sustainability

MDPI

Article

Heuristic Surface Path Planning Method for AMV-Assisted Internet of Underwater Things

Jie Zhang, Zhengxin Wang, Guangjie Han * and Yujie Qian

Department of Internet of Things Engineering, Hohai University, Changzhou 213022, China
* Correspondence: hanguangjie@gmail.com

Abstract: Ocean exploration is one of the fundamental issues for the sustainable development of human society, which is also the basis for realizing the concept of the Internet of Underwater Things (IoUT) applications, such as the smart ocean city. The collaboration of heterogeneous autonomous marine vehicles (AMVs) based on underwater wireless communication is known as a practical approach to ocean exploration, typically with the autonomous surface vehicle (ASV) and the autonomous underwater glider (AUG). However, the difference in their specifications and movements makes the following problems for collaborative work. First, when an AUG floats to a certain depth, and an ASV interacts via underwater wireless communication, the interaction has a certain time limit and their movements to an interaction position have to be synchronized; secondly, in the case where multiple AUGs are exploring underwater, the ASV needs to plan the sequence of surface interactions to ensure timely and efficient data collection. Accordingly, this paper proposes a heuristic surface path planning method for data collection with heterogeneous AMVs (HSPP-HA). The HSPP-HA optimizes the interaction schedule between ASV and multiple AUGs through a modified shuffled frog-leaping algorithm (SFLA). It applies a spatial-temporal k-means clustering in initializing the memeplex group of SFLA to adapt time-sensitive interactions by weighting their spatial and temporal proximities and adopts an adaptive convergence factor which varies by algorithm iterations to balance the local and global searches and to minimize the potential local optimum problem in each local search. Through simulations, the proposed HSPP-HA shows advantages in terms of access rate, path length and data collection rate compared to recent and classic path planning methods.

Keywords: autonomous marine vehicles; data collection; heuristic surface path planning; time-sensitive interaction; shuffled frog-leaping algorithm

Citation: Zhang, J.; Wang, Z.; Han, G.; Qian, Y. Heuristic Surface Path Planning Method for AMV-Assisted Internet of Underwater Things. *Sustainability* **2023**, *15*, 3137. https://doi.org/10.3390/su15043137

Academic Editor: Ripon Kumar Chakrabortty

Received: 4 January 2023
Revised: 27 January 2023
Accepted: 7 February 2023
Published: 8 February 2023

1. Introduction

With the growing interest in futuristic ocean technology concepts proposed in recent years, such as the oceanix city [1], smart coastal [2], underwater smart city [3], etc., the intelligent control technology for marine vehicles is becoming one of the most important issues in the implementation of the Internet of Underwater Things (IoUT) technologies. Accordingly, autonomous marine vehicles (AMVs) are rapidly developing and are widely used in practical tasks, typically the autonomous surface vehicle (ASV), autonomous underwater glider (AUG), autonomous underwater vehicle (AUV), etc., which are commonly used in seabed exploration to collect information on the distribution of underwater resources [4–6]. Accordingly, technical issues for controlling the AMVs, such as multi-vehicle collaboration, path planning, obstacle avoidance, etc., have been introduced in recent years to enable practical underwater exploration, and researchers have studied the innovation in terms of algorithm design and system frameworks to find optimal solutions for efficient, safe and energy-saving marine exploration tasks [7–9].

Depending on the characteristics of different subsea exploration tasks, AMVs are used in different application scenarios. ASV usually plays the role of relaying exploration data from the subsea and, therefore, tracks AMVs underwater on the sea surface. An AUV is

usually equipped with a propeller drive, has a high speed and is often used when tracking moving objects or when urgent detection is required [10]. An AUG is driven by buoyancy, which is slower in speed but lower in energy cost and is usually used when long-duration underwater exploration tasks in a large submarine area are required. It glides to a certain depth, performs an underwater detection task and then floats to the surface or to a certain depth for the next movement and meanwhile interacts with an ASV or land station [11–13]. This paper addresses the collaboration scenario between an ASV and AUG.

Figure 1 shows an example scenario of data collection by ASV and AUG collaboration. Multiple AUGs travel underwater to collect subsea data by a predefined trajectory, and they float up to a certain depth in each gliding cycle to interact with the ASV on the sea surface, forming a series of temporal interaction points underwater. There are two exploration routes with AUG exploration tasks, and the AUGs sail at different speeds so that the ASV has to plan a surface path at the interaction points that occur at different times. In previous works, the purposes of ASV path planning commonly target obstacle avoidance, energy efficiency, path length optimization, etc. [14–16] with fixed interaction points. However, in such a scenario with time-sensitive interaction points between ASV and AUGs, the ASV has to schedule an efficient sequence of access to the interaction points of AUGs that occur at specific times. Therefore, the ASV requires a path planning approach that takes into account time-sensitive points. For such a task, this paper proposes a Heuristic algorithm-based Surface Path Planning method for underwater data collection with heterogeneous AMVs (HSPP-HA).

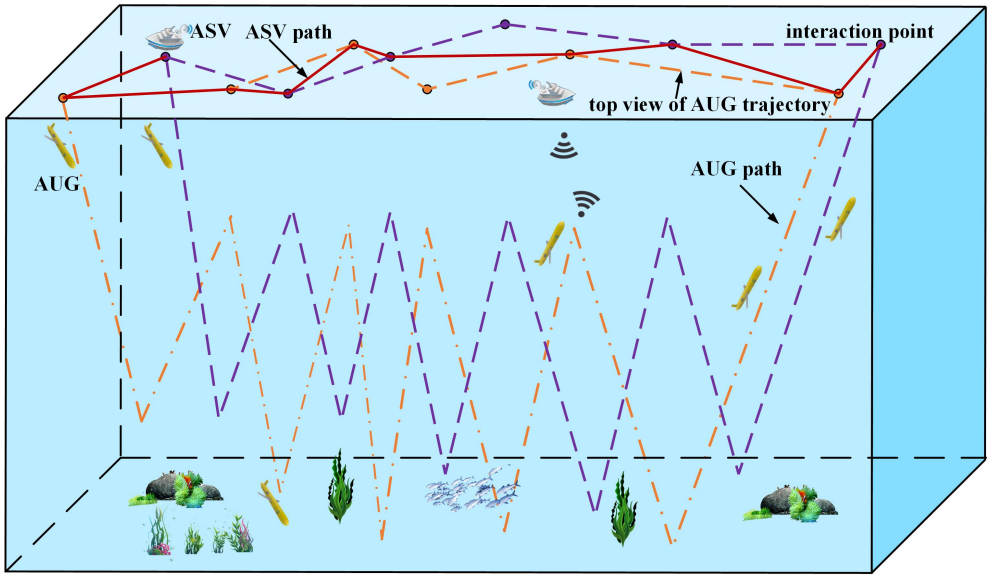


Figure 1. An example of collaboration between ASV and multiple AUGs.

Heuristic algorithms have been widely applied in the field of robot path planning [17,18], where each search individual is updated according to the surrounding environment, and then information is exchanged between individuals to determine the overall optimization direction of the population. For example, the thermal exchange optimization (TEO) [19] algorithm is a heuristic algorithm based on Newton's law of cooling, where the temperature represents a location and the search agent updates its temperature through heat exchange and heat transfer; the water strider algorithm (WSA) [20] mimics the life cycle of water striders, where each water strider represents a location and the strider populations exchange information with each other through ripples to find the food, which represents the best solution in a region; the shuffled frog-leaping algorithm (SFLA) [21] is a classical one that emulates the frog foraging behavior, where frogs seek different leaping points in a certain area to find food, during which each frog exchanges information through sharing their memes. The proposed HSPP-HA modifies the SFLA to adopt the data collection

scenario of ASV and AUG collaboration. First, the interaction points, which refers to food, are clusters by a spatial-temporal k-means algorithm to initialize the input of the SFLA; then, an adaptive iteration factor is designed so the SFLA adopts the heuristic iteration into the path planning scenario; and finally, the amount of information carried by each AUG, distance from ASV to each interaction point and required interaction period in each interaction point are analyzed by several constraints in iterative optimization of the SFLA.

This paper has made the following contributions:

1. A surface path planning method for underwater data collection with heterogeneous AMVs, named HSPP-HA, is proposed, which targets an application scenario where underwater data are collected through the collaboration of an ASV on the surface and multiple AUGs underwater based on underwater wireless communication and applies a modified SFLA to schedule the data collection between ASV and AUGs for the time-sensitive interactions between them;
2. An improved SFLA is designed with a spatial-temporal k-means algorithm and an adaptive iteration approach. The spatial-temporal k-means algorithm clusters the interaction points by their coordinates and times of occurrence to initialize the local searches; meanwhile, an adaptive iteration factor enables balanced local and global searches in optimizing the sequence of interactions and, furthermore, improves the convergence ability.

The remainder of this paper is organized as follows. Section 2 introduces related work; Section 3 describes the system model, including the task model, constraint model and objective optimization model; Section 4 describes the detail of the proposed HSPP-HA; Section 5 presents simulations to analyze the performance of the HSPP-HA; and the conclusion follows in Section 6.

2. Related Works

The challenge of the proposed data collection scenario is that the ASV needs to access a series of time-sensitive interaction points on the sea surface, which can be associated with path planning [22], task assignment [23] and the traveling salesman problem (TSP) [24]. Some recent related research works are introduced below.

Chen et al. [25] proposed a task assignment and path planning scheme for multiple AUVs to access multiple targets. The scheme targets optimizing the total sailing distance of AUVs and the balance of moving tasks and uses an algorithm that attracts the AUV to the static task point and pushes it away from the obstacles. Zhu et al. [26] also studied the assignment of multiple targets to be accessed by multiple AUVs, which used a bio-inspired neural network graph (BINN) to calculate the activity values of all AUVs for each target and select the ones with high activity values for assignment, target and obstacle locations were also calculated for activity values to guide the AUVs to travel. Wu et al. [27] proposed a scheme for multiple AUVs to perform multiple rescue missions, which is based on reinforcement learning (RL) to obtain different rewards based on the real-time environment and assign suitable rescue missions to AUVs; multiple rescue regions exist in each rescue mission, and then a particle swarm optimizer (PSO) is executed to plan the optimal rescue path for the AUVs. Wang et al. [28] proposed a global and local path planning scheme for the ASV, which applies the Theta* to plan a collision-free global path and uses fuzzy decision-making and fine dynamic window to perform local optimization to avoid obstacles, and then returns to the global path to continue traveling after avoiding the obstacles. Hu et al. [29] proposed a collision-free path planning scheme that complies with the convention on international regulations for preventing collisions at sea (COLREGs) with a multi-objective PSO algorithm. In their optimization process, priority is given to changing the ASV heading or velocity magnitude before considering the path length or path smoothness. The contributions of this study show that path-planning techniques are being innovated to design corresponding solutions for different problems in practice.

The path planning and task assignment contributions, including the above-mentioned works, present solutions in terms of planning collision-free paths for a single robot, planning

for multiple robots to access multiple target points and global and local path planning for robots, but rarely consider time-sensitive accesses to multiple task points. The task of targeted application background in this paper is similar to TSP, with the difference that the task points need to consider both distance and time, and the algorithm designed needs to ensure both a low total travel distance and access to as many task points as possible. The following works introduce the contributions for solving TSP in application scenarios of finding single or multiple routes.

Jian et al. [30] studied the multimodal optimization on TSP and proposed a niching regression algorithm that uses a linear regression mechanism in partitioning the environment, with each partition containing some task points, and the memetic algorithm is applied to perform local search and finally determine multiple optimal TSP, which demonstrates that it is feasible to solve multi-solution TSP using an effective meta-heuristic algorithm. Cai et al. [31] studied the multi-objective TSP and proposed a Lin–Kernighan heuristic algorithm, which uses the idea of decomposition to split the main problem into multiple sub-problems and search for knowledge to transfer between individuals to finally determine the optimal solution. Zhang et al. [32] studied dynamic TSP and proposed a deep reinforcement learning algorithm with strong feature extraction capability for the environment, which can quickly learn to return favorable actions for changing environments. The task points in this study are dynamically changing, which is different from the traditional static task points. Sanyal et al. [33] studied large-scale TSP, and as the scale of TSP increases, it leads to sub-optimal solution quality. They proposed a heuristic approach based on Neuro-Ising, where the Neuro layer is a clustering operation of task points using a graph neural network, and the Ising layer uses an Ising solver to solve each subregion in parallel.

For solving TSP, various optimization solutions from different perspectives are presented, and they have solved related types of problems, such as large-scale TSP, multi-solution TSP, multi-objective TSP and dynamic TSP. However, in practical tasks, there are situations where the task point to be accessed is time-sensitive, and if it is not accessed within its timeframe, then the information may be lost, or the task may fail. Therefore, our work considers dynamic task assignments, and the optimization approach needs a trade-off between the time and distance of interaction points. Accordingly, this paper presents the design of an objective optimization model with an adaptive heuristic algorithm to plan a path solution for ASV to access multiple interaction points of underwater AUGs.

3. System Model

This section describes the system model of the proposed HSPP-HA, including the task model, the constraint model and the objective optimization model.

3.1. Task Model

The task model of the proposed HSPP-HA is described as follows. Multiple AUGs collect data on the seafloor, and each AUG floats to a certain plane during each cycle to transfer data to the ASV on the surface, which generates a series of interaction points with different interaction times. The AUGs will buffer the collected data into their storages if the ASV does not reach the interaction point when they float to the interaction points, and if the ASV misses the interaction point several times, the data will be lost. Figure 2 represents a series of interaction points generated from three AUGs on the water surface, where the numbers are the order in which the AUGs move to the interaction points, and each AUG knows the coordinates of interaction points, the sequence of time of occurrence and the amount of data they carried. The blue, black and green dotted lines represent the top view of the trajectories of three AUGs, and the red solid line is the path of the ASV to each of the interaction points. The number next to each interaction point is their sequence, which is associated with the time of their occurrence.

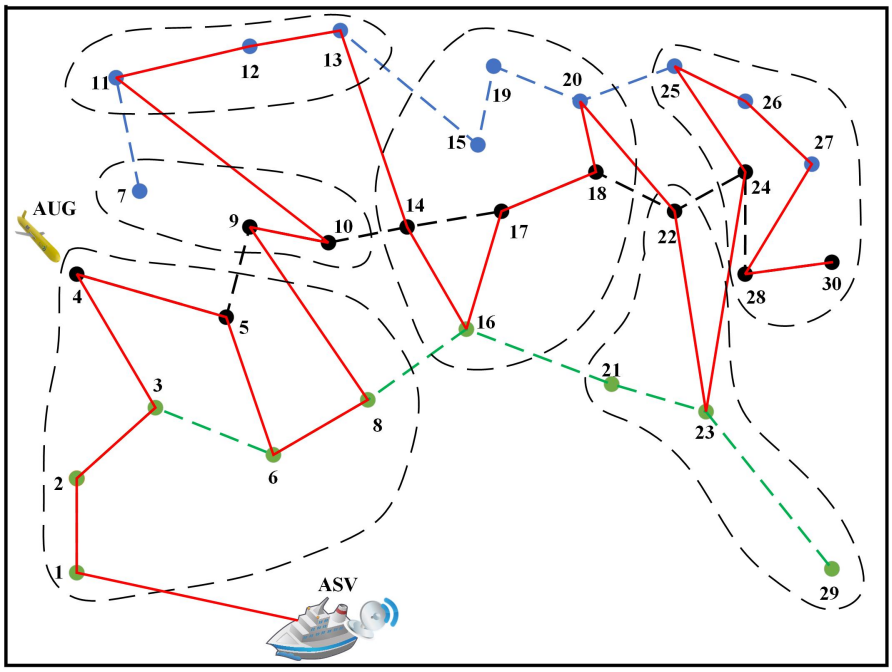


Figure 2. An example diagram of the task model.

The purpose of HSPP-HA is to guide the ASV to each interaction point and maximize data collection by optimizing the schedule for reaching the interaction points, which requires considering not only the location of the interaction points but also the time of their occurrence. Therefore, for the example scenario shown in Figure 2, the global interaction points are clustered by their spatial-temporal properties, and the local searches are performed with the heuristic algorithm to find the most optimal interaction schedule while maximizing global data collection. There are some interaction points that are not scheduled by the path of ASV, and those are the cases where ASV passes some interactions on purpose in order to globally optimize data collection. The missed data may be collected by the next interaction point of the same AUG if its buffer does not reach the threshold; if not, the data are missed but still maximizes the global data collection.

3.2. Constraint Model

The interaction points have three factors, time, location and the information amount. Accordingly, the path planning of HSPP-HA considers the constraints of time of occurrence (C_i^t), moving speed ($C_{i\to j}^v$) and information load (C_i^q) as follows.

C_i^t: Each interaction point occurs at a different time and should be given a different priority. The time of occurrence of each interaction point is denoted as a time quantum, expressed as:

$$t_i = \begin{cases} 0, \text{ ASV has reached} \\ 0, \text{ ASV decides not to access} \\ 1 - \frac{i-1}{N}, \text{ else} \end{cases} \quad (1)$$

where t_i is a time quantum from 0 to 1; i is the order of time of occurrence of each interaction point; N is the number of interaction points. C_i^t decides priorities of $t_1 > t_2 > \cdots > t_N$, which guarantees that the interaction point with a larger t_i is given a higher access priority, and if the interaction point has been accessed or decided not to be accessed, then the t_i will be reset to 0.

$C^v_{i\to j}$: To ensure that an ASV can reach an interaction point before it disappears, it should have enough speed to sail between interaction points i and j within a specific time period, expressed as:

$$\frac{d_{ij}}{|T_i - T_j|} \in (0, v_{max}] \tag{2}$$

where d_{ij} is the distance between interaction points i and j; T is the time when an interaction point occurs.

C^q_i: Each interaction point provides a different amount of data because some AUGs may not be able to interact with the ASV during an interaction cycle, thus buffering the data to the next interaction point [34,35]. However, the buffer storage should be limited to ensure that the data in the buffer can be transferred within an interaction period, expressed as:

$$q_i \le \frac{BI_i}{D} \tag{3}$$

where q_i is the count of data collected by an AUG i, which is reset to 0 when it completes an interaction; D is the amount of data collected by one interaction cycle; B is the bandwidth of the wireless channel from AUG to ASV; and I_i is the interaction period for i.

In the proposed HSPP-HA, it is assumed that the ASV knows the information of each interaction point, including the occurrence order, coordinates, the times of occurrence t_i, amount of information q_i, the serial number i and its neighboring interaction points j follow $C^v_{i\to j}$.

3.3. Objective Optimization Model

The optimization objective of the proposed HSPP-HA are t_i and q_i introduced in the constraints of C^t_i and C^q_i, respectively, and a gain factor N_i of accessing an interaction point. N_i is related to the distance between an interaction point and its surrounding interaction points follow constraint $C^v_{i\to j}$ in the next interaction period. An ASV can get higher gains in data collection with higher N_i because there are more potential interactions in the next interaction period. The objective optimization model of HSPP-HA mainly weighs between t_i and N_i, while the q_i is used as a coefficient to indicate the degree of urgency of an interaction point, and a fitness value f_i of an interaction point is expressed by an optimization function as follows:

$$f_i = \frac{1}{\mu_i \times (\omega_1 t_i + \omega_2 N_i)} \tag{4}$$

where ω_1 and ω_2 are coefficients that follow $\omega_1 + \omega_2 = 1$; μ_i is a degree factor of q_i, indicating the possibility of data loss, which is expressed as $\mu_i = 1 - \frac{1}{2^{q_i}}$; and N_i is the above-mentioned gain factor, which is expressed as follows:

$$N_i = \frac{|d_{asv\to q}| + (\min_{j\in\{Q-q\}} |d_{asv\to j}|) \times (n-1)}{|d_{asv\to i}| + \sum_{j\in\{Q-i\}} |d_{i\to j}|} \tag{5}$$

$$q = \arg\min_{j\in Q} |d_{asv\to j}| \tag{6}$$

where $d_{a\to b}$ is the distance between a and b; Q is a set of interaction points surrounding i and follow constraint $C^v_{i\to j}$; j is each interaction point in Q; n is the number of interaction points in Q. Figure 3 illustrates an example case where ASV decides an interaction points to access, which describes the meaning of N_i.

In Figure 3, the closest interaction point to ASV is 6, and the closest one to interaction point 6 is 8. In the case of the N_7, there are five interaction points following constraint $C^v_{5\to j}$, $j \in Q$, therefore, N_7 is related to $|d_{asv\to 6} + 4 \times d_{6\to 8}|$, and the gain factor is calculated by $N_7 = \frac{|d_{asv\to 6} + 4\times d_{6\to 8}|}{|d_{asv\to 7} + \sum_{j\in\{6,8,9,10\}} d_{7\to j}|}$.

Finally, an ASV selects an interaction point by finding the minimum overall fitness value of the interaction point.

$$P = \arg\min_{i \in Q} \left(\sum_{j=0}^{n} \min \left(f_{i+j,(i+j) \in \{Q-i\}} \right) \right) \tag{7}$$

where P is the selected interaction point.

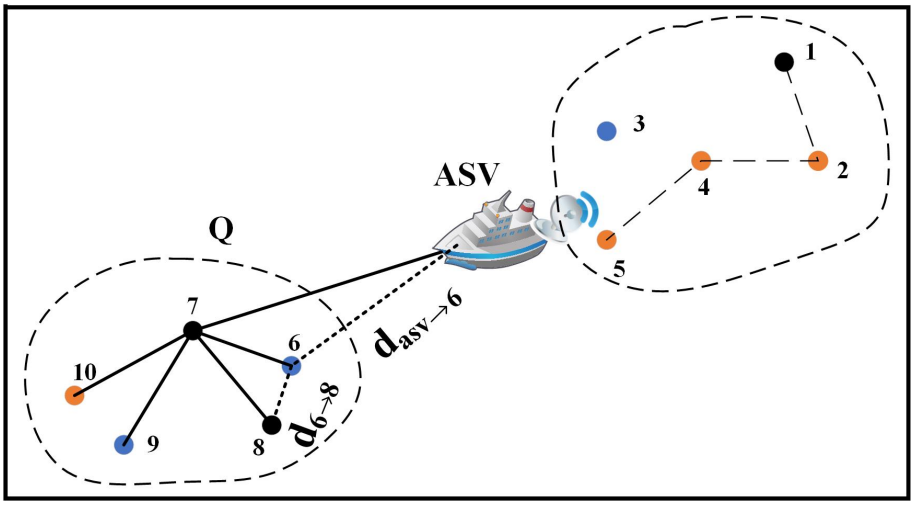


Figure 3. An example of an ASV deciding on an interaction point to access based on the N_i.

4. Heuristic Surface Path Planning Method for Underwater Data Collection

This section presents the proposed HSPP-HA. The HSPP-HA includes a spatial-temporal k-means clustering and an adaptive iteration approach.

The heuristic algorithm is a general framework for solving complex global optimization problems [36], and SFLA [21] is a swarm intelligence heuristic algorithm that combines the advantages of PSO and the memetic algorithm (MA) [37], where each frog exchanges information and the worst frog is updated according to the best frog.

SFLA mainly consists of two parts, which are global search and local search. The local search runs during the global search, and each round of global search is a mixed exchange of frogs at the end of the local search to determine the best contemporary frog. Then, it updates the population, assigns frogs and starts a new round of optimization. The local search is that each memeplex completes the search within the region independently, and the local search in each memeplex does not stop until the maximum number of iterations is reached. When all memeplexes have finished searching, it means that the local search is completed.

However, in the proposed task model, planning the optimal path between each interaction point should consider several attributes, such as location, time, data amount, etc., and multiple AUGs underwater will certainly bring about time-sensitive interaction points, leading to high complexity of traditional SFLA. Furthermore, traditional SFLA is randomly generated for some memeplexes initially, which may result in dividing searching points with results in clustering interaction points with different times of occurrence into a region, and due to the spatial-temporal nature of interaction points, randomly determined regions are not guaranteed to provide a reliable search memeplex. Therefore, an improved SFLA is designed for the task model with time-sensitive interaction points, which works as follows.

4.1. Spatial-Temporal Clustering

The global search process of SFLA searches the memeplexes over the entire region, while the local search process searches for an optimized solution within a memeplex. Hence, the HSPP-HA first clusters the interaction points to form multiple memeplexes, and since the interaction points are time-sensitive; the clustering has to take into account their spatial

and temporal proximities to group the interaction points having similar locations and time of occurrence for efficient local searches. Accordingly, the spatial-temporal clustering proposed in this paper uses the principle of the k-means algorithm and weights the spatial and temporal proximities to form memeplexes. The details are as follows:

Assume there are N interaction points placed in a $\{x, y\}$ region and the interaction points are denoted as $X = \{X_1, X_2, \cdots, X_N\}$, where each interaction point i has the properties of two-dimensional coordinates and their time of occurrence. Since the time of occurrence is denoted as a time quantum t_i according to the constraint model in Section 3.2, the coordinate of each interaction point (x_i, y_i) is normalized as $(\frac{x_i}{x}, \frac{y_i}{y})$, and, therefore, the property of each interaction point i is denoted as $X_i = (\frac{x_i}{x}, \frac{y_i}{y}, t_i)$.

According to the principle of k-means clustering [38], we denote the initial clusters as $M = \left\{M^1, M^2, \cdots, M^k\right\}$ and the related cluster center as R, and then the temporal proximity of an interaction point i to a cluster center j is expressed as:

$$P^T_{X_i \to R_j} = |t_i - t_j| \tag{8}$$

where P^T is the temporal proximity. Then, the spatial proximity between them is expressed as:

$$P^S_{X_i \to R_j} = \begin{cases} \sqrt{\frac{(x_i - x_j)^2}{x^2} + \frac{(y_i - y_j)^2}{y^2}}, \text{ if } v_{i \to j} \text{ follow } C^v_{i \to j} \\ 1, \text{ else} \end{cases} \tag{9}$$

where P^S is the spatial proximity; $v_{i \to j}$ is the required speed for ASV to sail from i to j, which can be calculated by the differences in their times of occurrence and physical coordinates; and $C^v_{i \to j}$ is the speed constraint described in Section 3.2.

Finally, the joint proximity is expressed as:

$$P_{X_i \to R_j} = \eta_1 P^T_{X_i - R_j} + \eta_2 P^S_{X_i - R_j} \tag{10}$$

where P is the joint proximity and η_1 and η_2 are weighting values calculated by solving the following equation:

$$\begin{cases} \eta_1 + \eta_2 = 1 \\ \frac{P^T_{AVG\_M^j}}{P^T_{MAX\_M^j}} \times \eta_2 = \frac{P^S_{AVG\_M^j}}{P^S_{MAX\_M^j}} \times \eta_1 \end{cases} \tag{11}$$

where $P^T_{AVG\_M^j}$ and $P^S_{AVG\_M^j}$ are the averaged temporal and spatial proximity, and $P^T_{MAX\_M^j}$ and $P^S_{MAX\_M^j}$ are the maximum temporal and spatial proximity in M^j, respectively.

The proposed spatial-temporal k-means algorithm applies such joint proximity with time of occurrence and coordinates in clustering the interaction point and the rest of the processes are described by Algorithm 1.

Algorithm 1: Spatial-temporal k-means clustering.

Input : N interaction points ($X_i = \{X_1, X_2, \cdots X_N\}$);
Maximum iteration numbers (h_{max});
The number of regions (k);
Output: Regions ($\left\{M^1, M^2, \cdots M^k\right\}$);

Random selection of k regional centers $R_j = \{R_1, R_2 \cdots R_k\}$ from N interaction points;
for $h = 1$ *to* h_{max} **do**
  Set $M^j = \emptyset, j \in [1, k]$;
  for $i = 1$ *to* N **do**
    Calculate the temporal proximity $P^T_{X_i \to R_j}$ by Equation (8);
    Calculate the spatial proximity $P^S_{X_i \to R_j}$ by Equation (9);
    Calculate the joint proximity $P_{X_i \to R_j}$ of X_i by Equations (10) and (11);
    Determine the region marker $\mu_i = \arg\min_{j \in \{1,2,\cdots,k\}} P_{X_i \to R_j}$ for X_i;
    Divide X_i into the corresponding region $M^{\mu_i} = M^{\mu_i} \cup \{X_i\}$;
  end
  for $j = 1$ *to* k **do**
    Update the new center R^*_j;
    if $R^*_j \neq R_j$ **then**
      Determine the regional center as R^*_j;
    else
      Determine the regional center as R_j;
    end
  end
  $h = h + 1$;
end
Return regions $M^j, j \in [1, k]$

4.2. Shuffled Frog-Leaping with Adaptive Iteration Factor

This subsection introduces the modifications of SFLA with an adaptive iteration factor, and the meaning of the variables associated with HSPP-HA is given in Table 1.

Table 1. Relevant parameters of HSPP-HA.

Parameters	Definition
M^j	Clustered memeplexes, $j \in [1, k]$
k	Number of clusters
n	Number of frogs in a cluster
S	Frog populations = $(k \times n)$
$F(i)$	A frog, $i \in S$
$f(i)$	Path fitness value of a frog, $i \in S$
q	Number of selected frogs in a local search
SM^j	Sub-memeplex with q frogs, $j \in [1, k]$
P_G	Leaping vector of the best frog in each iteration
P_B	Leaping vector of the local optimal frog in an SM
P_W	Leaping vector of the local worst frog in an SM
$F(q)$	The local worst frog in an SM
$F'_i(q)$	New frog after P_W updates, $i \in \{1, 2, 3\}$
$f'_i(q)$	Path fitness value of $F'_i(q)$, $i \in \{1, 2, 3\}$

In our task model, a frog's behavior represents a path solution, which is a sequence of accessed interaction points, and its leaping vector is a path between two interaction points.

The population of frogs is denoted as $\{F(1), F(2), \cdots F(S)\}$ and each $F(i)$ will have a $f(i)$ indicating a fitness value of a frog, which is calculated by Equation (4). The frogs are sorted by ascending order, and the frog having the best fitness value is $F(1)$, denoted as P_G. Note that the P_G is updated with each round of global search until a global optimal solution is obtained. The sorted frogs are assigned to M^j by the following:

$$M^j = \left\{ F^j(i) | F^j(i) = F(j + k \times (i-1)) \right\} \tag{12}$$

where k is the number of memeplexes; j is the sequence of memeplexes; i is the sequence of the frog population. Figure 4 explains the assignment expressed in Equation (12), where the ordered S frogs are sequentially assigned to different M^j until the Sth frog is assigned.

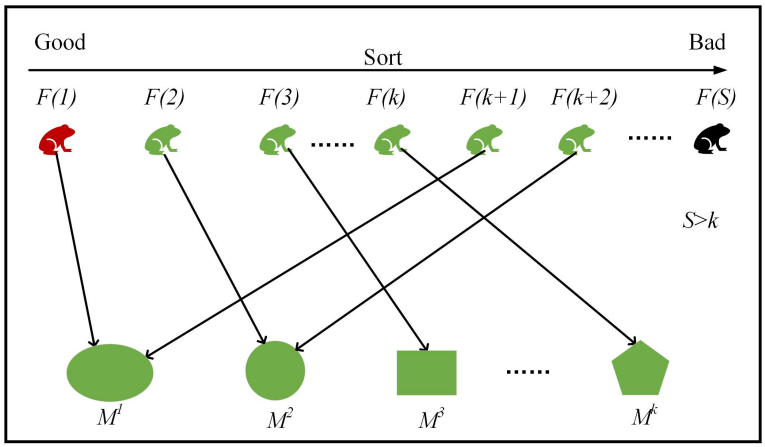


Figure 4. An example diagram of frog assignment.

Next, each frog in M^j performs a local search process, which contains three types of leap update mechanisms (LUM). The local search processes are performed during each round of the global search process, where a local search is performed for each M^j to determine the local optimal solution and update or eliminate the worst solution. Figure 4 illustrates a round of global iteration; when it finishes, the next step is exchanging the global information.

q $(q < n)$ frogs from M^j are selected according to the probability of $p_i = \frac{2(n+1-i)}{n(n+1)}$ to form a sub-memeplex SM^j, which is used for the local search. The smaller the $f(i)$ of a frog, the higher the probability of being selected into the SM^j. This selection process is randomized to ensure that the q frogs selected fully reflect the distribution of fitness values of frogs in that M^j. In the SM^j, the best frog is denoted as P_B, and the worst frog is $F(q)$, denoted as P_W.

Figure 5 shows an example of a global iteration. At each iteration round, the population S is updated to determine the optimal frog (red frog), M^j is a region containing n frogs, and q frogs are selected from the n frogs to form SM^j at each iteration time, and the frogs in SM^j and the other frogs in M^j are updated and exchanged after one iteration (green frog). The frogs in SM^j are sorted to get the best local frog (blue frog), the worst local frog (black frog) and the other frogs (orange frogs). The red, orange and blue dotted lines indicate the three types LUMs of P_W, as described in the following.

$$F_i'(q) = P_W + \lambda_i \tag{13}$$

$$\lambda_i = \begin{cases} \min\{\lceil R \times \triangle_i \rceil, \lambda_{max}\}, \text{positive updating} \\ \max\{\lceil R \times \triangle_i \rceil, -\lambda_{max}\}, \text{negative updating} \end{cases} \tag{14}$$

where $F_i'(q)(i = 1, 2, 3)$ is the updated solution of the worst solution PW under the three types of LUMs; R is a random array between $(0, 1)$ and $\lceil * \rceil$ means rounding up; $\lambda_i(i = 1, 2, 3)$ is the update factor among the three types LUMs; $\triangle_i(i = 1, 2, 3)$ is the update

methods of the three types LUMs; λ_{max} is the maximum update factor among the three types of LUMs, which controls the global search ability of the algorithm. The three types of LUMs are as follows.

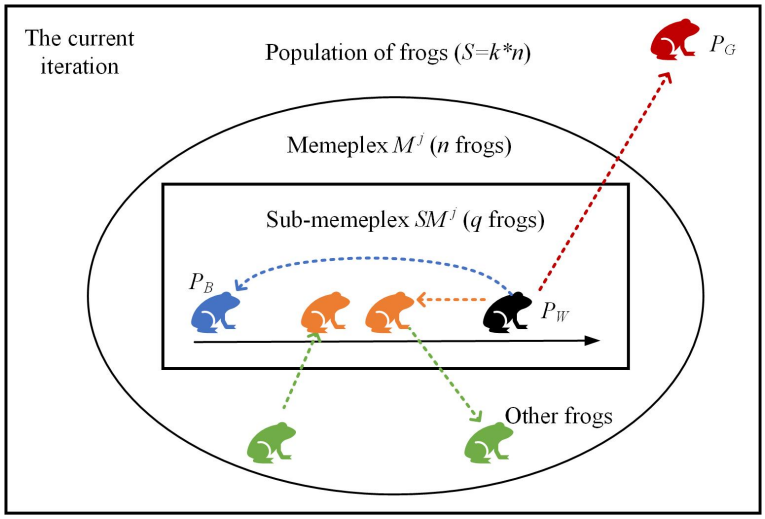


Figure 5. Schematic diagram of HSPP-HA.

The first type of LUM updates P_W according to the P_B in SM^j, the update method $\triangle_1$ is expressed by:

$$\triangle_1 = P_B - P_W \tag{15}$$

where $\triangle_1$ is the vector difference between P_B and P_W. $F_1'(q)$ is obtained by executing the first-type LUM. In this step, if the $F_1'(q)$ belongs to the feasible solution space, then the $f_1'(q)$ is calculated for this frog, and if the $f_1'(q)$ is better than $f(q)$, then the $F(q)$ is replaced by the $F_1'(q)$, otherwise, the second-type of LUM is performed.

In the traditional SFLA, if P_W cannot be improved after updating according to P_B, a randomly generated frog in the feasible solution space replaces the old poor one. The random generation eliminates the worst solution from the population and represents a fast convergence speed but it is not conducive to the evolution of the overall population in our task model. Therefore, an adaptive factor for generating a frog in each iteration is designed for the second type of LUM, expressed by:

$$\triangle_2 = \delta P_B - P_W, \delta = e^{-(\frac{t}{t_{max}}+r)} \tag{16}$$

where δ is the adaptive iteration factor, t is the number of current iterations; t_{max} is the max number of iterations; r is a random number between $(0, 1)$. The idea of such an approach is P_W should learn from P_B since they belong to the same M^j, and at the early stage of iterations, the learning behavior of the poor frog imitating the good frog should consider the convergence speed and global search capacity of the algorithm, while ensuring that it does not fall into a local optimum that a higher value of δ is preferable to speed up the learning and guarantee a large search space; and at the late stage of iterations, it should take into account the local search capacity of the algorithm for the accuracy of the solution. Therefore, δ should be a smaller value to slow the learning and ensure the local search capacity that eventually leads to balanced local and global searches.

Then, $F_2'(q)$ is obtained by the second-type LUM, and the same replacement process as the first-type LUM is performed; if there are no frogs that can be replaced, then the third-type LUM is performed, which is the global search. It updates P_W by the globally best frog P_G under the current iteration round; the update method $\triangle_3$ is expressed as follows:

$$\triangle_3 = P_G - P_W \tag{17}$$

where $\triangle_3$ is the vector difference between P_G and P_W. The third type of LUM is the last operation of P_W update, which ensures that the updated solution is not the worst solution of the S. By the three types LUMs, the worst frog $F(q)$ in SM^j is optimally updated, and the worst frog in M^j is eliminated. Meanwhile, the frogs in M^j are re-ordered incrementally according to $f(i)$ to determine the new M^j. The third type of LUMs loops until the search in each M^j has reached the maximum iteration number and all M^j have been searched, and finally returns the optimal path indicated by P_G.

The Algorithm 2 describes the details of the proposed HSPP-HA.

Algorithm 2: The overall process of HSPP-HA.

Input : N interaction points;
The k output regions M^j of Algorithm 1;
Maximum iteration rounds (T_{max});
Maximum iteration numbers (t_{max});
Output: Global optimal ASV path $(Path_{ASV})$;

Process I : **Global search**
Initialize the population S (n frogs, k M^j);
for $T = 1$ *to* T_{max} **do**
  Calculate $f(i)$ for the frog by Eqaution (4);
  Determine the P_G under T by sorting the frogs incrementally according to $f(i)$;
  Assign the frogs to M^j by Eqaution (12);
  Process II : **Local search**
  for $j = 1$ *to* k **do**
    for $t = 1$ *to* t_{max} **do**
      Random selection of q frogs to form SM^j;
      Determine the P_B and P_W in SM^j;
      Execute first-type LUM (Eqaution (15)), obtain $F_1^{'}(q)$, and calculate $f_1^{'}(q)$;
      if $f_1^{'}(q) < f(q)$ **then**
        Replace $F(q)$ with $F_1^{'}(q)$;
      else
        Execute second-type LUM (Eqaution (16)), obtain $F_2^{'}(q)$, and calculate $f_2^{'}(q)$;
        if $f_2^{'}(q) < f(q)$ **then**
          Replace $F(q)$ with $F_2^{'}(q)$;
        else
          Execute third-type LUM (Eqaution (17)), obtain $F_3^{'}(q)$ replacing $F(q)$ and calculate $f_3^{'}(q)$;
        end
      end
      $t = t + 1$;
    end
    $j = j + 1$;
  end
  Perform the global information exchange between M^j, update S, and record the P_G;
  $T = T + 1$;
end
Return $Path_{ASV}$;

5. Simulation and Analysis

In this section, the performance of the proposed HSPP-HA is analyzed through simulations by comparing with the SFLA [21], hierarchical multi-objective particle swarm optimization (H-MOPSO) [29] and artificial jellyfish search (JS) [39].

H-MOPSO is an improvement of PSO where priorities are given to evaluation factors for optimizing multiple objects. The optimal particles are stored in an archive after each iteration, and then the particles are updated by the local optimal particles and the global optimal particles in the archive, where the global optimal particles are selected from the archive by the Roulette mechanism and such an operation improves the diversity of the population to some extent. JS is a heuristic algorithm that emulates the food-fearching behavior of jellyfish, where jellyfishes switch to active motion by following ocean currents and to passive motion by following jellyfish swarms. The active and passive movements are controlled by a time control function, and the logistic chaotic mapping is used in initializing the jellyfish population.

Compared with the proposed HSPP-HA, they all aim to solve multi-objective optimization problems, but with different objectives. H-MOPSO improves the quality of local and global searches by enriching the population, while HSPP-HA balances local and global searches by adaptive convergence progress, JS applies the logistic chaotic mapping in initializing the population that decreases the randomness, while HSPP-HA uses the spatial-temporal clustering to adapt well to time-sensitive local search. The purpose of the simulation is to verify the adaptability of the proposed adaptive convergence approach and spatial-temporal clustering to the time-sensitive task model by comparing it with different heuristic algorithms with different convergence mechanisms.

The simulations are performed in a two-dimensional space of 500 $\times$ 500 m with a different number of interaction points generated by three AUGs at different times, and the evaluation metrics are the ASV's path length, access rate, data collection rate, path diagram and the convergence performance of algorithms. The operating parameters of the four methods are listed in Table 2, where the γ and β of JS are the motion coefficient of jellyfishes following swarms and the distribution coefficient of jellyfish swarms, respectively; the $C1$, $C2$, ω and S of H-MOPSO are the self-awareness learning factor, population cognitive learning factor, inertia factor and population size, respectively.

Table 2. Parameter setting table for the four algorithms.

Algorithms	Parameters	Values
JS	γ, β t_{max}, S	0.1, 3 300, 600
SFLA	k, n, q λ_{max}, T_{max}, t_{max}	10, 60, 25 3, 30, 300
H-MOPSO	C_1, C_2, ω t_{max}, S	1.6, 2, 0.9 300, 600
Proposed	k, n, q, r λ_{max}, T_{max}, t_{max} $\frac{BI}{D}$, x_{max}, y_{max}	10, 60, 25, 0.8 3, 30, 300 3, 500, 500

Figure 6 illustrates an example of interaction points generated by multiple AUGs. The dotted lines represent the top view of AUG trajectories, and the interaction points occur in numerical order in the figure, which disappears after a certain period. Each AUG carries a certain amount of data as it floats to the interaction point, and if the ASV misses an interaction, the AUG will buffer the data to the next interaction point. However, only up to three times because, in the case of more than three buffer instances, the buffered data exceeds the transferable amount for one interaction period, which causes data loss. The goal is to plan an optimized path for ASV on the sea surface as a way to sequentially access the interaction points while collecting as much data as possible.

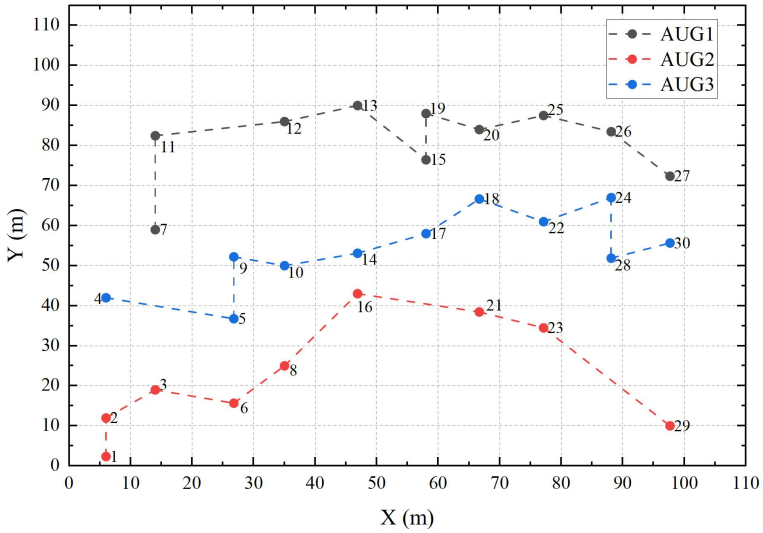


Figure 6. An example diagram of the interaction points generated by three AUGs.

Figures 7–9 demonstrate histograms of the path lengths, access rate and data collection rate at different numbers of interaction points. The access rate is the rate at which ASV accesses all the interaction points before they disappear, and the data collection rate indicates the rate of data collected from all AUGs.

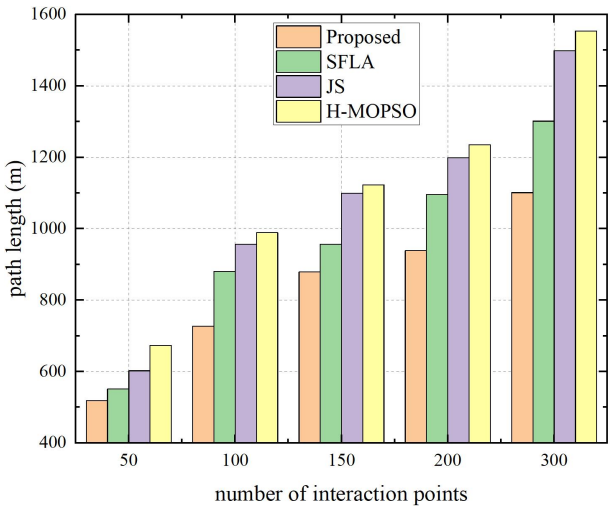


Figure 7. Comparison of path length under different numbers of interaction points.

In terms of path length (Figure 7) and access rate (Figure 8), the proposed HSPP-HA shows the highest performances, as expected, and SFLA follows, which demonstrates the adaptability of spatial-temporal clustering to our task model. H-MOPSO and JS show lower performance because the temporal and spatial properties of each interaction point are not correlated, thus confusing the path-planning decisions. Such a case can be explained by the case of 50 interaction points in Figure 8 because a lower density of interaction points results in a longer distance between them, so the paths planned between interaction points with similar times of occurrence but longer distances lead to lower global access rates. The proposed method outperforms SFLA because of the adaptive iteration approach. Specifically in terms of data collection rate (Figure 9), since the adaptive iteration factor well balances the global and local search, it shows significant improvements compared to the original SFLA, where the data collection rate has increased by more than 20% in the case of 300 interaction points and reached 98% in the case of 50 interaction points. H-MOPSO and JS performed poorly in terms of data collection rate and even shows values similar to the access rate in cases of 200 and 300 interaction points. This is because, in the case of

high-density interaction points occurring at different times, the data missed but buffered to the next interaction point are not collected in time.

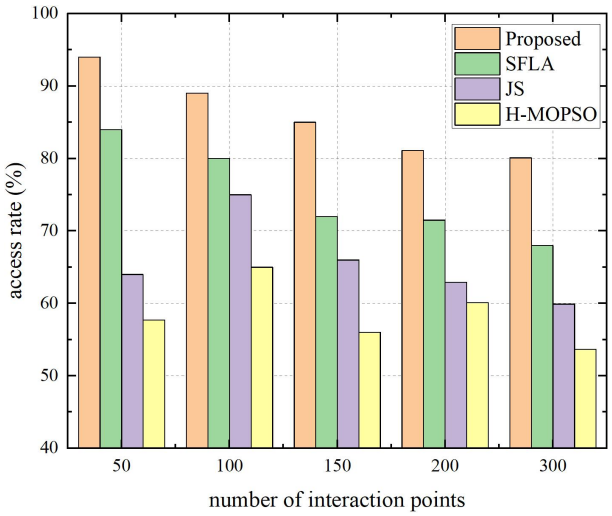


Figure 8. Comparison of the access rate under different numbers of interaction points.

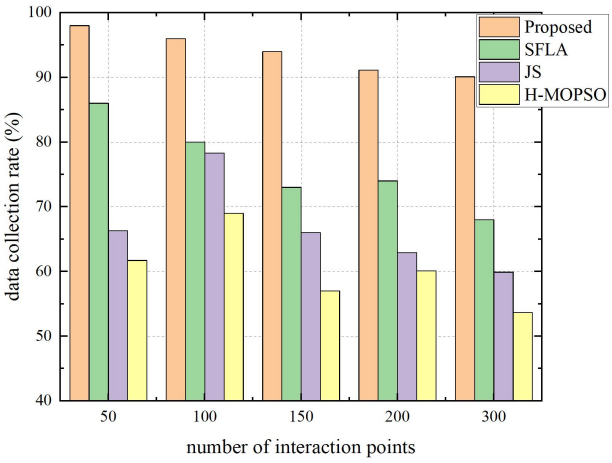
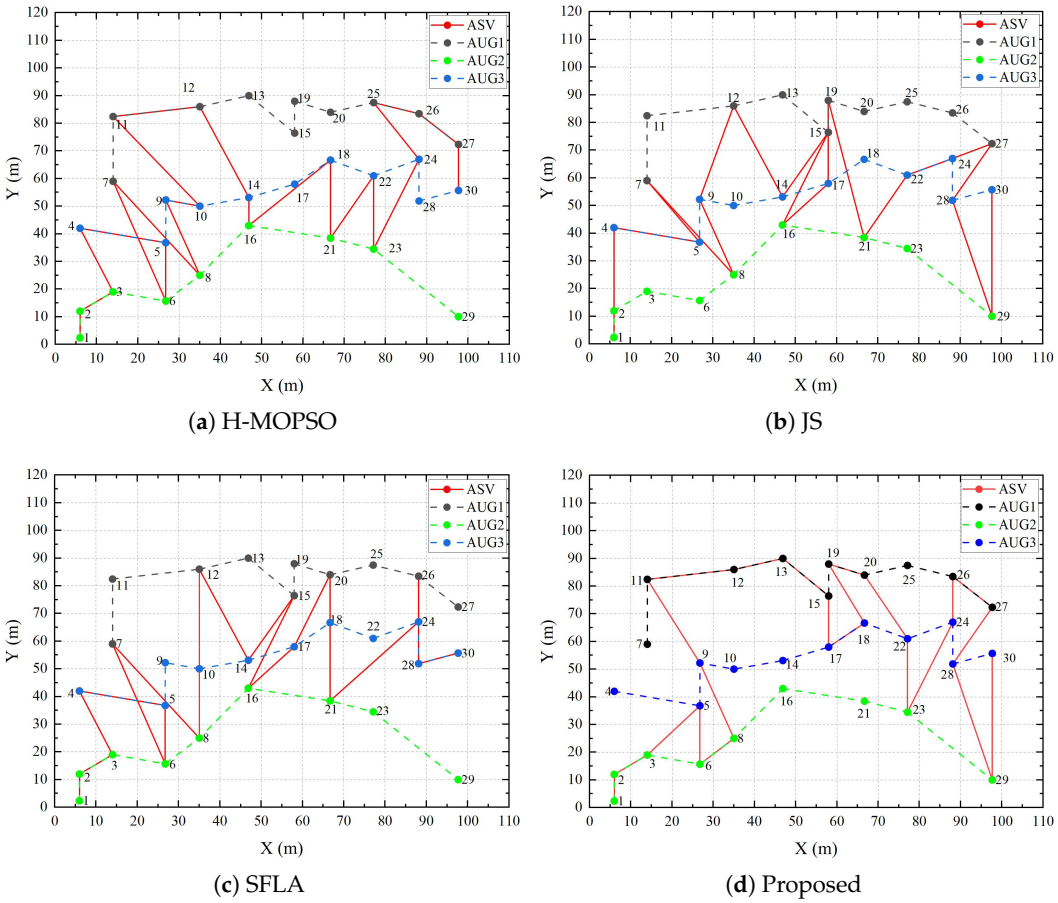


Figure 9. Comparison of the data collection rate under different numbers of interaction points.

In order to further analyze the detailed processes, the planned paths with four algorithms are compared in Figure 10, where three AUGs work in a 100 × 100 m region and generate 30 interaction points.

In Figure 10, the three dotted lines indicate the top view of the trajectories of three AUGs, and the solid lines indicate the paths of ASV. Figure 10a shows the path under H-MOPSO, where ASV successfully accessed 23 interaction points with an access rate of 76.67%. In the case of interaction points 13, 15, 19 and 20 from AUG1, the data from interaction points 13, 15 and 19 are not collected because they do not meet the C_i^q where $q_i = 3$; therefore, the ASV only collected the data buffered from 20 at interaction point 25. In the case of interaction point 29, the path between interaction points 27 and 29 does not meet the $C_{i \to j}^v$, so the ASV gives up accessing interaction point 29 and directly accesses interaction point 30 so that the data from interaction point 29 are lost. At this time, the data collection rate is 86.67%, and the path length is 547.8 m. Figure 10b shows the path under JS, where ASV successfully accessed 20 interaction points with an access rate of 66.67%, and the data on interaction points 20 and 25 are not successfully collected for the same reason. The data collection rate and path length are 93.33% and 501.3 m, respectively, which shows improved performances compared to the H-MOPSO, albeit with a lower access rate.



(**a**) H-MOPSO (**b**) JS

(**c**) SFLA (**d**) Proposed

Figure 10. ASV path diagram for four algorithms.

SFLA in Figure 10c results in a 70% access rate, 93.33% data collection rate and 436.8 m path length, while the proposed method in Figure 10d results in a 73.33% access rate, 100% data collection rate and 411.6 m path length. They brought higher data collection rates because the spatial-temporal clustering is adopted and each path is scheduled according to the occurrence time of the interaction points.

Table 3 summarizes the performance of the four algorithms in terms of path length, access rate and data collection rate for different numbers of interaction points. The time complexity of the four algorithms is also listed in the table. JS, SFLA and HSPP-HA are $O(\sum N_i) = O(N)$ where N is the population size, and i is the sequence of iterations. H-MOPSO is $O(N^2)$ because it additionally searches the particles in an archive in each iteration.

Finally, the convergence ability of the four algorithms is compared with the optimization function f (Equation (4)). Figure 11 compares the result where the number of interaction points is 30. The optimal value of the proposed method shows the fastest convergence, starting to converge at about the 80th iteration, while the H-MOPSO algorithm starts to converge at about the 200th iteration, and the proposed method results in a better optimal value after convergence, which demonstrates the ability of the proposed adaptive iteration approach to avoid the local optimum. The convergence ability of HSPP-HA and SFLA is better than JS, which indicates that the three types of LUMs mechanisms have higher adaptability to our task model since the worst solutions are removed from the population in each iteration and the time control function of JS may lead to the unbalanced local and global search; and due to the spatial-temporal clustering in initializing the population, HSPP-HA shows better convergence than SFLA, as expected. Regarding the H-MOPSO, it shows the worst performance because the PSO mechanism may lead to local optimum problems, especially in our task model where time-sensitive objectives constitute a complex optimization environment, while HSPP-HA uses the adaptive convergence factor in each

iteration that makes the algorithm have a strong global search capability in the early stage that avoids the local optimum and leads to a strong local search capability in the later stage that further improves the quality of the final solution.

Table 3. Statistical table for the four algorithms.

Algorithms	Number of Interaction Points	Path Length (m)	Access Rate (%)	Data Collection Rate (%)	Time Complexity
H-MOPSO	30	547.80	76.67	86.67	$O(N^2)$
	100	989.35	65.00	69.00	
	200	1234.68	60.00	60.00	
JS	30	501.30	66.67	93.33	$O(N)$
	100	956.49	75.00	78.00	
	200	1199.16	63.00	62.00	
SFLA	30	436.80	70.00	93.33	$O(N)$
	100	880.41	80.00	80.00	
	200	1096.13	71.00	74.00	
Proposed	30	411.60	73.33	100.00	$O(N)$
	100	727.58	89.00	96.00	
	200	939.23	82.00	90.00	

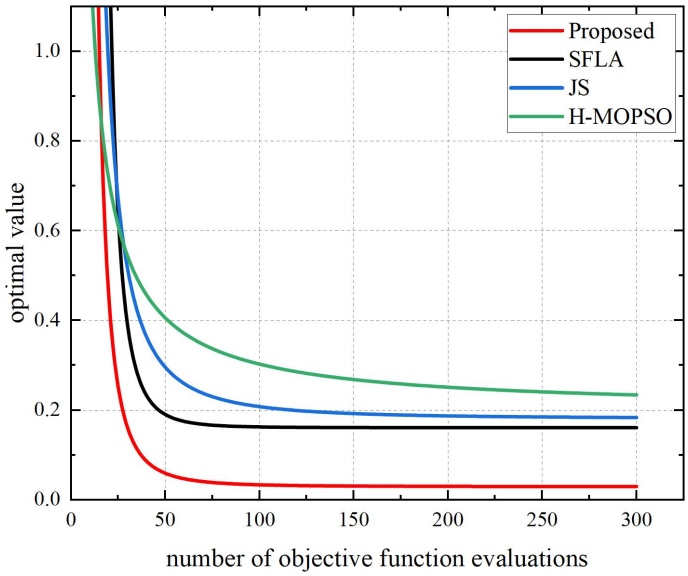


Figure 11. Convergence diagram of the four algorithms.

6. Conclusions

AMV collaboration is one of the fundamental technologies for practicing IoUT applications. Accordingly, this paper presents a heuristic surface path planning method for interacting heterogeneous AMVs, named HSPP-HA, which is designed for an application scenario of ASV on the sea surface, accessing a series of time-sensitive interaction points generated by AUGs underwater. The HSPP-HA is based on SFLA with some improvements for the application scenario, which are spatial-temporal clustering and an adaptive iteration factor. The spatial-temporal clustering assigns interaction points of AMVs that are close in time and location to the same access group for initializing a SFLA memeplex, which has the advantage in scheduling time-sensitive interactions; the adaptive iteration factor adjusts the convergence speed of SFLA in different iteration stages that balance the local and global searches of the heuristic convergences that avoids the local optimum problem to a certain extent. The simulations show that the proposed method has superiorities in terms of generated path length, access rate of the time-sensitive interaction points, data collection

rate and convergence performance compared to the classic and recent approaches. However, the proposed algorithm still has some space for improvement because the objective optimization function does not take into account the motion constraint of the ASV, which will be studied in our future work, and our future work will also focus on designing a subsea path planning method for underwater AMVs that should take into account complex underwater geography, irregular currents and dynamic obstacles.

Author Contributions: Conceptualization, J.Z. and Z.W.; methodology, J.Z. and Z.W.; validation, G.H. and Y.Q.; formal analysis, J.Z. and G.H.; investigation, Z.W.; writing—original draft preparation, J.Z. and Z.W.; writing—review and editing, J.Z., G.H. and Y.Q.; visualization, J.Z. and Z.W.; supervision, G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Fundamental Research Funds for the Central Universities, No. B220202021, in part by the National Natural Science Foundation of China under Grant No. 62072072, No. 61971206, No. 62002099 and in part by the Changzhou Foundation of Science and Technology No. CJ20210144.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Oceanix. Available online: https://oceanix.com/ (accessed on 1 January 2023).
2. Tao, S.; Liu, H.; Wang, S.; Li, C. Construction of Smart Coastal Cities Based on Digital Government. *J. Coast. Res.* **2020**, *110*, 154–158. [CrossRef]
3. Atham, S.B.; Guleria, K. Smart City in Underwater Wireless Sensor Networks. In *Energy-Efficient Underwater Wireless Communications and Networking*; IGI Global: Hershey, PA, USA, 2021; pp. 287–301.
4. McMahon, J.; Plaku, E. Autonomous Data Collection With Dynamic Goals and Communication Constraints for Marine Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2022**, 1–14. [CrossRef]
5. Han, G.; Qi, X.; Peng, Y.; Lin, C.; Zhang, Y.; Lu, Q. Early Warning Obstacle Avoidance-Enabled Path Planning for Multi-AUV-Based Maritime Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]
6. Han, G.; Gong, A.; Wang, H.; Martínez-García, M.; Peng, Y. Multi-AUV Collaborative Data Collection Algorithm Based on Q-Learning in Underwater Acoustic Sensor Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9294–9305. [CrossRef]
7. Chu, Z.; Wanf, F.; Lei, T.; Luo, C. Path Planning based on Deep Reinforcement Learning for Autonomous Underwater Vehicles under Ocean Current Disturbance. *IEEE Trans. Intell. Veh.* **2022**, *8*, 108–120. [CrossRef]
8. Yang, J.; Huo, J.; Xi, M.; He, J.; Li, Z.; Song, H. A Time-saving Path Planning Scheme for Autonomous Underwater Vehicles with Complex Underwater Conditions. *IEEE Internet Things J.* **2023**, *10*, 1001–1013. [CrossRef]
9. Zhang, J.; Sha, J.; Han, G.; Liu, J.; Qian, Y. A Cooperative-Control-Based Underwater Target Escorting Mechanism With Multiple Autonomous Underwater Vehicles for Underwater Internet of Things. *IEEE Internet Things J.* **2022**, *8*, 4403–4416. [CrossRef]
10. Yu, F.; Chen, Y. Cyl-iRRT*: Homotopy Optimal 3D Path Planning for AUVs by Biasing the Sampling into a Cylindrical Informed Subset. *IEEE Trans. Ind. Electron.* **2022**. [CrossRef]
11. Wen, J.; Yang, J.; Li, Y.; He, J.; Li, Z.; Song, H. Behavior-Based Formation Control Digital Twin for Multi-AUG in Edge Computing. *IEEE Trans. Netw. Sci. Eng.* **2022**. [CrossRef]
12. Wen, J.; Yang, J.; Wei, W.; Lv, Z. Intelligent Multi-AUG Ocean Data Collection Scheme in Maritime Wireless Communication Network. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 3067–3079. [CrossRef]
13. Hu, W.; Chen, F.; Xiang, L.; Chen, G. Multi-ASV Coordinated Tracking With Unknown Dynamics and Input Underactuation via Model-Reference Reinforcement Learning Control. *IEEE Trans. Cybern.* **2022**, 1–10. [CrossRef] [PubMed]
14. Jeong, M.; Lee, E.; Lee, M. An Adaptive Route Plan Technique with Risk Contour for Autonomous Navigation of Surface Vehicles. In Proceedings of the OCEANS MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–4.
15. Dubey, R.; Louis, S. VORRT-COLREGs: A Hybrid Velocity Obstacles and RRT Based COLREGs-Compliant Path Planner for Autonomous Surface Vessels. In Proceedings of the OCEANS 2021: San Diego—Porto, San Diego, CA, USA, 20–23 September 2021; pp. 1–8.
16. Vagale, A.; Bye, R.; Osen, O. Evaluation of Path Planning Algorithms of Autonomous Surface Vehicles Based on Safety and Collision Risk Assessment. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020; pp. 1–8.
17. Yu, Z.; Si, Z.; Li, X.; Wang, D.; Song, H. A Novel Hybrid Particle Swarm Optimization Algorithm for Path Planning of UAVs. *IEEE Internet Things J.* **2022**, *9*, 22547–22558. [CrossRef]

18. Qadir, Z.; Zafar, M.; Moosavi, S.; Le, K.; Mahmud, M. Autonomous UAV Path-Planning Optimization Using Metheuristic Approach for Predisaster Assessment. *IEEE Internet Things J.* **2022**, *9*, 12505–12514. [CrossRef]
19. Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [CrossRef]
20. Kaveh, A.; Dadras Eslamlou, A. Water strider algorithm: A new metaheuristic and applications. *Structures* **2020**, *25*, 520–541. [CrossRef]
21. Muzaffar, E.; Kevin, L.; Fayzul, P. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Eng. Optim.* **2006**, *38*, 129–154.
22. Yang, J.; Ni, J.; Xi, M.; Wen, J.; Li, Y. Intelligent Path Planning of Underwater Robot Based on Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* **2022**. [CrossRef]
23. Abbasi, A.; MahmoudZadeh, S.; Yazdani, A. A Cooperative Dynamic Task Assignment Framework for COTSBot AUVs. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1163–1179. [CrossRef]
24. Scott, D.; Manyam, S.; Casbeer, D.; Kumar, M. A Lagrangian Algorithm for Multiple Depot Traveling Salesman Problem With Revisit Period Constraints. *IEEE Trans. Autom. Sci. Eng.* **2022**. [CrossRef]
25. Chen, M.; Zhu, D. A Workload Balanced Algorithm for Task Assignment and Path Planning of Inhomogeneous Autonomous Underwater Vehicle System. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 483–493. [CrossRef]
26. Zhu, D.; Zhou, B.; Yang, S. A Novel Algorithm of Multi-AUVs Task Assignment and Path Planning Based on Biologically Inspired Neural Network Map. *IEEE Trans. Intell. Veh.* **2021**, *6*, 333–342. [CrossRef]
27. Wu, J.; Song, C.; Ma, J.; Wu, J.; Han, G. Reinforcement Learning and Particle Swarm Optimization Supporting Real-Time Rescue Assignments for Multiple Autonomous Underwater Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6807–6820. [CrossRef]
28. Wang, N.; Xu, H. Dynamics-Constrained Global-Local Hybrid Path Planning of an Autonomous Surface Vehicle. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6928–6942. [CrossRef]
29. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyam, Z.; Raeburn, C.; Salter, I.; Pekcan, C. A Multiobjective Optimization Approach for COLREGs-Compliant Path Planning of Autonomous Surface Vehicles Verified on Networked Bridge Simulators. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1167–1179. [CrossRef]
30. Jian, S.; Hsieh, S. A Niching Regression Adaptive Memetic Algorithm for Multimodal Optimization of the Euclidean Traveling Salesman Problem. *IEEE TRansactions Evol. Comput.* **2022**. [CrossRef]
31. Xin, C.; Kang, W.; Yi, M.; Zheng, L.; Jun, Z.; Qing, Z. Decomposition-based Lin-Kernighan Heuristic with Neighborhood Structure Transfer for Multi/Many-objective Traveling Salesman Problem. *IEEE Trans. Evol. Comput.* **2022**. [CrossRef]
32. Zhang, Z.; Liu, H.; Zhou, M.; Wang, J. Solving Dynamic Traveling Salesman Problems With Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [CrossRef] [PubMed]
33. Sanyal, S.; Roy, K. Neuro-Ising: Accelerating Large-Scale Traveling Salesman Problems via Graph Neural Network Guided Localized Ising Solvers. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 5408–5420. [CrossRef]
34. Mei, H.; Wang, H.; Shen, X.; Bai, W. An Adaptive MAC Protocol for Underwater Acoustic Sensor Networks With Dynamic Traffic. In Proceedings of the OCEANS MTS/IEEE Charleston of the Conference, Charleston, SC, USA, 22–25 October 2018; pp. 1–4.
35. Qiao, G.; Zhao, Y.; Liu, S.; Ahmed, N. The Effect of Acoustic-Shell Coupling on Near-End Self-Interference Signal of In-Band Full-Duplex Underwater Acoustic Communication Modem. In Proceedings of the 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST) of the Conference, Natl Ctr Phys, Islamabad, Pakistan, 14–18 January 2020; pp. 606–610.
36. Eghbal, M.; Saha, T.; Hasan, K. Transmission expansion planning by meta-heuristic techniques: A comparison of Shuffled Frog Leaping Algorithm, PSO and GA. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting of the Conference, Detroit, MI, USA, 24–28 July 2011; pp. 1–8.
37. Duarte, B.; de Oliveira, L.; Teixeira, M.; Barbosa, M. A comparison of Genetic and Memetic Algorithms applied to the Traveling Salesman Problem with Draft Limits. In Proceedings of the 47th Latin American Computing Conference (CLEI) of the Conference, Cartago, Costa Rica, 25–29 October 2021.
38. Huang, M.; Zhang, K.; Zeng, Z.; Wang, T.; Liu, Y. An AUV-Assisted Data Gathering Scheme Based on Clustering and Matrix Completion for Smart Ocean. *IEEE Internet Things J.* **2020**, *7*, 9904–9918. [CrossRef]
39. Jui, C.; Dinh, T. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *289*, 125535.