the field of metaheuristic algorithms, and the efficiency, and effectivity in solving complex problems has drawn much attention in recent years. However, some scholars believe that these similar algorithms lack novelty. Some algorithms have similar operations, by taking the different names and simulating the characteristics of another natural organism. The performance of the metaheuristic algorithm depends on how the algorithm balances two basic search mechanisms, intensification and diversification. The intensification gets the algorithm to perform a detailed search in the local, and the diversification prevents the solution from entering the local optimum prematurely. These two basic mechanisms include a variety of group operations. For example, fireflies' movements according to the light intensity and lightness in the firefly algorithm are intensification, and the random movement of fireflies is diversification.

Therefore, Tilahun et al. [37] proposed a swarm hyper-heuristic algorithm framework, trying to integrate the swarm meta-heuristic algorithm with a general hyper-heuristic framework, where the updating operators were recognized as low-level heuristics and guided by a high-level hyper-heuristic. Different learning methods are used to determine the intensified and diversified behavior of the algorithm. According to whether or not to learn, it can be divided into three categories, no learning SHH1, offline learning SHH2, and online learning SHH3. According to the conclusion of [37], the performance of the online learning method SHH3 was better than the other methods.

The swarm-based meta-heuristic algorithm is a swarm-based algorithm, in which the individual in the swarm interacts with each other through different update operators and achieves their own updates. The efficiency and effectivity of swarm hyper-heuristic algorithms are determined by two steps, including the heuristic selection and the heuristic generation. The heuristic selection is the method for choosing or selecting the appropriate heuristic at each iteration, while the heuristic generation is the procedure that generates various heuristics [36].

4.1.1. Heuristic Generation

Heuristic generation is the procedure that generates various heuristics. A heuristic, usually named as an operator, is one operation procedure that inputs a solution and outputs a new solution after an iteration. Assuming an operator is expressed as $O\langle\cdot\rangle$, the new solution x^{t+1} is the map using the operator $O\langle\cdot\rangle$ after receiving an input x^t, which can be expressed as the equation:

$$x^{t+1} = O\left\langle x^t \right\rangle \tag{11}$$

To a certain extent, the performance of the algorithm depends on these heuristic operators. Many scholars carried out related research and proposed new heuristic operators for heuristic algorithms. Swarm intelligent operators mimic the different swarming behaviors of different swarm organisms. Some outstanding examples are the foraging swarming behavior from ants (the ant colony optimization) or flies (fruit fly optimization algorithm), and the gathering behavior from fireflies (firefly algorithm) or masses (binary gravitational search algorithm such as ants). However, some heuristic operators have similar operations, just taking the different names and simulating the characteristics of another natural organism. The widely used operators are summarized in the following Table 1. There may be more types of updating operators in the future, but the swarm hyper-heuristic algorithm framework does not depend on the specific type of operator, and it is convenient to add new types of operators. Various operators comprise an operation set of operators, denoted as $\mathbf{O} = \{O_1, O_2, \ldots, O_k\}$. Different operators in this operation set have the different searching characteristics of increasing the degree of diversification or intensification. The diversified operator provides the search strategy away from the explored neighborhood region, whereas an intensified operator accelerates the convergence to the promising area.

Table 1. Some widely and recently used operators.

Operator Name	Operator Formula	Note to Explain
Random move in the neighborhood	$x_i := x_i + \lambda_{\min} \cdot rand \cdot u$	$\lambda_{\min}$ is an intensification step length
Following better solutions	$x_i := x_i + \lambda \cdot rand \cdot \left(x_j - x_i\right)$	$\forall x_j, f\left(x_j\right) \leq f(x_i)$
Following the best solution	$x_i := x_i + \lambda \cdot rand \cdot (x_b - x_i)$	x_b is the best solution in population
Following own best	$x_i := x_i + \lambda \cdot rand \cdot \left(x_i^{best} - x_i\right)$	x_i^{best} is the best performance of the solution x_i in history
Random long jump	$x_i := x_i + \lambda_{\max} \cdot u$	$\lambda_{\max}$ is a diversification step length
Mutation	$x_i := m(x_i)$	m is a variation function
Run away from the worst	$x_i := x_i + \lambda \cdot rand \cdot (x_i - x_w)$	x_w is the worst solution in population
Run away from worse solution	$x_i := x_i + \lambda \cdot rand \cdot \left(x_i - x_j\right)$	$\forall x_j, f\left(x_j\right) \geq f(x_i)$
Improving local search	$x_i := x_i + \lambda_{\min} \cdot rand \cdot u$	$u = \text{argmin}\{f(x_i + \lambda_{\min} \cdot rand \cdot u)$ $: u \in \left\{u_1, u_2, \ldots, u_i, \ldots, u_m, \vec{0}\right\}\}$

4.1.2. Heuristic Selection

Heuristic selection is the procedure of using one high-level strategy (HLS) to manipulate or manage a set of low-level heuristics (LLH). It is noted that the performance of the swarm hyper-heuristic algorithm depends mainly on how to balance the two basic search mechanisms of intensification and diversification. There are two issues for this problem, including how to evaluate the intensification and diversification performance after one iteration and what strategy is used to select the operator according to the intensification and diversification performance. The without learning SHH uses the strategy of giving equal probability to select each operator in each iteration, whereas the online learning SHH has the ability to evaluate probabilities of choosing which operators before or after an iteration. In this paper, online learning SSH was chosen to improve the adaptability. The intensification and diversification of the algorithm are measured by a certain, and then the selection probabilities of the centralized and diversified operations are adjusted so that the intensification and diversification are more balanced.

The degree of intensification is determined by the value of the best cost function before and after one iteration. For a minimization problem, the algorithm is more intensified in this iteration if $f(x_b^{t-1}) > f(x_b^t)$. The degree of diversification can be measured by the sum of the distances between solutions and the central solution, as shown in the following equation:

$$d = \sum_{i=1}^{N} ||x_i - x_c|| \tag{12}$$

where $x_c = \sum_{i=1}^{N} x_i/N$ is the central solution. For a minimization problem, the algorithm is less diversified in this iteration if $d^{t-1} > d^t$.

Therefore, for a minimization problem, the relationship between intensification and diversification after one iteration is shown in Figure 5. In the iterative process, the degree of diversification and intensification of the algorithm should be maintained at a considerable level. If the trend of algorithm diversification is detected to decrease, the selection probability of diversified operator in the algorithm should be appropriately increased.

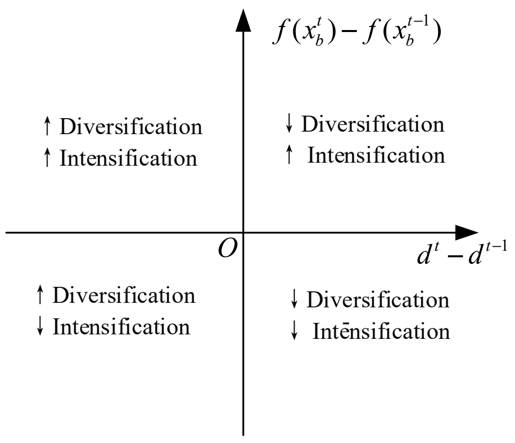

Figure 5. Online learning heuristic selection strategies of swarm hyper-heuristic algorithm.

The change of the degree of intensification or diversification can be achieved by modifying the selected probability of each operator. Consider an increase of the degree of diversification of search behavior, and the probability modifiers for all diversified operations by $P^t(o_i) = \gamma P^{t-1}(o_i)$. The probability modification factor is $\gamma > 1$ corresponding to all diversified operations. Then, the selection probability vector is normalized by $P = P/\sum P$.

4.2. Problem Solving

4.2.1. Initialization of B-spline Curve

In this paper, the path planning method uses the B-spline curve to represent the path of AUVs. The variable control point of a set of B-spline curves is an individual. Let the swarm size be $i_{\max}$, the dimension of each individual is M, then an individual in the swarm can be represented as a set of control points.

$$x_i = \left\{ \left(rp_{1,i}, rp_{2,i}, \ldots rp_{k,i}, \ldots rp_{M,i} \right) \middle| rp_{k,i} \in R^2, 1 \le i \le i_{\max}, 1 \le k \le M \right\} \tag{13}$$

Considering that the path generally does not appear to be distorted and intersected, the order of the control points is generally along with the start point to the goal point. Therefore, in the initialization phase, the equidistant point is generated on the line connecting the start point and the goal point, and the perpendicular line of the line segment is made on each equidistant points to obtain a vertical line, and a random point is generated on each vertical line. Thus, the entire initialized swarm is obtained, and the random point is calculated as shown in equation:

$$\begin{cases} rp_{k,i,x} = cp_{k,x} + dist \times rand + rand \\ rp_{k,i,y} = d_{k,y} - \frac{(rp_{k,i,x} - d_{k,x})}{slope} + rand \end{cases} \tag{14}$$

where *dist* is the end point of the vertical line to the left of the bisector, the distance between the start point S and the goal point D in the direction, *slope* is the slope of the vertical line and *rand* is a random value and *rand* $\in [-0.5, 0.5]$. The swarm obtained by initialization is shown in Figure 6a.

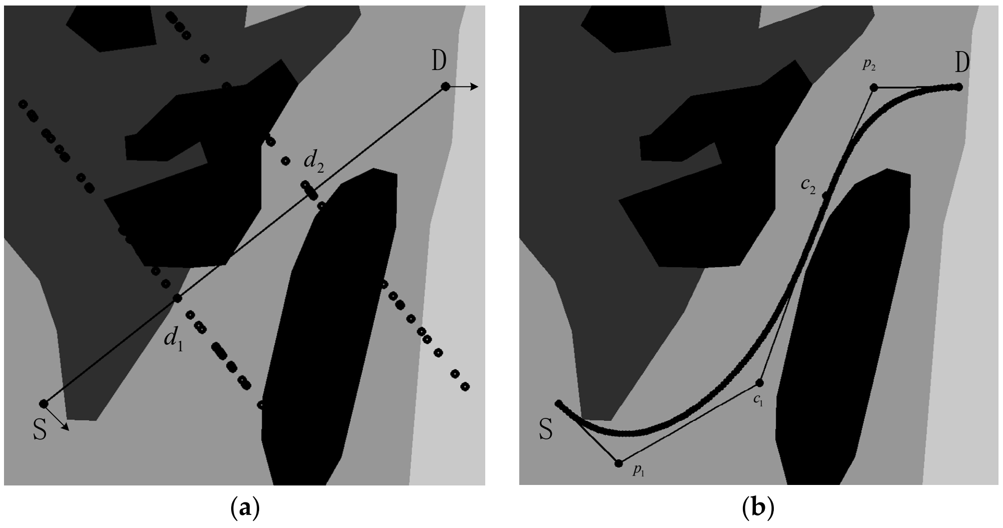

Figure 6. Schematic diagram of B-spline curves (**a**) initializing the swarm (**b**) setting of the fixed control points.

The fixed control point is used to control the start and end directions of the local path curve to be the same as the start and end directions of the AUV. The increase mode is to start from the starting point S and add a fixed control point p_1 along the starting direction of the AUV by a short distance. Starting from the end point D, a fixed control point p_2 is added to a short distance along the opposite direction of the end direction of the robot.

$$\begin{cases} p_{1,x} = S_x + d \cdot \cos\theta_1 \\ p_{1,y} = S_y + d \cdot \sin\theta_1 \end{cases} \quad \begin{cases} p_{2,x} = D_x - d \cdot \cos\theta_2 \\ p_{2,y} = D_y - d \cdot \sin\theta_2 \end{cases} \tag{15}$$

where $S, D \in \mathbf{R}^2$ is the start point and goal point respectively. θ_1 is the starting direction and θ_2 is the end direction. Figure 6b shows the schematic diagram of fixed control points.

4.2.2. Procedure of Optimization

The online learning swarm hyper-heuristic algorithm was used to solve the optimization problem proposed in formula (10). The pseudocode of the path planning using online learning SHH algorithm is shown in Algorithm 1. The method first sets the algorithm parameters, such as the number of iterations $t_{\max}$, the probability modification factor γ, the number of individual swarms $i_{\max}$ and the parameters of the B-spline curve. According to the equations from Table 1, the set of basic operators $\mathbf{O}$ is determined. Each operator O_i has an equal probability of being selected in the first iteration. In this way, the initial value of the selection probability vector P is determined. Then initializing the swarm, an individual x_i is represented by a set of variable control points of the B-spline curve, and the initial generation value of each individual is calculated using the cost function shown in the formula (10). Then, enter the iteration of the swarm until the number of iterations reaches the set value $t_{\max}$, and the optimal individual (the optimal control point) is used to generate the optimal local path. In each iteration step, one operator is selected from the set of basic operations for each individual in turn. The operator is applied to the individual, and the individual's cost value is updated until all individuals in the swarm have completed one operation. After each iteration, the degree of diversification can be evaluated by (12), and the selection probabilities of the basic operators are updated according to the four cases shown in Figure 5, making the trend of intensification and diversification more balanced in the next iteration.

Algorithm 1: Pseudocode of swarm hyper-heuristic algorithm for path planning

```
Input: $S$, $\theta_1$, $D$, $\theta_2$, $i_{max}$, $t_{max}$
Output: Path
1. initialize: Set the parameter of algorithm and B-spline curve;
2. initialize the population basic operation set O, and its selection probability vector P;
3. initialize the population according to the formula (13) and (14).
4. for $t = 1; 2;\quad ; t_{max}$ do
5.     for $i = 1; 2;\quad ; i_{max}$ do
6.         Selects the basic operation corresponding to the maximum value in the population basic operation
           selection probability vector $O_i$;
7:         operates on individual $i$ using the operation $O_i$;
8:         Updates the value of the individual $i$ according to the Equation (10);
9:     end for
10:    Caculates the degree of diversification $d^t$ according to the Equation (12);
11:    According to the Figure 5 to evaluate the degree of intensification and diversification of this iteration,
       and update the operation selection probability vector $P^t$;
12: end for
13: Output optimal Path;
```

5. Results and Discussion

The software running environment was Intel i7-8700K, 3.7 GHz. The operating system was ubuntu 16.04, and the algorithm was implemented by C++ language. The global path planning phase simply generates the discrete path points and obtains the shortest path points according to the A* algorithm, all of which can be completed in a relatively short time. The local path planning uses the B-spline curve to represent the path. Combined with time, map, curvature, and ocean current constraints, the local path planning problem is modeled as the nonlinear optimization problem with multi-constraint conditions shown in the formula (10). Then, the online learning SHH is adopted to solve this problem. In this section, three scenarios with the different marine environment, including only ocean current information, only dangerous area information and both ocean current information and dangerous area information contained, were simulated to test and analyze the running effectiveness of the model and algorithm. The general metaheuristic algorithms, such as particle swarm optimization (PSO) [27], firefly algorithm (FFA) [17], were used in the path planning problem. The FFA algorithm proved to have better performance compared with other algorithms in [17]. In this paper, the results of online learning SHH were compared with the PSO and FFA.

The number of B-spline control points was set to 7, where the number of variable control points was set to 3 and the curve order was set to 3. According to the parameter design method of [37], the number of iterations of the SHH algorithm was set to 80 times. The number of individual swarms was set to 100, random long-distance jump and mutation operations $\lambda_{\max} = 40$, random movement $\lambda_{\min} = 20$, and follow-up operations $\lambda = 0.4$. Referring to the paper [27], the inertia ω, the personal influence c_1 and the social influence parameters c_2 of PSO were set to 0.7298, 1.496, and 1.496 respectively. According to the reference [17], the number of FFA iterations was set to 80, the number of fireflies was set to 100, the attracting factor $\gamma_{FFA} = 0.05$, and the random moving factor $\alpha = 3$. The iterations times and the number of individual swarms of PSO, FFA, and SHH were set to the same value to compare the difference in the performance of the three algorithms under the same conditions.

It is worth noting that in the actual program implementation, the operation set only contained the first 6 operators, and did not contain the last 3 operators. After the inclusion of the runaway operators, the convergence of the algorithm drops sharply, and the results of the convergence cannot occur many times in the test. These two operations were not suitable for solving the optimization problem of the path planning model.

5.1. Simulation Results

5.1.1. Scenario1: Only Ocean Current Information

As mentioned above, the ocean current has a significant influence on the navigation of AUVs, affecting navigation accuracy and energy consumption. Therefore, if the planned path can go in the direction of the ocean current, the ocean current will be used to improve the navigation performance of the AUV. In this part, this study analyzed the planning effect of the SHH algorithm for the scenario that only contained ocean current information. The results of the SHH were compared with the path planning results using two heuristic algorithms, PSO and FFA.

The solid line in Figure 7 was the planning result using the SHH algorithm without considering the influence of the ocean current. Since there was no danger area, the planning result was a straight line with the shortest path. The dashed line was the result after considering the influence of ocean current information. It is evident that the planning path has obvious bending under the influence of the ocean current. Under the condition of ensuring smoothness, the direction of planned path tried to follow the direction of the ocean current as near as possible, which proved that the ocean current constraint in the optimization cost function influenced the planning result.

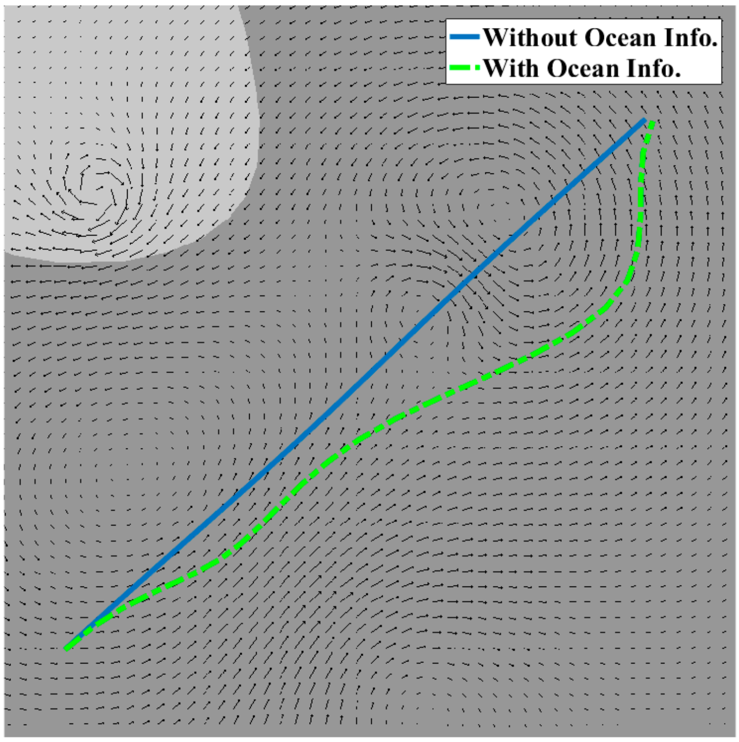

Figure 7. Comparison of the path planning results of the SHH algorithm without and with the ocean current information.

Figure 8 is a comparison of the results of PSO, FFA and SHH algorithms in the case of only ocean current information. The solid line was the result of the PSO; the dash-dot line described the result of the FFA algorithm, and the dashed line was the running result of the SHH algorithm. All of the three algorithms were planning a bending route because of the influence of ocean currents. The results show that all algorithms can generate the local paths that basically conform to ocean current information in scenario 1.

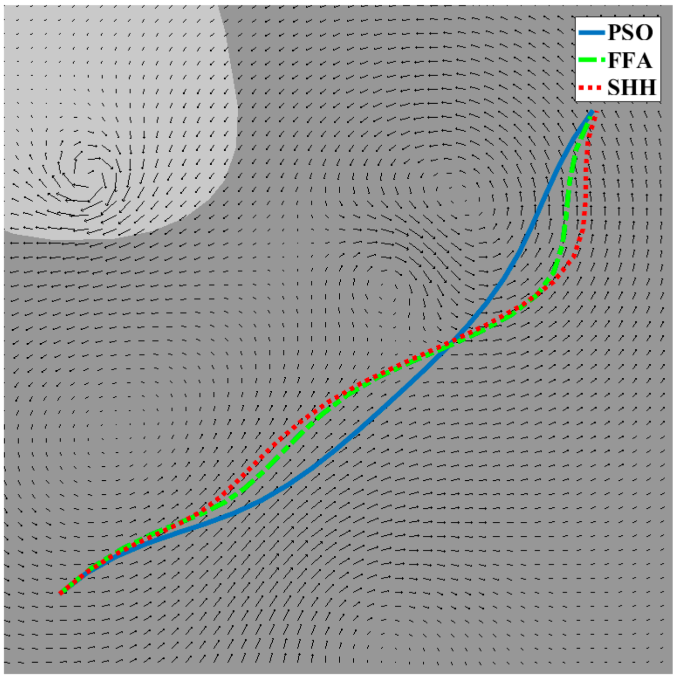

Figure 8. Comparison of the path planning results of PSO, FFA and SHH algorithms in scenario 1.

The results of the three algorithms running 200 times under the same conditions were analyzed. Table 2 is the result comparison of the PSO, FFA, and SHH in the scenario only containing ocean current information. The data in the table were the average values obtained after repeatedly running 200 times under the same conditions. The computing time refers to the interval time from the entry to the completion of the algorithm iteration. The average computing time of PSO, FFA and SHH were 2.2815 s, 1.1834 s, and 0.6214 s respectively. It appeared that the SHH algorithm had the shortest average computing time compared with the PSO and FFA algorithms because the swarm operations of the three algorithms were different. The FFA operation was only the darkness individual moving towards the bright individual. This operation needs to traverse all other individuals when operating one individual. This process is time-consuming. In addition to the operations of following the better individual which need to traverse all other individuals, the other operations can be done in constant time, so it is easily understandable that the SHH algorithm has high computing efficiency. The total cost value of the final optimization of PSO, FFA, and SHH were 998.8257, 701.4505, and 624.668 respectively. The cost functions of the three algorithm evaluation paths were consistent. Therefore, the results obtained by the SHH algorithm had the best cost value. It is visible that the SHH algorithm using multiple operation sets has more advantages in solving the proposed path planning problem. In addition, although the path length obtained by the SHH is longer, it is smoother and more suitable for ocean current constraints.

Table 2. Results comparison using PSO, FFA and SHH algorithms in Scenario 1.

	PSO	FFA	SHH
Computing time	2.2815s	1.1834s	0.6214s
Final total cost	998.8257	701.4505	624.6680
Time cost	171.2734	173.3448	176.5915
Curvature cost	73.0122	153.6648	150.5598
Ocean current cost	754.5393	374.4410	297.5167

5.1.2. Scenario 2: Only Danger Area Information

This section tests the operational effects of the three algorithms in the scenario containing only danger area information (Scenario 2). Figure 9 compares the results of PSO, FFA and SHH algorithms in scenario 2. The cost functions in scenario 2 are time, danger area, and curvature. The path is as short as possible on a smooth and bumpless basis.

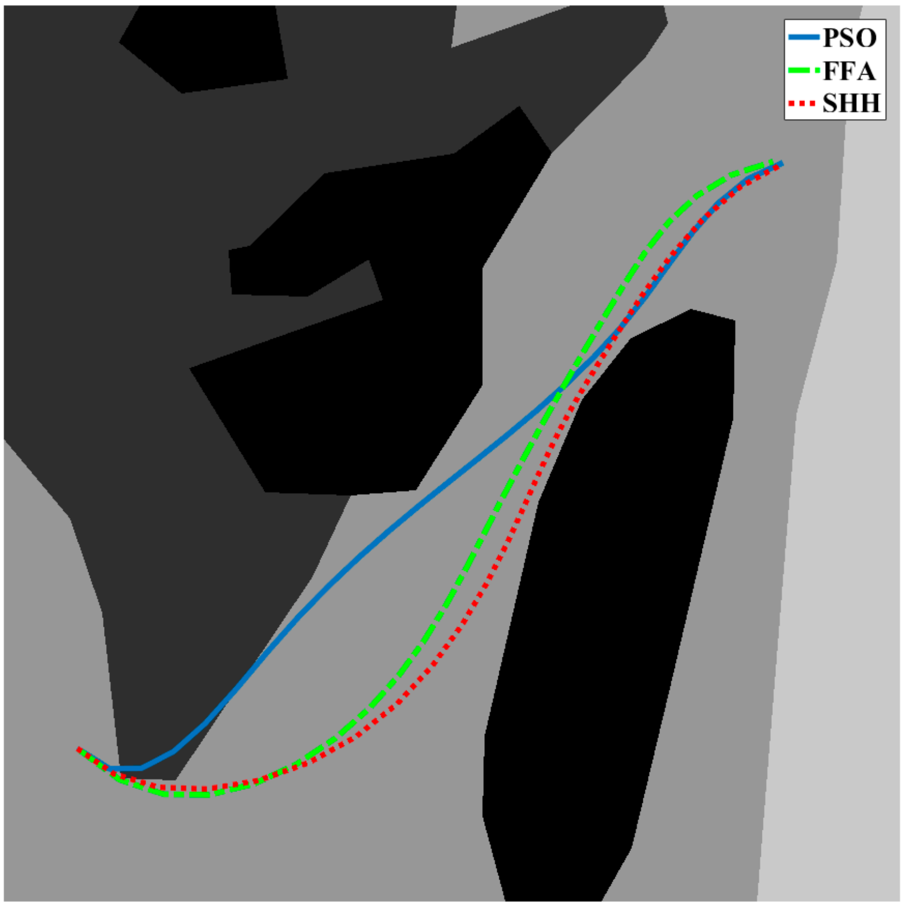

Figure 9. Comparison of the path planning results of PSO, FFA and SHH algorithms in scenario 2.

After running 200 times under the same conditions, the results of the three algorithms were analyzed. Table 3 shows the statistics of PSO, FFA and SHH algorithms in scenario 2 running under fixed conditions for 200 times. In terms of computing time, the SHH also had the greatest advantage, with an average computing time of 0.6217 s, while the average computing time of PSO and FFA were 2.4215 s and 1.2700 s respectively. The final ultimate average cost value of SHH was 306.4266, while the total average cost value of PSO and FFA were 588.3817 and 322.0653 respectively. This indicates the SHH algorithm can find the best optimal performance for the proposed model in scenario 2. The specific time cost and curvature cost of the SHH algorithm were more advantageous than the results obtained by the PSO and FFA. In addition, the map cost value in the cost function was 0, which indicated the visible paths had no collision with the environment obstacles.

Table 3. Results comparison using PSO, FFA and SHH algorithms in scenario 2.

	PSO	FFA	SHH
Computing time	2.4215s	1.2700s	0.6217s
Final total cost value	588.3817	322.0653	306.4266
Time cost value	185.6156	197.4390	196.8662
Map cost value	0	0	0
Curvature cost value	402.7660	124.6264	109.6048

5.1.3. Scenario 3: Ocean Current and Dangerous Area Information

This part compares and analyzes the operational effects of three algorithms in scenario 3 that contains both ocean current information and danger area information. Figure 10 compares the results of PSO, FFA, and SHH algorithms in such a scenario. The solid line was the result of the PSO, the dash-dot line was the result of the FFA, and the dashed line was the result of the SHH. Apparently, the paths planned by the three algorithms satisfy the constraints of no collision, smoothness, and to some extent along the direction of the ocean current.

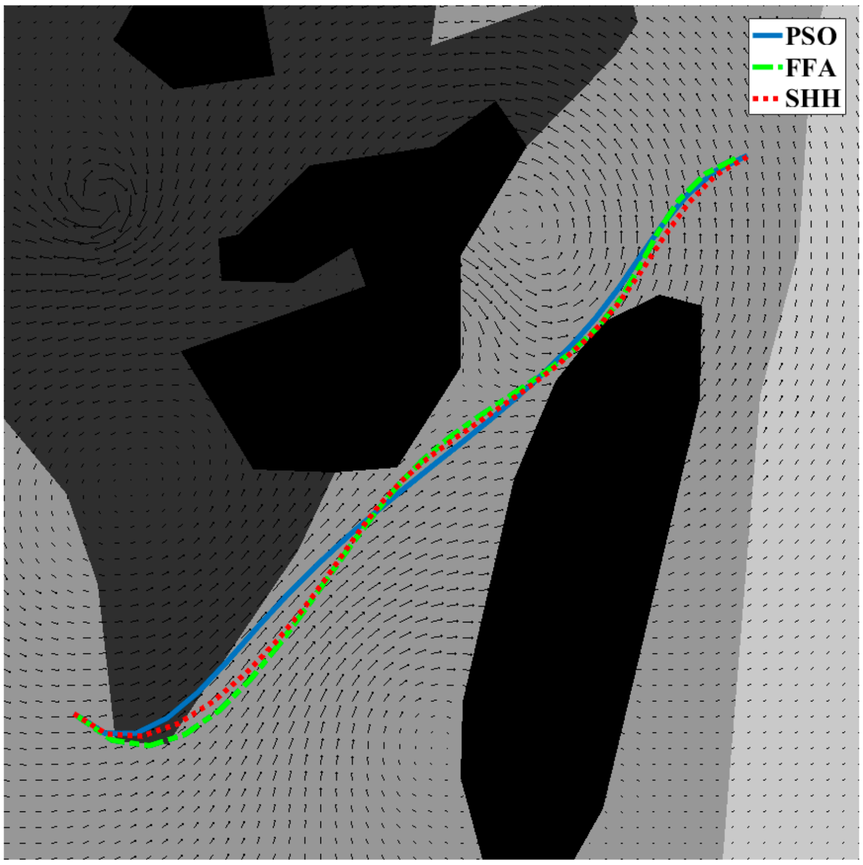

Figure 10. Comparison of the path planning results of PSO, FFA and SHH algorithms in scenario 3.

Table 4 is the result comparison of PSO, FFA and SHH algorithms in the case of scenario 3. In terms of computing time, the SHH algorithm used the shortest computing time, with the average time of 0.6241 s, which was 0.1992 s faster than the FFA. The total cost value of the final path using the FFA was optimized to 935.7844, while for the SHH, the optimal path cost was 893.2590. The SHH had significant advantages in terms of time cost and ocean current cost, but the FFA had a smaller curvature value. PSO had the worst performance in the aspect of computing time, final total cost, and curvature cost value. It shows that the path obtained by the SHH algorithm is shorter and more in line with the ocean current constraint.

Table 4. Results comparison using PSO, FFA and SHH algorithms in Scenario 3.

	PSO	FFA	SHH
Computing time	2.4010s	0.8233s	0.6241s
Final total cost	1072.8236	935.7844	893.2590
Time cost value	185.5981	188.2685	187.1679
Map cost value	0	0	0
Curvature cost value	408.0219	232.7617	240.2193
Ocean current cost value	479.2037	514.7541	465.8717

5.2. Stability Analysis

In addition to the analysis of the results and average data of the three scenarios, the stability and distribution of the results are also statistically analyzed. This subsection compares and analyzes the running time and the resulting stability of three algorithms.

Figure 11 is a comparison of the running time distribution of the PSO, FFA and SHH algorithm in term of running 200 times in scenario 3. The left graph is the statistical result of the PSO. It can be seen that the running time of PSO was concentrated between 2.48 s and 2.35 s, and the median value was approximately 2.39 s. There were many abnormal points on the upper side of the concentrated distribution range, and the largest abnormal point reached approximately 2.62 s. The middle graph is the statistical result of the FFA. Its running time was concentrated between 0.58 s and 1.2 s, and the median was approximately 0.75 s. There were more abnormal points on the upper side of the concentrated distribution range, and the largest abnormal point reached approximately 1.80 s. The right graph is the statistical result of the SHH. It can be seen that the running time of the SHH was concentrated between 0.61 s and 0.65 s, and the median value was approximately 0.62 s. A small number of abnormalities appeared on both sides of the concentrated distribution range. More specifically, the largest abnormal point was approximately 0.69 s, and the smallest abnormal point was approximately 0.60 s. Through comparing the running time distribution from Figure 11, the SHH algorithm that has the most stable and most concentrated distribution of running time among the three algorithms can be obtained. The results indicated the SHH algorithm had a good calculation time stability performance in solving the proposed path planning model in this paper. The time efficiency of the online learning SHH algorithm improved at least 20% compared with those of PSO and FFA.

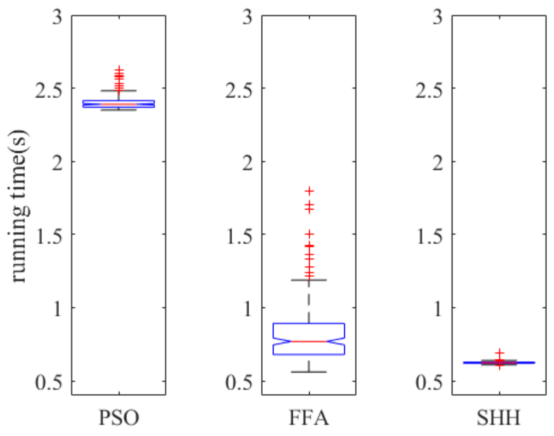

Figure 11. Comparison of the stability of running time of PSO, FFA and SHH algorithms.

Figure 12 is a statistical comparison of the distribution of the final cost values of PSO, FFA, and SHH algorithms after running 200 times in the case of scenario 3. The left graph is the running result of the PSO. It can be seen that the optimal cost was mainly concentrated in the range of 1027 to 1101, and the median value was approximately 1030. There were some abnormal points on the upper side of the concentration range, and the maximum value of these abnormal points was approximately

1195. The middle graph was the running result of the FFA. It can be seen that the optimal cost after optimization was mainly concentrated between 910 and 965, the median was around 935. A small number of abnormal points appeared on the upper side of the concentration range. The maximum was approximately 995. The right graph is the running result of the SHH. It can be seen that the optimal cost value after optimization was mainly concentrated between 860 and 930, the median was around 890. A small amount appears on both the upper side and the lower side of the concentration range. Among the abnormal points, the smallest abnormal point was near 840, and the largest abnormal point was near 980. By comparing the three cost stability analysis graphs, it can be obtained that the cost value distribution of the SHH and FFA have the consistent ranges, that is, the final cost stability of the two algorithms is equivalent. However, the overall cost value of FFA exceeded that of SHH to approximately 50, indicating the SHH algorithm had the best search capability in solving the proposed path planning model in this paper.

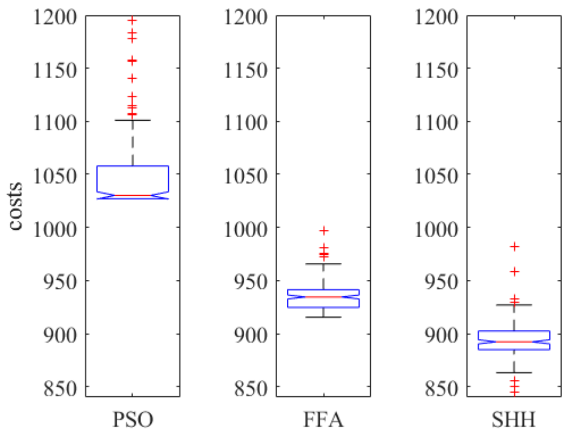

Figure 12. Comparison of the stability of the ultimate total cost value of PSO, FFA and SHH algorithms.

6. Conclusions

In this paper, a unique model was developed which combined the global path planning with the local path planning in order to meet the requirements for the large-scale complex environment. The global path planning only performs the homogenization operation on the task sequence route, so that the subsequent local path planning solution scale is in a relatively stable range. The local path planning was modeled as a nonlinear optimization with multi-constraints, including the time, the danger area, curvature, and ocean currents constraints. The online learning SHH was proposed to attack the complexity of the path planning model. This paper analyzed the results in three scenarios. Moreover, our rigorous comparison of the three algorithms, including PSO, FFA, and SHH algorithms, showed that the online learning SHH algorithm had significant advantages in solving the proposed path planning model in terms of computing time, optimization ability and stability.

7. Patents

The Chinese invention patent CN201910209731.X titled with the AUV path planning in an ocean current environment based on swarm hyper-heuristic algorithms results from the work reported in this manuscript.

Author Contributions: Conceptualization, D.W. and F.W.; data curation, F.W. and H.M.; funding acquisition, D.W.; investigation, H.M.; methodology, D.W. and F.W.; project administration, D.W.; software, F.W. and H.M.; validation, H.M.; writing—original draft, D.W.; writing—review & editing, D.W. and F.W.

Funding: This research was funded by the Sichuan Science and Technology Program (Grant No. 2019YFG0343) and the National Natural Science Foundation of China (Grant No. 51705066).

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

References

1. Zhang, F.; Marani, G.; Smith, R.N.; Choi, H.T. Future trends in marine robotics [tc spotlight]. *IEEE Robot. Autom. Mag.* **2015**, *22*, 14–122. [CrossRef]
2. Cruz, N.A.; Matos, A.C. The MARES AUV, a modular autonomous robot for environment sampling. In Proceedings of the OCEANS 2008, Quebec City, QC, Canada, 15–18 September 2008.
3. Mallios, A.; Ridao, P.; Ribas, D.; Carreras, M.; Camilli, R. Toward autonomous exploration in confined underwater environments. *J. Field Robot.* **2016**, *33*, 994–1012. [CrossRef]
4. Geoffroy, M.; Cottier, F.R.; Berge, J.; Inall, M.E. AUV-based acoustic observations of the distribution and patchiness of pelagic scattering layers during midnight sun. *ICES J. Mar. Sci.* **2016**, *74*, 2342–2353. [CrossRef]
5. Ohta, Y.; Yoshida, H.; Ishibashi, S.; Sugesawa, M.; Fan, F.H.; Tanaka, K. Seabed resource exploration performed by AUV "Yumeiruka". In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016.
6. Kojima, M.; Asada, A.; Mizuno, K.; Nagahashi, K.; Katase, F.; Saito, Y.; Ura, T. AUV IRSAS for submarine hydrothermal deposits exploration. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016.
7. Chen, Q.; Zhang, L.-g. Analysis of current situational development trend of US military UUV. *Ship Sci. Technol.* **2010**, *7*, 129–134.
8. Yuh, J.; Marani, G.; Blidberg, D.R. Applications of marine robotic vehicles. *Intell. Serv. Robot.* **2011**, *4*, 221. [CrossRef]
9. MahmoudZadeh, S.; Powers, D.M.W.; Sammut, K.; Yazdani, A. Toward efficient task assignment and motion planning for large-scale underwater missions. *Int. J. Adv. Robot. Syst.* **2016**, *13*. [CrossRef]
10. Zeng, Z.; Lian, L.; Sammut, K.; He, F.; Tang, Y.; Lammas, A. A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Eng.* **2015**, *110*, 303–313. [CrossRef]
11. Kanayama, Y.; Yuta, S. Vehicle path specification by a sequence of straight lines. *IEEE J. Robot. Autom.* **1988**, *4*, 265–276. [CrossRef]
12. Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
13. Bakolas, E.; Tsiotras, P. Optimal partitioning for spatiotemporal coverage in a drift field. *Automatica* **2013**, *49*, 2064–2073. [CrossRef]
14. Tsourdos, A.; White, B.; Shanmugavel, M. *Cooperative Path Planning of Unmanned Aerial Vehicles*; John Wiley & Sons: Hoboken, NJ, USA, 2010; Volume 32.
15. Wagner, P.; Kordas, J.; Michna, V.; Kotzian, J. Motion control for robots based on cubic hermite splines in real-time. *IFAC Proc. Vol.* **2010**, *43*, 150–155. [CrossRef]
16. Jolly, K.; Kumar, R.S.; Vijayakumar, R. A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits. *Robot. Auton. Syst.* **2009**, *57*, 23–33. [CrossRef]
17. Zadeh, S.M. Autonomous reactive mission scheduling and task-path planning architecture for autonomous underwater vehicle. *arXiv* **2017**, arXiv:1706.04189.
18. Li, D.; Wang, P.; Du, L. Path Planning Technologies for Autonomous Underwater Vehicles-A Review. *IEEE Access* **2019**, *7*, 9745–9768. [CrossRef]
19. Arinaga, S.; Nakajima, S.; Okabe, H.; Ono, A. A motion planning method for an AUV. In Proceedings of the Symposium on Autonomous Underwater Vehicle Technology, Monterey, CA, USA, 2–6 June 1996.
20. Carroll, K.P.; McClaran, S.R.; Nelson, E.L.; Barnett, D.M.; Friesen, D.K.; William, G.N. AUV path planning: An A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, USA, 2–3 June 1992.
21. Carsten, J.; Ferguson, D.; Stentz, A. 3D field D: Improved path planning and replanning in three dimensions. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006.
22. Cui, R.; Li, Y.; Yan, W. Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 993–1004. [CrossRef]
23. McMahon, J.; Plaku, E. Mission and motion planning for autonomous underwater vehicles operating in spatially and temporally complex environments. *IEEE J. Oceanic Eng.* **2016**, *41*, 893–912. [CrossRef]

24. Carreras, M.; Hernández, J.D.; Vidal, E.; Palomeras, N.; Ridao, P. Online motion planning for underwater inspection. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016.
25. Kruger, D.; Stolkin, R.; Blum, A.; Briganti, J. Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007.
26. Putra, Y.; Dae Gil, P.; Wan Kyun, C. Emergency path planning method for unmanned underwater robot. In Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Goyang, Korea, 28–30 October 2015.
27. Roberge, V.; Tarbouchi, M.; Labonté, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans. Ind. Inform.* **2012**, *9*, 132–141. [CrossRef]
28. Zhang, Y.; Gong, D.-W.; Zhang, J.-H. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing* **2013**, *103*, 172–185. [CrossRef]
29. Mirjalili, S.; Dong, J.S.; Lewis, A. *Ant Colony Optimizer: Theory, Literature Review, and Application in AUV Path Planning, in Nature-Inspired Optimizers*; Springer: Cham, Switzerland, 2020; pp. 7–21.
30. Zhu, D.; Cao, X.; Sun, B.; Luo, C. Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *10*, 304–313. [CrossRef]
31. Zeng, Z.; Sammut, K.; Lian, L.; He, F.; Lammas, A.; Tang, Y. A comparison of optimization techniques for AUV path planning in environments with ocean currents. *Robot. Auton. Syst.* **2016**, *82*, 61–72. [CrossRef]
32. Jian, S.; Xing, L. Path Plan of Unmanned Underwater Vehicle Using Particle Swarm Optimization. In Proceedings of the 2015 International Conference on Intelligent Systems Research and Mechatronics Engineering, Zhengzhou, China, 11–13 April 2015; Atlantis Press: Paris, France, 2015.
33. Wang, P.; Meng, P.; Ning, T. Path planning based on hybrid adaptive ant colony algorithm for AUV. In Proceedings of the 2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science, Guilin, China, 19–22 October 2012.
34. Alvarez, A.; Caiti, A.; Onken, R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE J. Ocean. Eng.* **2004**, *29*, 418–429. [CrossRef]
35. Garau, B.; Alvarez, A.; Oliver, G. AUV navigation through turbulent ocean environments supported by onboard H-ADCP. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006 (ICRA 2006), Orlando, FL, USA, 15–19 May 2006.
36. Burke, E.K.; Hyde, M.R.; Kendall, G.; Ochoa, G.; Özcan, E.; Woodward, J.R. A Classification of Hyper-Heuristic Approaches: Revisited. In *Handbook of Metaheuristics*; Gendreau, M., Potvin, J.-Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 453–477.
37. Tilahun, S.L.; Tawhid, M.A. Swarm hyperheuristic framework. *J. Heuristics* **2018**, 1–28. [CrossRef]