A Multi-AUV Path Planning System Based on the Omni-Directional Sensing Ability

Jingyu Ru 1, Shuangjiang Yu 1, Hao Wu 2, Yuhan Li 3, Chengdong Wu 1,*, Zixi Jia 1,* and Hongli Xu 1,*

1 Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China; rujingyu@mail.neu.edu.cn (J.R.); yushuangjiang@stmail.neu.edu.cn (S.Y.)
2 Engineering Faculty, University of Sydney, Sydney, NSW 2006, Australia; hawu1598@uni.sydney.edu.au
3 Graduate School of Economics, Kyoto University, Kyoto 606-8501, Japan; ri.ukan.62r@st.kyoto-u.ac.jp
* Correspondence: wuchengdong@mail.neu.edu.cn (C.W.); jiazixi@mail.neu.edu.cn (Z.J.); xuhongli@mail.neu.edu.cn (H.X.)

Abstract: Following the development of autonomous underwater vehicles (AUVs), multiple trajectory-based submarine target information collection constitutes one of the key technologies that significantly influence underwater information collection ability and deployment efficiency. In this paper, we propose an underwater information collection AUV, O-AUV, that can perceive the omnidirectional area and could detect a larger area than the traditional AUV. A 3D sensing model for the O-AUV is proposed to describe the complex underwater information collection spaces. Thereafter, a cube-based environment model involving candidate observation point calculation methods are suggested to adapt the O-AUV model. A voyage cost map is also built according to the multi-AUV path planning for a common submarine mission that must traverse numerous mission targets in complex environments through the R-Dijkstra algorithm. Specifically, the voyage planning problem is solved through a critical algorithm called ANSGA (accelerated NSGA-II algorithm), which in turn, is developed by modifying the non-dominated sorting genetic algorithm (NSGA-II) to accelerate the optimization rate for the Pareto solution. Experiments are carried out in MATLAB, and the results verify the validity of the proposed O-AUV+ANSGA algorithm framework.

Keywords: underwater sensor; omnidirectional sensing model; autonomous underwater vehicle; voyage planning; multi-AUVs; non-dominated sorting genetic algorithm

1. Introduction

The research of ocean exploration is significant for human survival and development. The ocean contains abundant resources, including fossil resources and approximately 38 billion tons of proven oil reserves which account for 34% of the world’s total oil resources. Moreover, the ocean is also largely unknown, with 95% of the ocean being unexplored and containing unidentified creatures and monuments, as well as wrecks.

Given the increasing exploration of the ocean, academics have conducted extensive research in this field. Autonomous underwater vehicles (AUVs) have become popular for conducting various submarine missions [1,2] because of the environmental difficulties. For one thing, traditional AUVs are mostly torpedo-shaped, with the visible direction forward, and their perceptual model is mostly a single sector [3,4], thereby virtually limiting the detection efficiency of AUVs. Therefore, AUVs with multi-sensory orientation should be seriously investigated, although this matter has always been neglected in prior research. For another, some academics have used several underwater autonomous vehicles to perform joint tasks in complex and dangerous underwater environments [5,6]. However, they did not consider the factors of the AUVs’ model, the shielding in real environments, and the deployment efficiency of the AUVs.

In analyzing the information collection of multiple AUVs in complex submarine environments, the quality and efficiency of performing underwater exploration tasks will be
improved if the assumption is made that AUVs can perceive multiple directions. In particular, the AUV can capture more information and enhance the accuracy of information collection when its perception region is omnidirectional. Moreover, the omnidirectional sensing capability of AUVs facilitates reduction of the steering motion in the water while making the AUV trajectory smoother, thereby minimizing the energy consumption of the task.

Studying the seabed information collection problem for multi-sensing direction AUVs in complex seabed environments is challenging because of the following reasons. First, in attaining a multi-sensing direction, the perceived scope of the AUVs is no longer a point but a three-dimensional perceptual area and we must consider factors such as multi-perception area and masking [7] in the complex underwater environment during modeling. Secondly, problems of trajectory optimization accuracy and optimization efficiency will be raised when the multi-AUV system performs underwater information collection tasks according to the new multi-perception model.

To address these issues, this work proposes and designs a novel AUV with omnidirectional sensing capability and which carries multiple sensing cameras to simultaneously explore around the robot in the complex underwater environment. The omnidirectional mentioned in this paper is not complete coverage in the strict mathematical sense, the visual area still has some blind spots. However, compared with conventional sensors [8], it does have “omnidirectional” coverage. By focusing on aspects, we (1) establish a three-dimensional omnidirectional underwater information collection model by considering complex underwater environments, and (2) design a set of multi-AUV trajectory optimization strategies that consider overall energy consumption on the basis of the motion characteristics of the robot. This work makes the following contributions to the literature:

1. We propose an underwater information collection AUV, the O-AUV, that can perceive an omnidirectional area. Then, we develop an underwater three-dimensional omnidirectional perception model according to the characteristics of the O-AUV.
2. We present a trajectory optimization algorithm for multiple O-AUVs, and this algorithm is adapted from targets traveling missions in large and complex seabed environments.
3. The validity of our data is verified according to virtual submarine data.

The rest of our paper is organized as follows. Section 2 presents the motivation for this work. Section 3 discusses the detecting model of the O-AUV and the multiple O-AUV path planning with our model. Section 4 evaluates our design in MATLAB with a virtually generated seabed dataset and experiments. Section 5 summarizes related works. Finally, Section 6 concludes this paper.

2. Motivation

In this section, we illustrate the benefits of AUV path planning for target traveling missions in complex seabed environments. Then, we explain the need to use multiple omnidirectional sensing AUV to accomplish these tasks, followed by a discussion of the challenges of establishing the system of the O-AUV exploration optimization algorithm.

2.1. Benefits of Underwater Exploration for Multiple AUVs

According to the AUV related investigation of the seabed, researchers will be offered a significant opportunity for more extensive exploration of underwater conditions.

People have different requirements for AUV underwater exploration. On the one hand, exploration with a single AUV already has a broad range of applications, such as dam crack detection [9], pipeline fault diagnosis [10], and underwater ecological exploration [11,12]. On the other hand, the expectation of underwater exploration capability becomes higher with the expansion of the boundaries of human activity. More and more jobs require multiple AUVs to work together, such as for the mapping of broad underwater landscapes, underwater disaster search [13] and rescue [14], and node deployment [15]. For these tasks, is taking a cruise of AUVs for multiple target points is necessary, and they often entail a certain mission duration. Given factors including UAV energy consumption limitations
and numerous, scattered target points, a single AUV cannot complete the above tasks quickly and accurately.

To solve these problems, building a multi-AUV system composed of AUVs and a movable surface vehicle (MSV) has gained popularity because of the development of automatic AUV launch and recovery (L&R) from movable platforms, e.g., an unmanned surface vehicle (USV) [16], to conduct missions in an ocean space distant from the coast, as shown in Figure 1.

![Figure 1](image1.png)

Figure 1. 2D exemplification of a multi-AUV target information collection mission by L&R from an MSV.

Through the MSV and multi-AUV exploration system, the information collection results of underwater complex environment are ascertained so as to assist researchers.

2.2. The Need for O-AUV Path Planning

Despite the many benefits of multi-AUV exploration, some difficulties persist in the efficient implementation of the system.

First, most path planning research for the problem of underwater cruising at multiple target points currently neglect the information collection area and range of the AUV, a situation which means that the AUV has to move to each target point, as shown in Figure 2a. Assuming that an O-AUV is employed, i.e., one with an omnidirectional information collection capability, then the cruise trajectory will change. The O-AUV neither requires precise manipulation of the target, nor does its position coincide with the target exactly. As long as the target appears in its information collection range, the O-AUV can perform certain specific tasks in shallow water areas with good water conditions. For example, in Figure 2b, the black line represents the trajectory of the O-AUV, and that path is flatter and shorter than that of the normal AUV in Figure 2a. However, the O-AUV also uses fewer AUVs.

![Figure 2](image2.png)

Figure 2. Multiple trajectory planning problem. (a) Multiple trajectory planning problem for AUVs. (b) Multiple trajectory planning problem for O-AUVs.

Second, most information collection models in many studies ignore the complex surface of the seabed. These always assume that the seabed is horizontal and model the sensing area as a sector or trapezoid on this basis [17]. When the underwater monitoring area is a plane, the projection is a trapezoid $D_1D_2D_3D_4$, as shown in Figure 3a [18].
Figure 3. Motivation of information collection model. (a) Schematics of the monitoring area. (b) 3D sensing model of the O-AUV.

However, the actual subsea environment is tumultuous, and AUVs work in an obstructed 3D space, as shown in Figure 3a. In this situation, obstructions, such as hills will block AUV’s information collection area, as shown in Figure 3b. Fast and accurate calculation of the masking area is necessary to improve the information collection efficiency of O-AUV.

Third, the problem of execution efficiency is encountered during the cruise of multiple targets with the O-AUV. In addition, a single O-AUV can probe a wider area than conventional AUVs, but the effectiveness of the cruise probe cannot be guaranteed if the target points are spread over a wide area because of the energy limitation of AUVs. To solve the aforementioned problems, multiple O-AUVs must be employed for multi-target cruise information collection to ensure mission completion.

We can now see that in underwater information collection experiments, the complex environment will influence the effectiveness of multiple target information collection. Thus, we need a multiple O-AUV path planning algorithm on the basis of a complex underwater environment to optimize the trajectories of the O-AUVs.

2.3. Challenges

Many challenges are associated with the implementation of the system mentioned above. Below are some of these problems that may lead to an inefficient system.

The 3D omnidirectional sensing model for multiple O-AUVs is complex. On the basis of a special AUV with omnidirectional perception ability, we must analyze the sensing model from an omnidirectional perspective. Building a 3D sensing model for the O-AUV is difficult. Since there are three sensor directions, we have to make reasonable calculations to obtain accurate results.

Any obstruction in the seabed environment will have an impact on the information collection results. The sensor orientation, obstacle location and raw surface data have a significant impact on the information collection range when modeling a O-AUV system, for example, different relative positions between sensors and obstacles generate different information collection results.

Optimizing the trajectories of multiple O-AUVs is arduous. As different target points must be detected by only one O-AUV, the sequence of information collection tasks is vital and is directly related to the efficiency of the system operation. Different information collection orders and the trajectories of arbitrary target affect others, thereby influencing the cruise outcome of the O-AUVs. Therefore, analysis of numerous trajectories containing a large number of points is difficult.

3. Model Design

On the basis of the motivation above, we propose an efficient and accurate multiple O-AUV-based trajectory planning algorithm framework.
The overall architecture of the approach is illustrated in Figure 4. Specifically, we first design the O-AUV sensing model to estimate the sensing space. Then, the model of the candidate observation points (COPs) is built to constrain the cube-based COPs. The R-Dijkstra algorithm is executed according to the cubed-based search domain to build a voyage cost map (CM) among COPs to hasten voyage evaluation during voyage trajectory planning. Subsequently, a multiple object optimal algorithm based on the accelerated NSGA-II algorithm (ANSGA) framework is executed to plan a set of O-AUV voyages to visit all the targets. Finally, the R-Dijkstra search is adopted to draw detailed trajectories according to the O-AUV voyages obtained by the O-AUV+ANSGA framework.

Figure 4. Workflow and dataflow of the proposed approach.

In this section, we will present the details of our algorithm. The algorithm, which is a hybrid approach, consists of four parts.

We first choose the O-AUV sensing model in Section 3.1 to model the coverage of the O-AUV. Then, the cube-based environment modeling is introduced in Section 3.2 to build an environment model that fits the information collection mission in Section 3.3. In the next step, we propose an R-Dijkstra algorithm to build the voyage CM in Section 3.4. Finally, the ANSGA framework for the multiple O-AUV trajectory planning will be introduced in Section 3.5.

3.1. O-AUV Sensing Model

The O-AUV carries multiple sensing cameras to simultaneously explore in the complex underwater environments, with each sensing camera evenly distributed and angled at 120° to each other, and the top view of the internal structure of O-AUV is shown in Figure 5. Information is collected based on pictures.

Figure 5. Top view of the internal structure of O-AUV.

The sensor monitoring model of the O-AUV for each camera is denoted by four-tuple $(P, \vec{R}, \alpha, \xi)$, as Figure 6 shows. $P$ is the spatial coordinate of the sensor, and our assumption of sensor information collection range is a quadrilateral cone $P - D_1D_2D_3D_4$, which has a
limit information collection surface of $D_1D_2D_3D_4$ with $F$ as the center point. Additionally, $\vec{R} = (\gamma, \theta)$ denotes the main sensing direction $\overrightarrow{PF}$, for which $\gamma$ is the angle between $\overrightarrow{PF}$ and the negative direction of the $Z$-axis. The sensor is fixed on the O-AUV, and the two directions are the same, so $\overrightarrow{PF}$ is also the moving direction of the object.

![Figure 6. 3D sensing model for each camera.](image)

The length of $PF$ refers to the information collection range of the sensor, where $\theta$ is the angle of the anticlockwise rotation from the positive direction of the $X$-axis to $OF$. $OF$ is the shortest distance from $F$ to the $Z$-axis. $F_1$ is the middle point of $D_1D_2$, and $F_2$ is the middle point of $D_1D_4$. Accordingly, $\xi = \angle F_1PF$ and $\alpha = \angle F_2PF$. Thus, the sensor can monitor the area in the distance range of $[\theta - \alpha, \theta + \alpha]$ and $[\gamma - \xi, \gamma + \xi]$.

To obtain better coverage results, we propose an O-AUV that consists of three information collection sensors evenly and can sense the surrounding underwater environment. The “O” stands for omnidirectional and is also a visualization of the omnidirectional perception range.

The perception model, shown in Figure 7, consists of three independent perception models. The vector $\theta$ of three perception models satisfy $\theta' = \theta + \omega$, $\theta'' = \theta - \omega$, where $\omega = 120^\circ$ meet the requirement of the omnidirectional sensing.

![Figure 7. The perception model.](image)

During information collection, we assume that the image resolution is constant at the time of capture, while the sensor can capture the corresponding object contours. Since proper tilting of the camera can improve the accuracy of recognition, we make $\gamma$ greater than zero. Please note that the O-AUV proposed in this paper is isomorphic, i.e., all the O-AUVs have the same information collection depression angle $\gamma$.

### 3.2. Cube-Based Environment Modeling

The environment is transformed into a search graph for more convenient trajectory planning. Accordingly, given a rectangular 3D environment with a square base with bottom side length $L$, and height $H$, we decompose the figure into exclusive cuboids of
the same size, say, with a side length $C_L$ and height $C_H$. Therefore, there is only one cube surrounding each point within the environment.

Each cube can be denoted by a unique triple integer index $c_i = (x, y, z)$, and the cube center point $p = (i, j, k)$ satisfies $x = i \cdot C_L$, $y = j \cdot C_L$, $z = k \cdot C_H$ where $i \in [0, L/C_L]$, $j \in [0, L/C_L]$, $k \in [0, L/C_H]$. In addition, we categorize the cubes and give different cube values, which is more beneficial to reduce the travel cost calculation of the trajectory. In particular, if the cube intersects an obstacle or bathymetry, we define it as an obstacle cube, and the cube will be assigned the feasibility factor $s(i, j, k) = \infty$ to indicate that it is infeasible, displayed as blue in Figure 8.

Therefore, if the path intersects the feasible cube, then this path is feasible and $s(i, j, k) = 1$, displayed as white in Figure 8. To construct the search graph $G = (V, E)$, the vertex set is composed of all feasible cubes, and we define the vertex of $G$ in $s_i j k$ as the $s$ cube, and the cube will be assigned the feasibility factor $s(i, j, k) = \infty$ to indicate that it is infeasible, displayed as blue in Figure 8.

![Figure 8. Cube-based environment modeling.](image)

The edge set $E$ is generated between all vertexes $v_p$ of cubes in $G$. Each edge must be feasible because it only intersects the feasible cube. Therefore, we define that for any two cubes vertexes $v_p$ and $v_p'$, the distance between them is as follows:

$$e(v_p, v_p') = \text{dist}(v_p, v_p') \cdot s \cdot s'$$  

(1)

where the $\text{dist}$ function represents the calculation of the Euclidean distance between $v_p$ and $v_p'$, and $s$ and $s'$ are their feasibility factors, respectively. Thus, if any one of the two points is blocked, then the distance between the two points will be infinite, i.e., each vertex has at most 26 edges connected to it.

Next, we define $e(v_p, v_{pa})$ as the value of edge which is the distance from $v_p$ to $v_{pa}$, where the $v_{pa}$ is the adjacent vertex to the $v_p$. Their coordinates are $c_p = (i_p, j_p, k_p)$ and $c_{pa} = (i_{pa}, j_{pa}, k_{pa})$, and they satisfy the following formula:

$$\max(|i_p - i_{pa}|, |j_p - j_{pa}|, |k_p - k_{pa}|) \leq 1$$  

(2)

The distance of $v_p$ and $v_{pa}$ is defined as $e(v_p, v_{pa})$, and we define the number of adjacent vertices to the $v_p$ as $na(v_p)$.

### 3.3. Calculation of Candidate Observation Points

We use the above O-AUV sensing model to calculate the COPs of the targets that must be detected. On the one hand, any target location is limited in discrete space because of the limitations of the O-AUV sensing model. On the other hand, the information collection depression angle $\gamma$ of the O-AUV sensor is a fixed value during the experiment and will not change during movement. Furthermore, the working angle of the O-AUV is assumed to always be horizontal, so the main factor determining the information collection range is the vertical section of the model in Figure 9, with a detailed example as shown in Figure 9. (The
dark blue area is the information collection range of the O-AUV, and light blue counterpart is the area where the information collection range is blocked.)

Figure 9. Calculation of COPs. (a) XOZ view of the detected area model; (b) XOZ view of the detected area model ($\gamma = 90^\circ$).

Figure 9a is the XOZ view of the detected area model, where $F_1$ is the target point. As the potential horizontal information collection range is $PF_1F_2$ (blue box in Figure 9), the possible position of O-AUV is shown in area $PP'F_1$ of Figure 9, and its mathematical derivation is as follows:

As the horizontal direction of the O-AUV movement $\theta$ can be arbitrary, the movement position is unlimited and the position of the target point $F_1$ that can be observed in space is the three-dimensional space obtained by horizontally rotating the profile $PP'F_1$ around the point $F_1$ ($\gamma = 0^\circ$), i.e., as long as the O-AUV is located in this space, the target point $F_1$ can be sensed by the O-AUV.

We identify each feasible cube, where $s(i,j,k) = 1$ in the cube-based environment modeling space are the COPs.

3.4. Voyage CM Building
3.4.1. The Dijkstra Algorithm

Dijkstra’s algorithm, proposed by Dutch computer scientist Edsger Wybe Dijkstra in 1959, solves the single-source shortest path problem of an empowered graph using a similar approach to breadth-first search. According to the algorithm, let $G = (V, E)$ be a weighted directed graph and divide the set of vertices $V$ into two sets, $S$ and $Q$. The set $S$ stores all the vertices with known actual shortest path values, and the set $Q$ stores the remaining vertices. Then the vertices in $Q$ are added to $S$ according to the ascending order of the shortest path length until all the vertices in $Q$ are put into $S$. During the addition process, the shortest paths from the source SP to the vertices in $S$ are always kept no longer than the shortest paths from the source SP to the vertices in $Q$.

3.4.2. Improvement of Dijkstra Algorithm

For rapid evaluation of voyages, we need a voyage CM to indicate the best costs with sufficient number of mission target COPs.

We first build a Dijkstra framework-based algorithm named R-Dijkstra and adopt it for CM building as shown in Algorithm 1. In this algorithm, the graph is searched once from each target $T$ so as to ascertain its traveling cost among COPs. We use $T(n)$ to represent the n-th target, and its m-th COP in COPs set is represented as $p(n,m)$. The number of COPs for each $T(n)$ is $M_n$.

We assume that $N$ target points exist, and the number of their COPs is defined as $List = \{M_1, M_2, \ldots, M_n \ldots M_N\}$. We define a CM matrix $C$ of size $\sum_{n=1}^{N} M_n \times \sum_{n=1}^{N} M_n$ to store the distance between COPs of different target points for any one of the elements.
in the i-th row and j-th column $i \leq \sum_{n=1}^{N} M_n$, $j \leq \sum_{n=1}^{N} M_n$. The m and n values can be decoded through the table. The specific steps are as follows:

- The corresponding $n_i$ can be decoded by the calculation $M_n < i < M_{n+1}$.
- Then, we identify the corresponding $m_i$ value through $i - M_n$.
- The calculation method of the $m_j$ and $n_j$ of $j$ is consistent with the above method. Thus, the shortest path from the $N_i$ COP of the target point $m_i$ to the $n_j$ COP of the target point $m_j$ is stored in the element $(i, j)$ in CM. The formula is as follows:

$$CM(i, j) = Dist(p(n_i, m_i), p(n_j, m_j))$$ (3)

where $Dist$ represents the distance obtained through the R-Dijkstra algorithm, and $n_i \neq n_i$, the numerical relationship between $m_j$ and $m_{i, j}$ is not required.

Algorithm 1 R-Dijkstra Algorithm

| Algorithm 1 R-Dijkstra Algorithm |
|----------------------------------|
| **Initialization**               |
| $sp = P(n_i, m_i)$                |
| **for each vertex $v_p$ in $V_p$**| |
| $d[v_p] = \text{infinity}$;       |
| $\text{previous}[v_p] = \text{undefined}$; |
| $\text{Mark}[v_p] = 0;$           |
| **end**                          |
| $P' = \text{COPs set -} P(n_i, \forall(m))$; |
| **for each** $P(n_j, m_j)$ in $P'$ do |
| $ep = P(n_j, m_j);$               |
| **if** $\text{Mark}[ep] = 1$ then |
| \hspace{1cm} **return** $d[ep]$;     |
| **end**                          |
| **end**                          |
| **function** R-Dijkstra($G, E, sp, ep$) |
| $d[sp] = 0$                      |
| $S = \text{empty set}$          |
| $Q = \text{set of all vertices}$ |
| **while** $Q$ is not an empty set do |
| $u = \text{Extract\_Min}(Q)$     |
| $\text{mark}[u]=1;$             |
| S.append($u$)                    |
| **for each** edge outgoing from $u$ as $(u, v_p)$ do |
| \hspace{1cm} **if** $d[v_p] > d[u] + E(u, v_p)$ then |
| \hspace{2cm} $d[v_p] = d[u] + E(u, v_p)$ |
| \hspace{2cm} $\text{previous}[v_p] = u$ |
| **end**                          |
| **end**                          |
| **if** $u$ is $ep$ then          |
| \hspace{1cm} **break and return** $d[ep]$; |
| **end**                          |
| **end**                          |
| $CM(i, j) = d[ep]$;              |

Calculating CM is very time-consuming, and we propose a shortest distance by reusing an algorithm to improve the efficiency of CM construction. First, the R-Dijkstra algorithm can find the shortest path between any two points in the map through an undirected graph. Therefore, the distances from $(i, j)$ to $(j, i)$ are completely consistent, a situation which means $CM(i, j) = CM(j, i)$. Moreover, all the diagonal elements in CM are zero, so we need just find the CM $(i, j)$ that satisfies $(i > j)$ or $(i < j)$.

Second, the R-Dijkstra algorithm traverses many vertices in the process of calculating the undirected graph and calculates and records the shortest path and distance from
the starting point to these points until the traversal to the end point. When calculating trajectories with the same starting point, i.e., when calculating a certain line in the CM, many repeated calculations will occur, thereby wasting computing resources. For example, when calculating CM \((i, j)\) and CM \((1, j + 1)\), if \(j\) is very close to the COP represented by \(j + 1\), then more than 90% of the calculated data may be the same. Calculating the final result of CM \((i, j + 1)\) may even be possible by traversing a very small distance on the result of CM \((i, j)\).

In summary, considering the partial reuse of the results from the trajectory calculation at the same starting point and different end point is a very effective approach for improving system efficiency. Accordingly, we propose R-Dijkstra, a distance reuse-based multi-shortest trajectory calculation method based on the Dijkstra algorithm framework. Compared with the traditional Dijkstra algorithm, this algorithm saves the shortest distance \(d\) of each node after each trajectory calculation, marks it in mark, and finally stops when the nearest element meets the condition of \(u = ep\). All cubes adjacent to each cube are checked for expansion and stored in the queue \(Q\). Be aware that because there can be multiple paths to the cube in a graph, data about optimal paths to the cube may be updated several times. Thus, after the path extension at each closest cube \(u\), the cube should be reset as unvisited by the Mark function.

When calculating the trajectory of the same starting point later, the first query was whether the point has been traversed. If that point has been traversed, then we directly select CM\((i, j) = d[ep]\) as the final result, and this approach can be regarded as the reuse of path information.

The well-known Dijkstra’s algorithm works fairly well for path planning from a given source node in the n-th target group to a set targets and can be run in time \((N - M_n) \times O(|V| + |E| \times \log |V|)\). The algorithm proposed in this paper can reduce the computational cost of building the voyage CM, and the algorithm complexity can reach \(O(|V| + |E| \times \log |V|)\).

3.5. ANSGA for Trajectory Planning

A multi-objective optimization method has been used to solve many modern engineering problems. Many of those problems were successfully resolved using evolutionary algorithms. If the importance of the criteria is varies, then a layered solution can be found. On the contrary, if the criteria are equally important, the Pareto optimal solution can be considered. One of the most outstanding evolutionary algorithms is the non-dominated sorting genetic algorithm (NSGA-II). With NSGA-II, we can find an equally good set of solutions called the Pareto front or Pareto optimum solutions. The solutions in the collection do not dominate one another. In fact, in considering all the target functions, none of the solutions outperform the others.

In this work, we overlook the nonlinear relationship between the motion form of the O-AUV and its energy consumption and simplify the matter with a linear model, i.e., its distance length represents the energy consumption of the O-AUV. Moreover, the driving distance length also represents the operating time of the O-AUV because the velocity of the O-AUV is assumed to be uniform. By contrast, we assume that the starting and ending points of each O-AUV information collection path are the same when multiple O-AUVs perform information collection tasks simultaneously, i.e., the position of the MSV receiving the O-AUV is unchanged. The start time of the information collection task is the same, but the end times vary. Consequently, the information collection system mounted on the MSV must wait for the O-AUV with the farthest mission path to complete the task before the information collection task is completed. Therefore, this algorithm must also consider the length of the path with the longest task execution time as one of the optimization factors. Accordingly, we want the path with the longest time to be as short as possible.
According to the analysis above, two criteria with three constraints are considered for evaluation of the optimal path as follows.

\[
\begin{align*}
  p_1 &= \sum_{i=1}^{N_O} O_i \\
  p_2 &= \text{maximum}(O_i) \\
  p_1 &\leq E_T \\
  p_2 &\leq E_O \\
  s(O_i) &= 0
\end{align*}
\]

where \( p_1 \) can also be considered to be the summation of the loads of all targets in voyage. \( E_T \) and \( E_O \) illustrate the constraints on the O-AUV, where \( E_T \) is defined as the necessary energy cost limitation of all the detailed trajectories to complete the voyage and \( E_O \) is defined as the cost limitation for any single O-AUV. \( s(O_i) = 0 \) prevents the O-AUV voyage from having any collisions with the bathymetry and any obstacle.

In addition to the above limitations, the traversal process must also ensure that all \( N \) target points need to be traversed at least once.

We assume that the \( i \)-th O-AUV is required to traverse \( K \) targets, a calculation of the distance \( O_i \) proceeds as follows:

\[
O_i = \sum_{k=1}^{K-1} o^k_i 
\]

where \( o^k_i \) means the O-AUV trajectory length from the \((k-1)\)-th target to the \( k \)-th target.

When the traversal point is the \( m_{k-1} \)-th COP of the \((k-1)\)-th target, and the \( m_k \)-th COP of the \( k \)-th target, then according to Equation (5), \( o^k_i \) is calculated as follows:

\[
o^k_i = CM(p(k-1,m_{k-1}), p(k,m_k))
\]

Here \( p_1 \) and \( p_2 \) are the optimization objective functions, and we expect them to be maximally reduced.

According to the GA algorithm framework, the initial chromosome group that is randomly generated by the multiple 0/1-encoding of the O-AUVs’ trajectories is obtained. Then, the genetic operation is performed, and the offspring function is used to replace the parent function through the Mutation, Crossover, Selection operation based on the \( p_1 \) and \( p_2 \).

The crowding distance is calculated primarily to maintain population diversity while maintaining population size. Each solution has two properties and a dominating set: the non-domination rank \( I_{\text{rand}} \), crowding distance \( I_d \), and the dominating set \( S_p \). The calculation formula for the crowding distance is shown as follows:

\[
I_d(j) = \sum_{i=1}^{2} \frac{p_{i}^{j+1} - p_{i}^{j-1}}{p_{i}^{\text{max}} - p_{i}^{\text{min}}}
\]

where \( p_{i}^{j+1} \) is the \( i \)-th objective function value of individual \( j \). The parameters \( p_{i}^{\text{max}} \) and \( p_{i}^{\text{min}} \) are the maximum and minimum values of the \( i \)-th optimization objective function.

To have a more uniform set of Pareto solutions, we propose a crowding distance-based ANSGA. We construct an accelerated coefficient of variation on the basis of crowding distance to increase the variation speed of the system’s solutions in the high-density region and achieve an equilibrium optimal solution as well as improve the search speed of the system. The formula of ANSGA mutation probability is as follows.

\[
p_a(j) = \min(p_b, e^{I_d(j)}, 1)
\]

where \( p_b \) is the basic mutation probability, and the minimum operation involves limiting the upper bound of the probability.

The principle is to determine the density of the solution around each solution and remove solutions with high-density values. Then, the iteration is repeated until the number
of iterations and the Pareto-optimal front are obtained \( (I_{\text{rand}} = 1) \). When the search ends, a series of Pareto optimal solutions are found, and the officers can refer to these solutions for multiple target information collection.

4. Evaluation

In this section, all experiments are performed with MATLAB 2020A. The experiments were executed on a desktop with an AMD Ryzen 9-3950X central processing unit (CPU) @ 3.50 GHz and with a 64 GB memory.

4.1. O-AUV Sensing Model

In this subsection, we evaluate the construction of the O-AUV sensing model. The basic parameters of the O-AUV are shown in Table 1.

| Parameter Settings | Value     |
|--------------------|-----------|
| PF                 | 10 m      |
| PO                 | 2 m       |
| \( \omega \)       | 120°      |

Table 1. Parameter settings of O-AUV.

Figure 10 depicts the coverage area of the O-AUV sensing model for different parameters, in which the green circles represent the O-AUV, the pink line is the height \( PO \) of the O-AUV, and the yellow squares represent the area that is detected by the sensor.

(a)              (b)

**Figure 10.** O-AUV sensing model. (a) the coverage area of the O-AUV sensing model on the plane area; (b) the coverage area of the O-AUV sensing model on the complex area.

The coverage areas of these two O-AUVs have different characteristics. In Figure 10a, the \( \xi = \alpha = 30^\circ \) and the \( \gamma = 60^\circ \). In this situation, the detecting surface is slightly flat and no obstruction could block the visual field of the O-AUV information collection sensor. As can be observed, the information collection direction of the O-AUV is uniformly separated and the sensors it carries project a information collection area on a relatively flat surface.

In Figure 10b, the \( \xi = \alpha = 60^\circ \) and the \( \gamma = 90^\circ \). As the parameters \( \xi \) and \( \alpha \) in Figure 10b are larger, the probe area in Figure 10b is larger than the area in Figure 10a. In Figure 10b, We can conclude that it covers a much larger area than the other one, due to the fact that the submarine peaks obstruct a large area. Therefore, the O-AUV sensing model coverage can estimate the different complex coverage areas precisely and efficiently.

4.2. Calculation of the COPs

In this subsection, we verify the calculation of COP calculation of the O-AUV using three examples. All the parameter settings of the O-AUV are the same as the ones in Figure 10b,
for which the $\gamma = 90^\circ$. The information collection area is $60 \times 60$ m, is meshed by $30 \times 30$ grids, and has a length of 2 m.

The COPs of these three O-AUVs have different characteristics. The blue circle represents the target, and the red points represent the COPs. From Figure 11, we can conclude that the coverage area of the COPs very much resembles a “ring” with the central part as concave. In addition, the partial zoom of T (1) is shown in Figure 11b. The reason for this occurrence was explained in Section 3.3.

The T(1) and T(3) have much more COPs than the T(2), because T(2) is located in a valley of the sea and the terrain around it interferes with its observable area. The area of T(1) is at the bottom of the sea on the flat horizontal plane, and its shape is approximately a hemisphere which is upside down on the flat surface. The COPs of T(3) are asymmetric. Although the mountain peaks of the seabed block almost half of the COPs, it still can be detected by the COPs below it for its suspended position. This outcome indicates that the COP calculation model can estimate the COPs of each target in different complex areas in precisely and efficiently.

The experimental results of this paper are shown in Figure 12. Our assumption is that the area is $30 \times 30$ m, is meshed by $30 \times 30$ grids, and has a length of 1 m. To more closely match the underwater reality, the acceptable information collection distance is assumed to be $PF = 2.5$ m, $\gamma = 90^\circ$, the total number of target points is 40 and the deployment location is random.

Figure 12 clearly indicates that the COPs of the neighboring target points overlap considerably, an outcome which provides some support for this multi-trajectory planning problem according to O-AUVs.

![Figure 11. COPs and the partial zoom of the targets. (a) COPs of targets; (b) The partial zoom of T(1).](image)

![Figure 12. COPs of targets on the complex area.](image)
4.3. Voyage CM

First, constructing a voyage CM is necessary. Figure 13 is the shortest 3D path between any two points \( i \) and \( j \) using the R-Dijkstra algorithm in the above environment, and the length of the trajectory is an element of \( CM(i,j) \). Figure 13 indicates that the shortest path between two points is a 3D shortest trajectory constrained by the topography of the seafloor in the non-line-of-sight situation. However, given that the O-AUV can hover in water, the trajectory can be partially suspended, i.e., the O-AUV can traverse through two points without always staying close to the topography of the seafloor.

Figure 13. The path of the \( CM(i,j) \) calculated by the R-Dijkstra algorithm.

Table 2 shows the running time of a voyage CM consisting of \( T(1) = \{25, 7, 9\} \) and \( T(2) = \{1, 17, 1\} \) when \( \gamma = 90^\circ \) is used and different \( PF \) are selected.

Table 2. Comparison between R-Dijkstra and Dijkstra.

| Algorithm   | Time \((PF = 0 \text{ m})\) | Time \((PF = 1 \text{ m})\) | Time \((PF = 2 \text{ m})\) | Time \((PF = 3 \text{ m})\) |
|-------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| R-Dijkstra  | 4.1 s                       | 19.7 s                      | 46.0 s                      | 75.2 s                      |
| Dijkstra    | 4.1 s                       | 37.2 s                      | 190.6 s                     | 459.6 s                     |

The results of the voyage CM constructed by the R-Dijkstra and Dijkstra algorithms above are consistent. Table 2 reveals no difference between the R-Dijkstra and Dijkstra algorithms when \( PF = 0 \). Moreover, the proposed R-Dijkstra algorithm in this paper can effectively reduce the computation of the voyage CM as the \( PF \) increases.

4.4. Effectiveness of the O-AUV-Based Multiple Trajectory Planning

In this subsection, we use ANSGA to confirm the efficiency of O-AUV-based multi-trajectory planning. The sensor and environment settings for all parameters are identical to those reported in Section 4.3. As shown in Table 3, ANSGA’s parameters are set.

Table 3. Parameter settings of ANSGA.

| Parameter Settings                          | Value   |
|--------------------------------------------|---------|
| population                                 | 200     |
| Maximum number of evolutions                | 1000    |
| Basic mutation probability \( p_b \)        | 0.7     |
| Number of O-AUV \( N_O \)                  | 5       |
| Target number \( K \)                      | 40      |
| Energy cost limitation of single O-AUV \( E_O \) | 120 m   |
| Energy cost limitation of all O-AUVs \( E_T \) | 700 m   |
Figure 14a,b show the comparison plots of the trajectories obtained by the full traversal of the target points in Figure 14 with conventional algorithms AUV+NSGA-II and O-AUV+NSGA-II, respectively. Each trajectory has a minimum number of 1000 evolutions, and the trajectories correspond to the smallest $p_1$ in the Pareto optimal solution, with $p_1 = 244$ m(O-AUV) and $p_1 = 276$ m(AUV). Among the trajectories, Figure 14a is the conventional AUV trajectory planning and Figure 14b shows the O-AUV trajectory planning. The figures depict the top views with different colored line segments which represent the trajectories of various robots. In Figure 14b, the O-AUV does not have to move completely above T(1) in detecting T(1), but to reach one of its COPs. Moreover, instead of traversing each point, the O-AUV reaches a common COP in detecting T(2), T(3), and T(4). Thus, less time is required to process the data. Above all, a trajectory traversed by the O-AUV will be more rounded, less polyline, and have a shorter distance than the traditional AUV, thereby resulting in less energy consumption and shorter working time.

Figure 14. Comparison plots of the trajectories.

Figure 15 shows the Pareto solution set ($I_{rand} = 1$) for 1000 iterations of running NSGA-II using the O-AUV and conventional AUV, with the blue curve being the Pareto-optimal frontier for the conventional AUV and the red curve being the Pareto-optimal frontier for the O-AUV proposed in this work. The red curve is closer to the X-Y than the blue curve and indicates that obtaining an overall superior plan to conventional AUVs using O-AUV is possible. In other words, the two-optimization metrics, $p_1$ and $p_2$, are better than those of the conventional AUV.

Figure 15. The Pareto front of the AUV+NSGA-II and O-AUV+NSGA-II.
Table 4 illustrates the results of the comparison between the O-AUV and the normal AUV performing the task under the same condition of PF = 2.5 m. We experimented 10 times and averaged the results, then compared them. The results show a 6.43% reduction in $p_1$ and a 22.13% in $p_2$, which illustrates the efficiency of the O-AUV model. The experimental data were attached to the Appendix A.

|                | The Average Value of $p_1$ | The Average Value of $p_2$ |
|----------------|---------------------------|---------------------------|
| Normal AUV     | 293.5584 m                | 82.19484 m                |
| O-AUV          | 274.69 m                  | 64.00 m                   |
| Ratio          | −6.43%                    | −22.13%                   |

We also conducted comparison experiments with different algorithm combinations using the O-AUV+ANSGA optimization framework proposed in this study. The experimental results appear in Figure 16. Here, each optimization curve is calculated as shown in the figure, where the horizontal coordinate represents the optimized algebra and the vertical coordinate is the total driving distance $p_1$. Among the symbols, “O-AUV+ANSGA” stands for the optimization of the multi-robot trajectory using the O-AUV perceptual model and the ANSGA optimization framework mentioned in Section 3.5.

Figure 15 reveals that three optimization curves of the proposed O-AUV perception model are always below those of the traditional AUV model in the optimization process. Thus, the O-AUV perception model can effectively minimize the energy consumption lower bound of the system trajectory. Additionally, a comparison test between the NSGA-II framework and the DE-C-ACO framework proposed in [8] indicated that the NSGA-II converges faster in optimizing the $p_1$ value because of the simpler NSGA-II framework and the more efficient iterative optimization. Meanwhile, the total energy consumption lower bound of the DE-C-ACO algorithm is higher than that of our proposed algorithm for several reasons. First, DE-C-ACO sacrifices some environmental accuracy through the clustering of the environment to reduce the computational effort. Second, the $A^*$ algorithm adopted as the final algorithm cannot guarantee that its planned trajectory is globally optimal. Finally, the algorithm based on the ANSGA framework is basically consistent with the NSGA-II algorithm after multiple optimization iterations, but the former has a significantly faster convergence rate, thereby proving that the proposed accelerated coefficient of variation based on crowding distance may play a role in accelerating the search. In summary, the proposed O-AUV+ANSGA framework can greatly decrease the energy consumption of the underwater multi-robot system and effectively improve the system’s search speed while ensuring that the global optimal solution can be found.
The results of Figure 16 are shown in Table 5. When \( p_1 = 273.27 \text{ m} \), it will take 1000 generations for AUV to use the NSGA-II algorithm, 400 generations for O-AUV to use the DE-C-ACO algorithm, 350 generations for O-AUV to use the NSGA algorithm and 200 generations for O-AUV to use the ANSGA algorithm. It can be seen that the efficiency of O-AUV using ANSGA algorithm is 80%, 50% and 42.86% higher than the other three methods respectively. When \( p_1 = 249.18 \text{ m} \), the ANSGA algorithm is 60% more efficient than the DE-C-AO algorithm and 33.33% more efficient than the NSGA-II algorithm. This shows that the model and algorithm presented in this paper are more efficient.

Table 5. Comparison between different algorithms and models.

| Different Algorithms | \( p_1 \) (Generations = 1000) | \( p_1 = 273.27 \text{ m} \) | \( p_1 = 249.18 \text{ m} \) |
|----------------------|-------------------------------|-----------------|-----------------|
| AUV+NSGA-II          | 273.27 m                      | 1000            |                 |
| O-AUV+DE-C-ACO       | 249.18 m                      | 400             | 1000            |
| O-AUV+NSGA-II        | 234.36 m                      | 350             | 600             |
| O-AUV+ANSGA          | 231.24 m                      | 200             | 400             |

Furthermore, we perform experiments on the multi-trajectory planning system proposed in this work by aiming at different \( PF \) values. Figure 17a is the 3D trajectory planning rendering for \( PF = 1.5 \text{ m} \) and Figure 17b is for \( PF = 2.5 \text{ m} \). The blue dots represent the points to be explored, and the gray dots are the COPs. Different colored trajectories represent the action trajectories of different robots. The trajectory planning results vary with different \( PF \) values. The smaller the \( PF \) value, the closer the results are compared to the optimization results of the traditional AUV (Figure 17a).

![Figure 17a](image1.png)  
(a) \( PF = 1.5 \text{ m} \).

![Figure 17b](image2.png)  
(b) \( PF = 2.5 \text{ m} \).

Figure 17. Trajectories of O-AUV+ANSGA with different \( PF \) values.
5. Related Works

Much research has been done in the field of underwater robot path planning, resulting in corresponding algorithms for different situations. In recent decades, lots of classical algorithms have been applied in the path planning of AUVs, including Dijkstra [19], A* [20], GA [21–23], Fast Marching (FM) [24] and Artificial Potential Field (APF) [25] et al.

Many new results have also been produced in recent years. For trajectory planning of a single AUV, Sun et al. [26] presented an optimized fuzzy control algorithm for three-dimensional AUV path planning. Fan et al. [27] enhanced the APF algorithm to solve some inherent shortcomings of original algorithm in path planning. Cao et al. [28] proposed a method based on genetic algorithm to search the optimal path of AUV, which greatly accelerates the speed of convergence, ensuring the requirements of the time limit. Teng et al. [29] used terrain-aided navigation for localization and trajectory planning of AUVs. Sands [30] developed a proposed approach of deterministic artificial intelligence to control the motion of unmanned underwater vehicles.

For trajectory planning of multi-AUV systems, Yu et al. [8] proposed an algorithm, called DE-C-ACO, which is effectively for surface point location and voyage generation. Wang et al. [16] improved particle swarm algorithm enables effective obstacle avoidance behavior in the path planning process. Zhu et al. [31] embedded a bio-inspired neural network into the SOM neural network to plan the path of the AUV system. Cui et al. [32] applied an adaptive path planning algorithm for multiple AUVs to estimate the scalar field over a region of interest.

Most of these studies are based on one sensor model for performing tasks, which will complicate the trajectory of the actuation process. An O-AUV sensing model is proposed in our paper that can perceive the omnidirectional area, and detect a larger area than the traditional AUV. This model has the omni-directional sensing capability that allows more efficient completion of tasks without reaching task points through our modified ANSGA algorithm, fundamentally improving the above problem.

6. Conclusions

In this paper, we present an O-AUV, an omnidirectional area perception-based AUV that can be widely used according to different demands. We propose a 3D O-AUV sensing model for the O-AUV. A cube-based environment modeling, the calculation of candidate observation points, and voyage cost map building methods are also proposed for adopting the model. In addition, we modified the ANSGA algorithm to accelerate the search speed for the optimal solution. Extensive experiments have been performed to study the effectiveness of our algorithm. The test shows that with $PF = 2.5 \, \text{m}$, the total distance can be reduced by 22.13% on average. Under the condition of using the O-AUV model, compared with other algorithms, the efficiency of the ANSGA algorithm is increased by 60% at most.

**Author Contributions:** Data curation, Z.J.; Formal analysis, J.R.; Funding acquisition, C.W. and H.W.; Methodology, J.R.; Project administration, C.W. and H.W.; Software, Y.L.; Supervision, J.R., H.X. and H.W.; Visualization, H.W.; Writing—original draft preparation, J.R.; Writing—review and editing, S.Y. and Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant nos. U20A20197, 61973063, U1713216, Northeast University Postdoctoral Fund 20210303, Liaoning Key Research and Development Project 2020JH2/10100040, the Foundation of National Key Laboratory OEIP-O-202005 and the Fundamental Research Fund for the Central Universities of China N2126002, N2126006.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available in Appendix A.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to this work.
Appendix A

Table A1. Data used in the experiment (provided as coordinates).

| X  | 8  | 22 | 25 | 26 | 20 | 18 | 9  | 11 | 18 | 23 | 14 | 11 | 27 | 25 | 15 | 16 | 4  | 16 | 25 | 24 | 23 | 11 | 27 | 11 | 3  | 29 | 32 | 16 | 22 | 29 | 15 | 24 | 30 | 26 | 11 | 27 | 2 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| Y  | 15 | 5  | 23 | 23 | 28 | 5  | 15 | 30 | 8  | 3  | 30 | 20 | 32 | 2  | 2  | 2  | 31 | 32 | 10 | 30 | 24 | 11 | 11 | 35 | 26 | 32 | 14 | 11 | 26 | 31 | 33 | 7  | 35 | 3  | 25 | 18 | 3  | 3  | 19 | 7 |

Experimental Data for Task Points
References

1. Williams, S.B.; Pizarro, O.; Johnson-Roberson, M.; Mahon, I.; Webster, J.; Beaman, R.; Bridge, T. AUV-assisted surveying of relic reef sites. In Proceedings of the OCEANS 2008, Quebec City, QC, Canada, 15–18 September 2008; pp. 1–7. [CrossRef]

2. Chatzichristofis, S.; Kapoutsis, A.; Kosmatopoulos, E.; Doitsidis, I.; Rovas, D.; Borges de Sousa, J. New insights into the formation of submarine glacial landforms from high-resolution Autonomous Underwater Vehicle data. Geomorphology 2020, 370, 1–17. [CrossRef]

3. Cao, J.; Su, Y.; Zhao, J. Design of an adaptive controller for dive-plane control of a torpedo-shaped AUV. J. Mar. Sci. Appl. 2011, 10, 333–339. [CrossRef]

4. Loc, M.B.; Choi, H.S.; Seo, J.M.; Baek, S.H.; Kim, J.Y. Development and control of a new AUV platform. Int. J. Control. Autom. Syst. 2014, 12, 886–894. [CrossRef]

5. Tsiokgas, N.; Saigol, Z.; Lane, D. Distributed multi-AUV cooperation methods for underwater archaeology. In Proceedings of the OCEANS 2015-Genova, Genoa, Italy, 18–21 May 2015; pp. 1–5. [CrossRef]

6. Fang, Z.; Wang, J.; Jiang, C.; Zhang, Q.; Ren, Y. Aol inspired collaborative information collection for AUV assisted Internet of underwater things. IEEE Internet Things J. 2021, 1–13. [CrossRef]

7. Tiwari, S. Mosaicking of the ocean floor in the presence of three-dimensional occlusions in visual and side-scan sonar images. In Proceedings of the Symposium on Autonomous Underwater Vehicle Technology, Monterey, CA, USA, 2–6 June 1996; pp. 308–314. [CrossRef]

8. Yu, X.; Chen, W.N.; Hu, X.M.; Gu, T.; Yuan, H.; Zhou, Y.; Zhang, J. Path Planning in Multiple-AUV Systems for Difficult Target Traveling Missions: A Hybrid Metaheuristic Approach. IEEE Trans. Cogn. Dev. Syst. 2020, 12, 561–574. [CrossRef]

9. Neto, E.C.; Sá, R.C.; Holanda, G.C.; Da Mata, F.A.; Varela, A.T.; Araújo, A.L.; Loiola, I.J.; De Alexandria, A.; De Albuquerque, V. Autonomous underwater vehicle to inspect hydroelectric dams. Int. J. Comput. Appl. 2014, 101, 1–11. [CrossRef]

10. Jiang, C.; Sun, L.; Sun, Y.s. Design of motion control system of pipeline detection AUV. J. Cent. South Univ. 2017, 24, 637–646. [CrossRef]

11. Wulff, T.; Bauerfeind, E.; von Appen, W.J. Physical and ecological processes at a moving ice edge in the Fram Strait as observed with an AUV. Deep Sea Res. Part I Oceanogr. Res. Pap. 2016, 115, 253–264. [CrossRef]

12. Forrest, A.L.; Lund-Hansen, L.C.; Sorrell, B.K.; Bowden-Floyd, I.; Lucker, V.; Cossu, R.; Lange, B.A.; Hawes, I. Exploring spatial heterogeneity of Antarctic sea ice algae using an autonomous underwater vehicle mounted irradiance sensor. Front. Earth Sci. 2019, 7, 1–13. [CrossRef]

13. Campbell, K.J.; Kinnear, S.; Thame, A. AUV technology for seabed characterization and geohazards assessment. Lead. Edge 2015, 34, 170–178. [CrossRef]

14. Matos, A.; Martins, A.; Dias, A.; Ferreira, B.; Almeida, J.M.; Ferreira, H.; Amaral, G.; Figueiredo, A.; Almeida, R.; Silva, F. Multiple robot operations for maritime search and rescue in euRathlon 2015 competition. In Proceedings of the OCEANS 2016, Shanghai, China, 10–13 April 2016; pp. 1–7. [CrossRef]

15. Maqsood, H.; Javaid, N.; Yahya, A.; Ali, B.; Khan, Z.A.; Qasim, U. Mobile-AUV: AUV-Aided Localization Scheme for Underwater Wireless Sensor Networks. In Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Fukuoka, Japan, 6–8 July 2016; pp. 170–175. [CrossRef]

16. Wang, H.; Yuan, J.; Lv, H.; Li, Q. Task allocation and online path planning for AUV swarm cooperation. In Proceedings of the OCEANS 2017, Aberdeen, Scotland, 19–22 June 2017; pp. 1–6. [CrossRef]

17. Xue, T.; Li, R.; Tokgo, M.; Ri, J.; Han, G. Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm. Soft Comput. 2017, 21, 2421–2437. [CrossRef]

18. Ru, J.; Jia, Z.; Yang, Y.; Yu, X.; Wu, C.; Xu, M. A 3D coverage algorithm based on complex surfaces for UAVs in wireless multimedia sensor networks. Sensors 2019, 19, 1902. [CrossRef] [PubMed]

19. Arinaga, S.; Nakajima, S.; Okabe, H.; Ono, A.; Kanayama, Y. A motion planning method for an AUV. In Proceedings of the Symposium on Autonomous Underwater Vehicle Technology, Monterey, CA, USA, 2–6 June 1996; pp. 477–484. [CrossRef]

20. Garau, B.; Alvarez, A.; Oliver, G. Path Planning of Autonomous Underwater Vehicles in Current Fields with Complex Spatial Variability: An A* Approach. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 194–198. [CrossRef]

21. Chang, Z.H.; Tang, Z.D.; Cai, H.G.; Shi, X.C.; Bian, X.Q. GA path planning for AUV to avoid moving obstacles based on forward looking sonar. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; pp. 1498–1502. [CrossRef]

22. Zhang, Q. A Hierarchical Global Path Planning Approach for AUV Based on Genetic Algorithm. In Proceedings of the 2006 International Conference on Mechatronics and Automation, Luoyang, China, 25–28 June 2006; pp. 1745–1750. [CrossRef]

23. Alvarez, A.; Cati, A.; Onken, R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. IEEE J. Ocean. Eng. 2004, 29, 418–429. [CrossRef]

24. Petres, C.; Pailhas, Y.; Patron, P.; Petillot, Y.; Evans, J.; Lane, D. Path Planning for Autonomous Underwater Vehicles. IEEE Trans. Robot. 2007, 23, 331–341. [CrossRef]
25. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; pp. 500–505. [CrossRef]

26. Sun, B.; Zhu, D.; Yang, S.X. An optimized fuzzy control algorithm for three-dimensional AUV path planning. *Int. J. Fuzzy Syst.* 2018, 20, 597–610. [CrossRef]

27. Fan, X.; Guo, Y.; Liu, H.; Wei, B.; Lyu, W. Improved artificial potential field method applied for AUV path planning. *Math. Probl. Eng.* 2020, 2020, 1–21. [CrossRef]

28. Cao, J.; Li, Y.; Zhao, S.; Bi, X. Genetic-Algorithm-Based Global Path Planning for AUV. In Proceedings of the 2016 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 10–11 December 2016; pp. 79–82. [CrossRef]

29. Teng, M.; Ye, L.; Yuxin, Z.; Yanqing, J.; Zheng, C.; Qiang, Z.; Shuo, X. An AUV localization and path planning algorithm for terrain-aided navigation. *ISA Trans.* 2020, 103, 215–227. [CrossRef] [PubMed]

30. Sands, T. Development of Deterministic Artificial Intelligence for Unmanned Underwater Vehicles (UUV). *J. Mar. Sci. Eng.* 2020, 8, 578. [CrossRef]

31. 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]

32. 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]