

Article

An Improved Reeds–Shepp and Distributed Auction Algorithm for Task Allocation in Multi-AUV System with Both Specific Positional and Directional Requirements

Hongfei Li ¹, Daqi Zhu ^{1,*} , Mingzhi Chen ^{1,*} , Tong Wang ¹ and Hongxiu Zhu ²

¹ School of Mechanical Engineering, The University of Shanghai for Science and Technology, 516 Jungong Road, Shanghai 200093, China; jshmlihongfei@163.com (H.L.); zdq367@aliyun.com (D.Z.); 2010360@tongji.edu.cn (T.W.)

² Shanghai Marine Equipment Research Institute, Shanghai 200031, China; m13122779196@163.com

* Correspondence: mingzhichen2008@163.com

Abstract: Task assignment is of paramount importance in multi-AUV systems, particularly in applications such as bridge inspection where task execution is direction-specific. In such scenarios, the underactuation of AUVs is a critical factor that cannot be ignored. Therefore, it is essential to consider the AUV's kinematic model comprehensively to ensure minimal energy consumption during task execution. In this paper, we introduce an improved Reeds–Shepp algorithm in conjunction with a distributed auction approach. We treat AUVs as car-like models in our approach, paying meticulous attention to their operational characteristics during path planning. Importantly, we effectively utilize their backward driving capabilities. Our analysis reveals that this model successfully fulfills the directional requirements of detection tasks. Furthermore, the distributed auction approach optimizes the overall task distribution in the multi-AUV system. We support our method with simulation results that underscore its effectiveness.

Keywords: underactuated AUV; task allocation; improved Reeds–Shepp; distributed auction



Citation: Li, H.; Zhu, D.; Chen, M.; Wang, T.; Zhu, H. An Improved Reeds–Shepp and Distributed Auction Algorithm for Task Allocation in Multi-AUV System with Both Specific Positional and Directional Requirements. *J. Mar. Sci. Eng.* **2024**, *12*, 486. <https://doi.org/10.3390/jmse12030486>

Academic Editor: Rafael Morales

Received: 23 February 2024

Revised: 8 March 2024

Accepted: 12 March 2024

Published: 14 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bridges standing over bodies of water serve as critical infrastructures. They are, however, exposed to numerous environmental challenges such as river washes, sediment deposits, corrosion due to polluted water, freezing damages, melting ice, and impacts from boats and other floating objects. These factors can induce defects, reduce structural bearing capacity and durability, and ultimately compromise the operational safety of a bridge. Therefore, regular inspections and evaluations are indispensable to ensure the normal use of these structures and prevent safety incidents [1]. To enhance inspection efficiency and minimize labor costs, the application of unmanned underwater vehicles has garnered considerable attention. As such, the deployment of multiple autonomous underwater vehicles (multi-AUVs) has become increasingly significant, providing an effective solution for executing complex inspection missions [2]. A multi-AUV system is a collective of AUVs specifically engineered to perform coordinated actions, enabling the accomplishment of objectives that a single AUV cannot achieve [3]. This system offers numerous potential benefits, including an elevated capacity for handling complex tasks, improved overall performance, and increased reliability [4]. Through the use of multi-AUV systems, bridge inspections can be conducted more efficiently. However, optimizing task allocation to enhance inspection performance remains one of the most daunting challenges within a multi-AUV system [5]. This task assignment problem becomes even more complex considering that AUVs must not only reach specific locations but also orient themselves in specific directions at those positions [6]. Task allocation plays a pivotal role in the bridge inspection of a multi-AUV system. It serves as the foundational aspect of multi-AUV

systems, reflecting the organizational structure and operational mechanisms at the system's decision-making level [7]. The efficiency of task allocation has a direct impact on task completion and is closely tied to whether an AUV can fully utilize its capabilities [8]. As a result, task allocation is both crucial and challenging. In essence, the task assignment problem is classified as an NP-hard problem [7]. In recent years, numerous scholars have introduced a variety of algorithms aimed at addressing this issue, including market-based mechanisms, intelligent optimization methods, neural network algorithms, etc.

Intelligent optimization algorithms are engineered to address allocation challenges by identifying the optimal solution from a set of possible choices [9]. These algorithms encompass particle swarm optimization (PSO), ant colony optimization (ACO), and their various derivatives [10]. Such methods have gained widespread application in resolving task allocation issues in multi-AUV systems. ACO, in particular, as a renowned heuristic optimization technique, has proven effective in solving complex problems like the traveling salesman problem [11]. Traditional implementations of ACO and PSO algorithms, however, tend to encounter difficulties with becoming ensnared in local optima when applied to task allocation challenges. To overcome this, Wang et al. introduced a hybrid approach that integrates these two algorithms, employing ACO for the initial task distribution and then refining the allocation with PSO to minimize resource consumption. Despite its innovation, this method does not fully account for real-time constraints, which can lead to inefficiencies in multi-AUV systems during certain inspections. To mitigate these issues, Wu et al. put forward the R-RLPSO (real-time reinforcement learning PSO) algorithm [12], which merges reinforcement learning with PSO to enhance the real-time computational capabilities of the algorithm. This approach has been tested in 3D simulation environments and has demonstrated high efficiency and speed. While the aforementioned algorithm is adept at solving task allocation problems, it typically employs a centralized strategy. This can be problematic in underwater communication, which, unlike terrestrial communication, faces challenges such as limited scalability and poor fault tolerance [13].

To mitigate the challenges associated with centralized decision making in multi-AUV systems, the self-organizing map (SOM) algorithm has been introduced as a distributed solution. In this approach, each AUV independently selects tasks based on its own sensory input, enhancing the system's robustness by reducing the dependency on a central command structure. The SOM algorithm was initially applied to solve multitasking problems in mobile robots [14] and later adapted by Zhu et al. for dynamic task assignment and path planning in multi-AUV systems [15]. Zhu et al. refined the traditional SOM by modifying the winner selection function to strike a balance between energy conservation and workload distribution. This modification ensured that AUVs were assigned only to tasks that they had the highest likelihood of completing successfully. They also addressed the challenges of obstacle avoidance and abrupt speed changes during task execution by incorporating the Glasius bio-inspired neural network (GBNN) into the AUV control systems [16]. This integration enabled AUVs to navigate underwater environments safely, avoiding collisions with obstacles. However, while the SOM algorithm tends to focus on finding the best local solution, achieving a globally optimal task allocation is often critical in multi-AUV systems. For this purpose, the auction algorithm is considered more appropriate for global optimization as it is designed to determine the most efficient distribution of tasks across the entire fleet of AUVs.

Auction algorithms have earned recognition for their exceptional effectiveness, with the concept originating from Bertsekas's work at MIT in 1979, inspired by the principles of economic auctions [17]. These algorithms boast advantages such as low computational demands and high efficiency [18]. Initially, auction algorithms were centralized, depending on a common data repository. However, the logistical and financial burdens of setting up such centralized systems underwater have spurred the development of distributed auction algorithms. Zavlanos et al. addressed the challenge by merging the auction algorithm with the consensus algorithm, thus eliminating the need for centralized storage and achieving a fully distributed algorithm [19]. Considering the prohibitive costs of centralized control in

marine settings, a distributed architecture is more practical for multi-AUV systems [20]. Building on this concept, Lee et al. introduced a resource-constrained decentralized auction algorithm tailored for multi-agent systems. This algorithm facilitates task assignment through inter-agent communication, with tasks allocated to the agent that can be completed in the shortest time [21]. Further refining the approach, Cheng et al. developed a mathematical model that incorporates dynamic constraints within multi-UAV systems, allowing the allocation problem to be resolved by the computation of bidding prices [22]. The performance of multi-AUV systems is significantly influenced by disturbances and the kinematic models of the AUVs. Consequently, there is a critical need for a robust task allocation auction algorithm that can function effectively in complex underwater scenarios. While the task reward feedback mechanism introduced by Bin et al. [23] and Li et al. [13] has proven to be effective in reducing costs and facilitating dynamic task allocation under intricate underwater conditions, it does not consider the effects of the AUVs' kinematic models. Addressing this gap is essential for developing an auction algorithm that is not only cost-effective but also kinematically aware, ensuring efficient task allocation in the face of the dynamic and often unpredictable underwater environment.

Among the numerous algorithms used to solve the path planning problem for AUVs, A* (A star) and the genetic algorithm (GA) are widely used. The A* algorithm is a heuristic search algorithm that uses a heuristic function to estimate the minimum cost from the current node to the target node. This can effectively guide the search direction and reduce the complexity of the search [24]. The traditional A* algorithm, however, cannot perform functions such as obstacle avoidance. Singh et al. [25] improved the A* algorithm by considering the effects of ocean currents and obstacles, planning a globally suitable and safe path. However, they did not consider the kinematic constraints of the AUV and the angle requirements for task execution. While the fully hierarchical multi-objective genetic algorithm (FHMOGA) and the Pareto optimal path generation algorithm in random traffic networks perform well in many aspects, they have some limitations in handling the kinematic constraints of AUVs and the angle requirements for actual task executions. Furthermore, due to their global search characteristics, the FHMOGA may not be able to adapt to dynamically changing environments and task requirements during actual task executions. This algorithm may also require complex communication and coordination mechanisms and may be affected by the diversity and quality of the solutions [26].

The underactuated nature of AUVs necessitates a comprehensive integration of their kinematic models into task allocation strategies. Recognizing the importance of directional constraints, Beverley et al. developed a path planning approach that respects the AUVs' orientation limitations by utilizing the Dubins path algorithm and the tactical placement of waypoints [27]. This method effectively leverages the AUVs' ability to move forward but does not account for their potential to navigate in reverse. Building upon the foundation laid by the Dubins path algorithm, Reed et al. introduced the Reeds–Shepp algorithm, an advancement that encompasses a broader range of curve planning [28]. The most significant enhancement offered by the Reeds–Shepp algorithm is the inclusion of reverse driving capabilities in its calculations. This algorithm allows for the planning of efficient paths that are more aligned with the operational characteristics of AUVs, ensuring that both forward and backward movements are considered in the design of their trajectories.

In this paper, we address the task allocation problem for multi-AUV systems by introducing an enhanced Reeds–Shepp algorithm tailored to satisfy the directional demands of multi-AUV operations. Our algorithm acknowledges the underactuation of AUVs and optimizes their movement capabilities, enabling both forward and reverse motions to refine the path planning process. Consequently, this results in a superior path optimization for the entire fleet. Task allocation is efficiently executed through a distributed auction algorithm, which precludes idle AUV states during inspection tasks, thus expediting detection and minimizing the overall system expenditure.

Moreover, our approach incorporates constraints into the AUV path planning to reflect real-world operational scenarios, such as bridge inspections, ensuring reliable AUV

navigation and successful mission completion. To demonstrate the efficacy of our proposed algorithm, we conduct simulations and comparative experiments, pitting the improved Reeds–Shepp approach against traditional Reeds–Shepp and Dubins methods. The paper’s key contributions are twofold:

(1) We account for the underactuation of AUVs and establish a car-like model by leveraging the enhanced Reeds–Shepp algorithm. This model allows the AUVs to meet the stringent directional requirements of inspection tasks with precision. Notably, it effectively utilizes the AUVs’ ability to move in reverse within the execution path, thereby optimizing task performance.

(2) We employ a distributed auction algorithm specifically designed to minimize task execution loss within the system. This approach significantly boosts overall efficiency, ensuring that the AUVs perform their tasks optimally.

The rest of this paper is organized as follows. The problem is formulated and described in Section 2. Section 3 presents the proposed algorithm, which contains the distributed auction in Section 3.1 and improved Reed–Shepp in Section 3.2. Section 4 demonstrates the simulation studies and discussion, which includes a comparison of the allocation algorithms and a comparison of the path planning algorithms. And last, Section 5 draws the conclusion.

2. Model Establishment

In bridge inspection projects, it is imperative for AUVs to approach the inspection points at specific angles, as depicted in Figure 1. Merely ensuring the shortest path to the task site is insufficient; the kinematic model of the AUV must also be considered to enable task execution at the appropriate angle. The blue arrow represents the AUV’s starting angle, while the red curve in Figure 1 indicates a potential path the AUV might take to perform a task. If the task cannot be executed at the required angle, certain inspection objectives may remain unfulfilled, underscoring the importance of the AUV’s execution angle.



Figure 1. Multi-AUV system performs the bridge detection.

Figure 2 showcases an AUV equipped with a BLUEVIEW sonar (M900) device (TELEDYNE MARINE, Daytona Beach, FL, USA), optimized for conducting inspections of bridge piers from specific directions. This configuration complicates the task of inspecting at other orientations. As illustrated in Figure 2a, tasks executed at incorrect angles could hinder the normal survey process. Practical experiments have confirmed that the AUV’s directional alignment upon reaching the target location is crucial as it has a direct impact on the success of the mission.

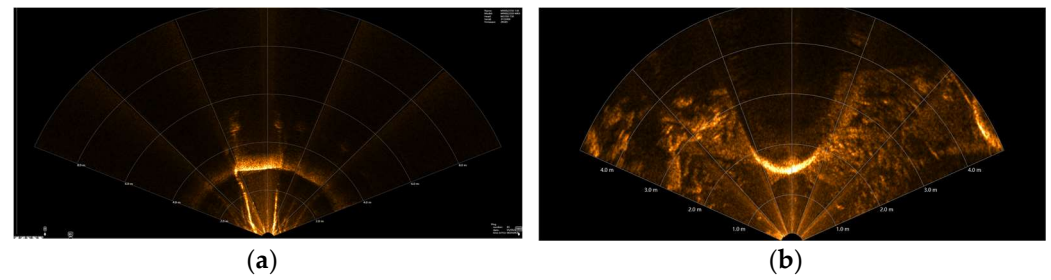


Figure 2. The AUV is equipped with BLUEVIEW sonar (M900) to detect. (a) At a wrong angle. (b) At a right angle.

This article focuses on the multi-AUV performing bridge inspection, considering the underactuated characteristic of the AUV, and inputs a series of tasks $T = \{T_1, T_2, \dots, T_n\}$ and homogeneous AUVs $A = \{R_1, R_2, \dots, R_k\}$ randomly set. This problem can be formulated as (1)

$$\underset{i \in R, j \in f}{\operatorname{argmin}} \sum_{i=1}^k \sum_{j=1}^n D_{(\xi_s, \xi, \xi_e)} * \operatorname{result}_{i,j} \quad (1)$$

s.t.

$$\operatorname{result}_{i,j} = \begin{cases} 1, & \text{assign AUV}_i \text{ to perform task } T_j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\sum_{i=1}^k \operatorname{result}_{i,j} = 1, \forall j$$

$$\begin{aligned} \xi_s &= [x_s \ y_s \ \phi_s]^T \\ \xi_e &= U([x_e \ y_e \ \phi_e]^T, [\delta_x \ \delta_y \ \delta_\phi]^T) \\ \begin{bmatrix} x \\ y \\ \phi \end{bmatrix} &= \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \\ r &\in [-r_a, r_a] \end{aligned} \quad (3)$$

Equation (1) is the objective function based on the shortest distance, where $D_{i,j}$ represents the i th AUV's distance to the j th task. For the distance cost, the shorter the total distance the AUVs can perform the tasks, the better. *result* means the executed order in Formula (2). To be clear, there are also limits to the AUVs required for each round of tasks, which is represented as $\sum_{i=1}^k \operatorname{result}_{i,j} = 1$, meaning the AUV can only execute one task or not execute any in a round of assignments.

Task allocation also necessitates the strategic planning of individual AUV routes to ensure an optimal and efficient task execution. In Figure 3a, the yellow icon represents an AUV that reaches its assigned task point. The green solid circle indicates a completed task, while the white hollow circle marks the location of a task that remains to be executed. However, this straightforward route planning, while theoretically sound, does not cater to the complexities of real-world scenarios, particularly within the context of the bridge inspection project discussed in this paper (as shown in Figure 1). Consequently, the path depicted in Figure 3b provides a more realistic and appropriate solution.

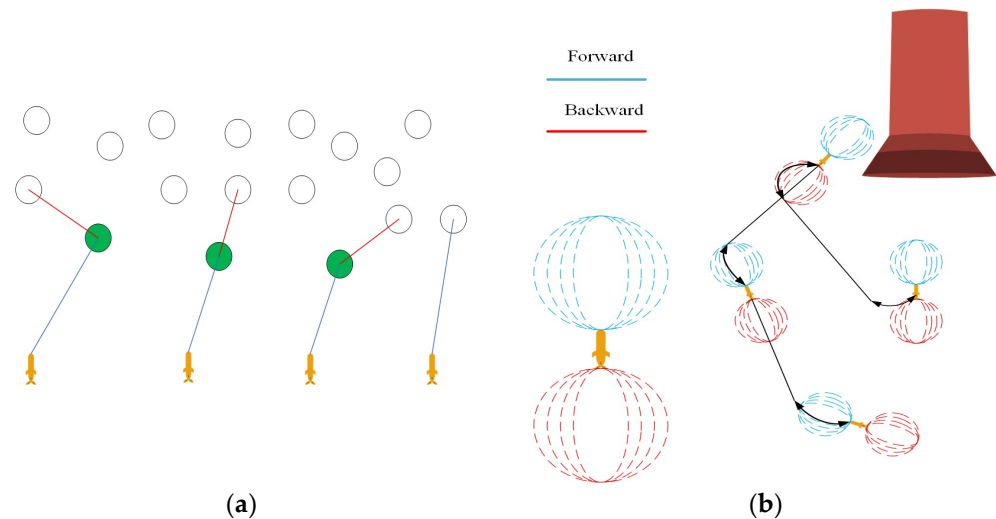


Figure 3. (a) Simple diagram of AUV execution task. (b) Path planning that meets orientation requirements.

The task allocation model for the multi-AUV system is subject to kinematic constraints and is characterized by an initial velocity vector. It also adheres to a minimum turning radius of r . Both the initial velocity vector and the final mission vector are pre-randomly predetermined. This problem can be formally expressed as Formula (3) [29]. $\zeta_s = [x_s \ y_s \ \phi_s]^T$ represents the AUV starting points with initial direction, and $\zeta_e = [x_e \ y_e \ \phi_e]^T$ represents the task point with the required angle. Due to the underdrive characteristic of the AUV, the turning radius of the AUV is also limited, denoted by $r \in [-r_a, r_a]$. In this kinematic model, it should be noted that since the AUV is underactuated, the side thrust is 0, then $v = 0$.

3. Mathematical Model Establishment and Post-Processing Methods

3.1. Distributed Auction Method on Task Allocation

To address the task allocation challenge within a multi-AUV system, this paper proposes a distributed auction algorithm, with the shortest distance serving as the objective function. Unlike centralized auction algorithms, the distributed auction method eliminates the need for a central unit or auctioneer. In the context of a multi-AUV system, the distance cost list for each AUV is not shared via a common memory unit. However, for more accurate task allocations, each AUV should be aware of the updated distance cost list. Consequently, each AUV updates its cost list in accordance with the shortest execution distance protocol. It is important to note though that the performance of the distributed auction algorithm may fluctuate based on the network topology. Information exchange typically occurs between the two closest nodes within the established network topology.

In the task allocation approach for the multi-AUV system based on the auction algorithm, three roles are essential: the bidders, the auctioneer, and the task allocation system. The latter serves as the auctioneer, disseminating information about each mission via the underwater communication network among AUVs. Upon task completion, each AUV is rewarded, optimizing the cost for the entire allocation system. Initially, every AUV places bids for the preset tasks, after which the distance cost list is stored. This list is defined by Equation (4), where d_{kn} represents the distance from the k th AUV to the n th task point.

$$D_{i,j} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{k1} & d_{m2} & \cdots & d_{kn} \end{bmatrix} \quad (4)$$

The optimization is based on Equation (1), incorporating both the benefit function $C_{i,j}$ and the profit value $\psi_{k,n}$. In the multi-AUV system, the cost is distance-based; the longer the distance, the higher the cost, and consequently, the lower the benefit. Therefore, the benefit function C is inversely proportional to D . In light of this, a conversion parameter β is introduced, set as $\beta = -1$. The optimized model is then defined as follows:

$$\begin{aligned} \max \sum_{i \in R}^k \sum_{j \in f}^n (C_{i,j} \cdot \psi_{i,j}) \cdot \text{result}_{i,j} \\ C_{i,j} = \beta \cdot D_{i,j} \\ \psi \in [0, 1] \end{aligned} \quad (5)$$

It is crucial to note that the profit value $\psi_{i,j}$ represents the reward value for the i th AUV performing the j th task. As per the settings in this study, if a task is uniquely assigned to an AUV, the reward value is higher and vice versa. It is clear that maximizing the benefit, that is, ensuring the shortest $d_{i,j}$ each time, is paramount. Therefore, the key focus of this study is the calculation of $d_{i,j}$. The pseudocode process for task assignment is as follows (Algorithm 1):

Algorithm 1. The pseudo-code of the proposed algorithms

Input:

Set of static targets T ; Set of AUVs A .

Total number of inspection tasks n ; total number of AUVs k ;

Output:

Assignment for AUVs : $\text{result}_{k,n}$;

Total trajectory for AUVs : $D_{k,n}$;

```

1: Initialization:  $C, \text{result} = \emptyset$ 
2: Calculate  $D$  based on improved Reeds–Shepp algorithm
3:  $C_{k,n} = \beta \cdot D_{k,n}$ 
4: for  $i = 1 \rightarrow k$  do
5:    $T_i = \text{argmax} C_{i,n} \cdot \psi_{i,n}$ 
6:   if task  $T_i$  is unassigned then
7:     Assign task  $T_i$  to AUV  $k$ 
8:   else
9:      $\text{result} = \text{result} \cup \{i\}$  add  $i$  to the set of unassigned AUVs
10:  end if
11: end for
12: while  $\text{result} \neq \emptyset$  do
13:   Initialization :  $\text{Task}' \neq \emptyset$ 
14:   for  $i \in \text{result}$  do
15:      $T_i = \text{argmax} C_{i,n} \cdot \psi_{i,n}$ 
16:      $\text{Task}' = \text{Task}' \cup T_i$ 
17:      $b_i = C_{i,n} \cdot \psi_{i,n}$ 
18:      $b_i' = \max C_{i,n} \cdot \psi_{i,n}$ 
19:     Update calculation bidding increment
20:   end for
21:   for  $T \in \text{Task}'$  do
22:     Assign task  $T$  to the AUV $i$ 
23:   end for
24: end while

```

3.2. Improved Reeds–Shepp Method on Path Planning

To address the path planning problem for underactuated AUVs with initial and final direction constraints, we introduce an enhanced Reeds–Shepp algorithm. In this approach, we utilize a car-like model for AUV path planning, taking full account of its underdrive characteristics. The Reeds–Shepp algorithm, initially proposed by J.A. Reeds and L.A. Shepp in 1988, builds upon the foundation of the Dubins algorithm. While the Dubins

algorithm effectively addresses the car-forward problem, it does not consider car-backward characteristics. The motion characteristics of AUVs are akin to those of a car, encompassing both forward and reverse movement capabilities, as depicted in Figure 4. In traditional path planning algorithms, the ability of AUVs to move backward is typically overlooked. However, our proposed algorithm allows for reverse motions at any point in time, as illustrated by the brown AUV in the figure. It is important to note that, due to being underactuated, AUVs lack side thrusters, which means they cannot move horizontally. Consequently, the red square shown in Figure 4 represents positions that cannot be directly reached by the AUV at the subsequent time point.

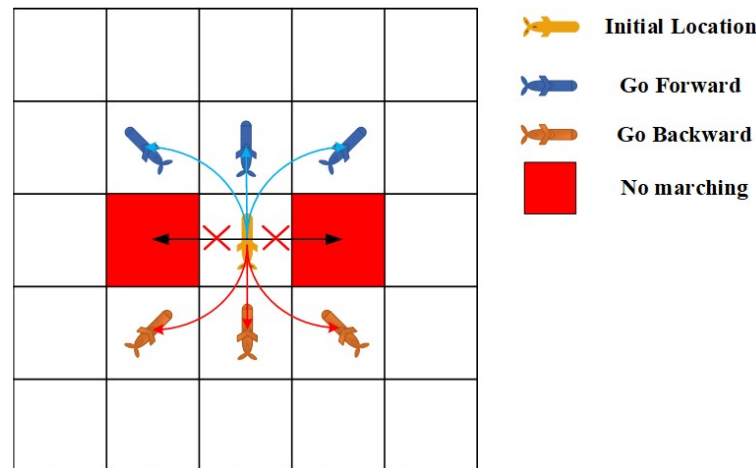


Figure 4. The path of the AUV moving is represented by the grid method.

The Dubins curve is mainly divided into two kinds of tracks: CSC (curve, straight, and curve) and CCC (curve, curve, and curve). Based on this, the Reeds–Shepp algorithm consists mainly of two paths: $C^\pm$ and $S^\pm$. Among them, the $^+$ indicates that the AUV is forward, and the $^-$ indicates that the AUV is backward.

$$\begin{aligned} x(t) &= x(0) + \int_0^t \varepsilon(\tau) \cos \phi(\tau) d\tau \\ y(t) &= y(0) + \int_0^t \varepsilon(\tau) \sin \phi(\tau) d\tau \\ \phi(t) &= \phi(0) + \int_0^t \eta(\tau) d\tau \end{aligned} \quad (6)$$

Formula (6) represents the state of an AUV at a given instant t , where t is the arc length, which is specified by AUV's position $(x(t), y(t))$. The planned path is the function $\gamma(t) = (x(t), y(t), \phi(t))$ for which we can find measurable functions ε and η . For each τ , $|\varepsilon(\tau)| = 1$ and $|\eta(\tau)| \leq 1$.

$$\sqrt{\dot{x}^2 + \dot{y}^2}(\tau) = |\varepsilon(\tau)| = 1 \quad (7)$$

The existence of $|\varepsilon(\tau)| = 1$ restricts the AUV's speed and cannot vary its preset direction $\phi(\tau)$ faster than on radian per time unit. Since the curvature is the reciprocal of the turning radius, its turning radius r is at least 1, or the curvature of its path is at most 1. When $\varepsilon(t)$ changes sign instantly, it must allow infinite acceleration at its cusp. Then, it must satisfy formula (8). Unlike vehicles traveling on land, the AUV travels relatively slowly in underwater, so it is reasonably compromised to achieve tractability.

$$\ddot{x} + \ddot{y} \leq g < \infty \quad (8)$$

$$\begin{aligned} \dot{x}(t) &= \varepsilon(t) \cos \phi(t) \\ \dot{y}(t) &= \varepsilon(t) \sin \phi(t) \\ \dot{\phi}(t) &= \eta(t) \end{aligned} \quad (9)$$

The trajectory of an admissible curve $\gamma(t)$ is $\bar{\gamma}(t) = (x(t), y(t))$; Formula (9) is differentiated through this. So, $\gamma(t)$ and $\bar{\gamma}(t)$ are rectifiable. We assume that z is admissible with $z(t_1) = L_1$ and $z(t_2) = L_2$ ($t_1 < t < t_2$); it delineates a trajectory leading from L_1 to L_2 under the restriction $\gamma = z|_{(t_1, t_2)}$. Then, its length is defined by $d(\gamma) = L_1 - L_2$. In short, the ultimate goal of the algorithm is to minimize $d(\gamma)$ so that the path is optimal.

Based on the C and S curves in the Dubins algorithm, a path of $l^{\pm}r^{\pm}s^{\pm}$ is constructed in the Reeds–Shepp algorithm, which stands for “go left”, “go right”, and “go straight” and “backward left”, “backward right”, and “backward straight”.

The following notations are the main path planning formulas. For $t \in \mathbb{R}$, let L_t, R_t, S_t be the position and direction angle at time t of a unit circle.

$$\begin{aligned} L_t &: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \\ R_t &: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \\ S_t &: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \end{aligned} \quad (10)$$

So, a $l_r^+ r_u^- s_v^- l_w^-$ curve, which starts from (x, y, ϕ) and ends at (12), is constructed by Formula (11).

$$\begin{aligned} &\text{go right :} \\ &R_t(x, y, \phi) = (x - \sin(\phi - t) + \sin \phi, y + \cos(\phi - t) - \cos \phi, \phi - t) \\ &\text{go left :} \\ &L_t(x, y, \phi) = (x + \sin(\phi + t) - \sin \phi, y - \cos(\phi + t) + \cos \phi, \phi + t) \\ &\text{go straight :} \\ &S_t(x, y, \phi) = (x + t \cos \phi, y + t \sin \phi, \phi) \end{aligned} \quad (11)$$

$$(X(t, a, b, c), Y(t, a, b, c), \Phi(t, a, b, c)) = R_c(S_b(R_a(L_t(x, y, \phi)))) \quad (12)$$

Then, its total length is shown as (13), where t, a, b, c (walking straight represents a straight-line distance) represents the four-segment constituent arc length of the trajectory curve. This is just one path demonstrated, and the rest of the path is calculated on the same principle.

$$d(t, u, v, w) = |t| + |a| + |b| + |c| \quad (13)$$

However, when planning the route of the AUV, at some moments, the entire route will be driven backward, which is not in line with reality, because the backward performance of the AUV is not as good as the forward performance. Therefore, based on this algorithm, this paper adds the restriction condition ς_{back} . When the length of the path curve of each planned backward run is less than ς_{back} , the algorithm will plan other optimal paths. Since this standard has not been set in the past, this article sets this minimum limit according to the actual situation.

$$\varsigma_{back} \geq -2 \quad (14)$$

Compared with the original Reeds–Shepp algorithm, another improvement made in this paper is to apply the particle swarm optimization (PSO) algorithm to optimize the turning radius r so that the planned path length will be better.

The PSO algorithm obtains the optimal solution by initializing ant colony random particles, which are updated by tracking two extremums $pbest$ and $gbest$ in each iteration. The speed update formula is as shown in (15), where V_i is the velocity of particle i , and X_i is the current position of particle i . c_1 and c_2 are learning factors, and R_1 and R_2 are random numbers that intervene between 0 and 1.

$$\begin{aligned} V_i^{\varepsilon+1} &= \mu V_i^{\varepsilon} + c_1 R_1 (pbest_i^{\varepsilon} - X_i^{\varepsilon}) + c_2 R_2 (gbest^{\varepsilon} - X_i^{\varepsilon}) \\ X_i^{\varepsilon+1} &= X_i^{\varepsilon} + V_i^{\varepsilon+1} \end{aligned} \quad (15)$$

Individual optimal index $pbest$ and overall optimal index $gbest$ are updated by Equation (16).

$$\begin{aligned} pbest_i^{\varepsilon+1} &= \begin{cases} pbest_i^l, F(pbest_i^l) \leq F(X_i^{\varepsilon+1}) \\ X_i^{\varepsilon+1}, F(pbest_i^l) > F(X_i^{\varepsilon+1}) \end{cases} \\ gbest^{\varepsilon+1} &= \begin{cases} gbest^{\varepsilon}, F(gbest^{\varepsilon}) \leq F(pbest_i^{\varepsilon+1}) \\ pbest_i^{\varepsilon+1}, F(gbest^{\varepsilon}) > F(pbest_i^{\varepsilon+1}) \end{cases} \end{aligned} \quad (16)$$

By fusing PSO with the Reeds–Shepp algorithm, the path where all particles are located is finally output so as to improve the effectiveness of the path planning of AUVs. The particle swarm size is adjusted according to the environmental scale, and the final degree of convergence of the particle swarm is controlled by adjusting the parameters so that the particle swarm can almost completely converge to the optimal path without too much divergence.

4. Simulation and Discussion

In this section, we begin by conducting simulations to assess the effectiveness of the proposed algorithm. Firstly, the task allocation of the distributed auction algorithm used in this paper is simulated, and the total task consumption under the SOM algorithm and K-means algorithm is compared to verify the efficiency of the proposed distributed auction algorithm. Then, the path planning of the proposed improved Reeds–Shepp algorithm is simulated, and the total task cost under the original Reeds–Shepp algorithm and the Dubins algorithm is compared to verify the efficiency of the proposed improved Reeds–Shepp algorithm. Additionally, we validate the algorithm’s applicability to a broader range of tasks within the context of bridge inspection environments. The simulation programs are implemented using MATLAB 2020 and executed on a computer equipped with a 3.2 GHz AMD Ryzen 7 5800H CPU and 32 GB of memory. This article assumes that there is communication between the AUVs.

4.1. Allocation Methods Comparison

We have defined the starting positions and angles for four AUVs as follows: AUV1 (5, 5, $\pi/2$), AUV2 (35, 5, $\pi/2$), AUV3 (65, 5, $\pi/2$), and AUV4 (95, 5, $\pi/2$). This section primarily simulates the path planning using the improved Reeds–Shepp algorithm under three distribution methods: distributed auction, SOM, and K-means. Eleven tasks are preset, and the simulation results are depicted in Figure 5. Under the distributed auction method, no AUV remains idle, and each task is executed by an AUV in every round, resulting in a total distance of 338.7 m. Figure 5b showcases the results using the SOM method. SOM primarily relies on the allocation of greedy algorithms, wherein AUV2 is assigned five tasks and AUV4 is assigned only one task. This uneven distribution may lead to some AUVs being idle at certain times, preventing the system from reaching a globally optimal state. The total distance covered in this method is 375.8 m. Figure 6c presents the allocation results using the K-means method. This method primarily divides the set of tasks into several clusters, with each AUV executing the tasks in the cluster that is closest to it. The total distance covered in this method is 364.7 m. The path data are displayed in Table 1.

The Monte Carlo experiment was performed on the three distribution methods to further demonstrate the advantage of the distributed auction algorithm, which is presented in Section 4.3.

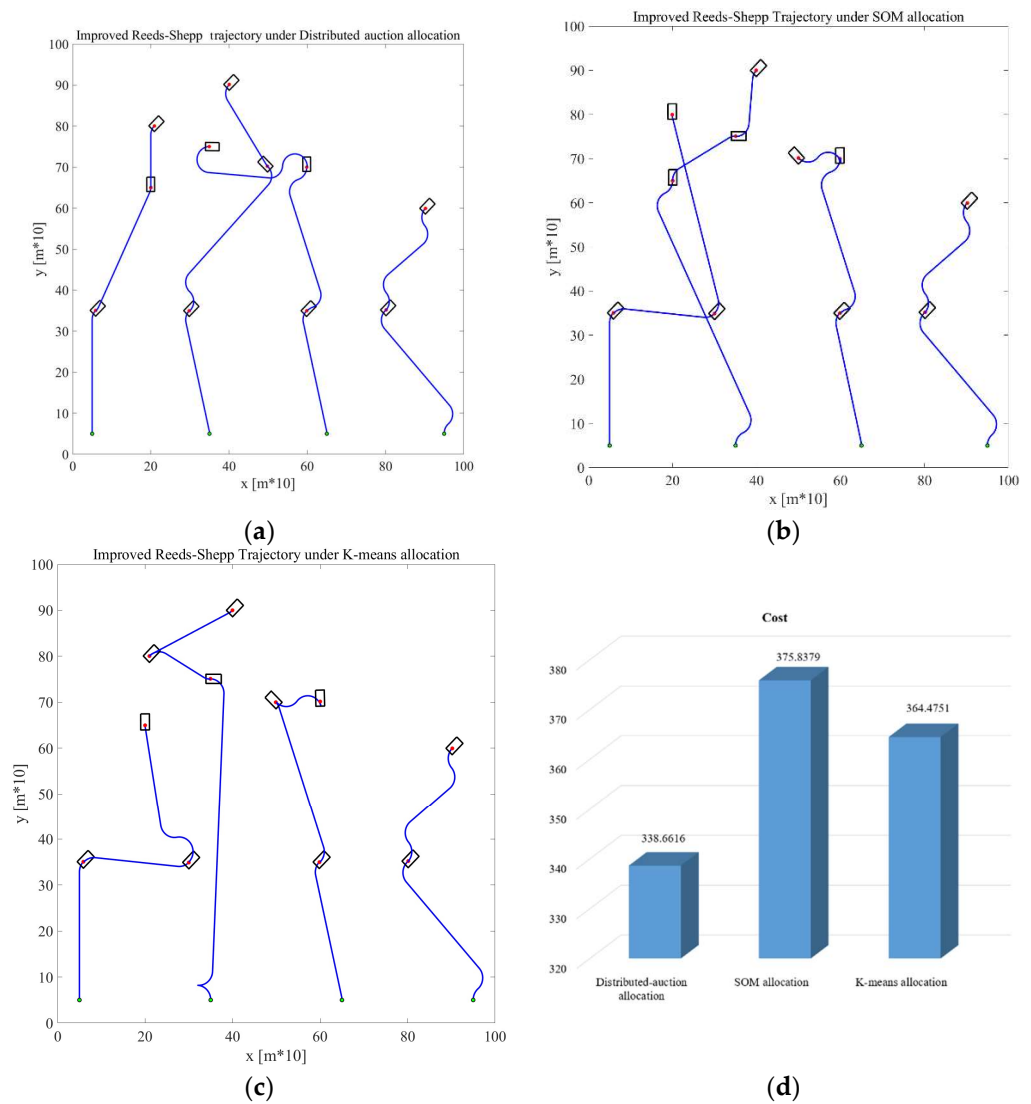


Figure 5. (a) Improved Reeds–Shepp trajectory under distributed auction allocation. (b) Improved Reeds–Shepp trajectory under SOM allocation. (c) Improved Reeds–Shepp trajectory under K-means allocation. (d) Cost.

Table 1. Path data.

Method	AUV1	AUV2	AUV3	AUV4	Cost
Auction	$(6, 35, \pi/4) \rightarrow (20, 65, \pi/2) \rightarrow (20, 80, \pi/4)$	$(30, 35, \pi/4) \rightarrow (50, 70, 3 \times \pi/4) \rightarrow (40, 70, \pi/4)$	$(60, 35, \pi/4) \rightarrow (60, 70, \pi/2) \rightarrow (35, 75, 0)$	$(80, 35, \pi/4) \rightarrow (90, 60, \pi/4)$	338.6616
SOM	$(6, 35, \pi/4) \rightarrow (30, 35, \pi/4) \rightarrow (20, 80, \pi/4)$	$(20, 65, \pi/2) \rightarrow (35, 75, 0) \rightarrow (40, 90, \pi/4)$	$(60, 35, \pi/4) \rightarrow (60, 70, \pi/2) \rightarrow (50, 70, 3 \times \pi/4)$	$(80, 35, \pi/4) \rightarrow (90, 60, \pi/4)$	375.8379
K-means	$(6, 35, \pi/4) \rightarrow (30, 35, \pi/4) \rightarrow (20, 65, \pi/2)$	$(35, 75, 0) \rightarrow (20, 80, \pi/2) \rightarrow (40, 90, \pi/4)$	$(60, 35, \pi/4) \rightarrow (50, 70, 3 \times \pi/4) \rightarrow (60, 70, \pi/2)$	$(80, 35, \pi/4) \rightarrow (90, 60, \pi/4)$	364.4751

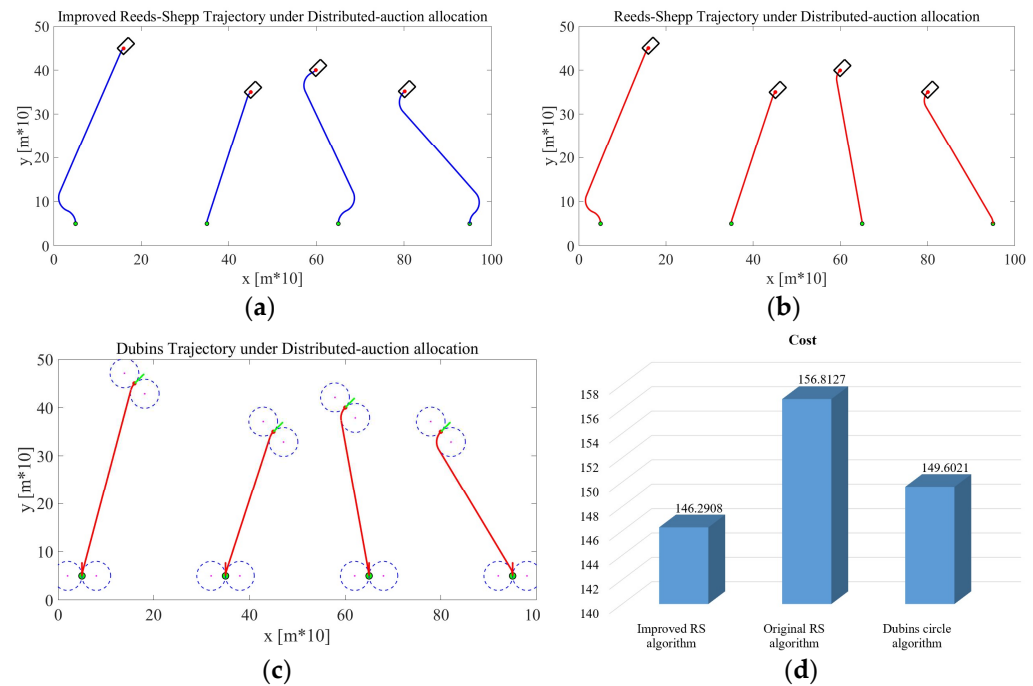


Figure 6. When the amount of tasks is the same as the number of AUVs ($k = n$). (a) Improved Reeds–Shepp method. (b) Reeds–Shepp method. (c) Dubins circle method. (d) Cost.

4.2. Path Planning Methods Comparison

Furthermore, a 100×50 ($m \times 10$) unit stationary environment is designed to test the algorithms' effectiveness. Four AUVs, AUV₁ ($5, 5, \pi/2$), AUV₂ ($35, 5, \pi/2$), AUV₃ ($65, 5, \pi/2$), and AUV₄ ($95, 5, \pi/2$), were used to reach the four target points: Target₁ ($16, 45, \pi/4$), Target₂ ($45, 35, \pi/4$), Target₃ ($60, 40, \pi/4$), and Target₄ ($80, 35, \pi/4$). Table 2 shows the path data under different path planning methods.

Table 2. Path data.

Method	AUV1	AUV2	AUV3	AUV4	Cost
Improve RS	$l^+ r^+ s^+ r^+$	$r^+ s^+ r^+$	$r^+ l^+ s^+ r^+$	$r^+ l^+ s^+ r^+$	146.29
RS	$l^+ r^+ s^+ r^+$	$r^+ s^+ r^+$	$l^+ s^+ r^+$	$l^+ s^+ r^+$	156.81
Dubins	CSC	CSC	CSC	CSC	149.60

In this simulation environment, the number of tasks is equal to the number of AUVs. Figure 6a presents the simulation results using the improved Reeds–Shepp algorithm. With this method, each AUV reaches its designated task location, with AUV1 executing Target1, AUV2 executing Target2, AUV3 executing Target3, and AUV4 executing Target4. The total distance covered is 146.3 m. Figure 6b illustrates the simulation results using the original Reeds–Shepp algorithm. This method results in the same task allocation but covers a total distance of 156.8 m. The proposed algorithm can efficiently complete the assigned tasks with fewer AUVs. Figure 6c shows the results using the Dubins circle method, covering a total distance of 149.6 m. These results indicate that the proposed improved Reeds–Shepp algorithm performs better in an environment with fewer tasks.

To validate the effectiveness of the proposed algorithm in a high-task volume environment, a simulation under multi-task conditions was performed. A 100×100 ($m \times 10$) unit stationary environment was designed to test the algorithms' effectiveness.

Figure 7 displays the simulation results when the number of tasks exceeds the number of AUVs. The AUV location information remains unchanged, and the task points are set as follows: target1 ($6, 35, \pi/4$), target2 ($30, 35, \pi/4$), target3 ($60, 35, \pi/4$), target4 ($80, 35, \pi/4$), target5 ($50, 70, \pi \times 3/4$), target6 ($20, 65, \pi/2$), target7 ($60, 70, \pi/2$), target8 ($35,$

75, 0), target9 (40, 90, $\pi/4$), target10 (21, 80, $\pi/4$), and target11 (90, 60, $\pi/4$). Under the simulation of the proposed algorithm, the total path cost is 338.7 m, while under the original Reeds–Shepp algorithm, the total path cost is 375.5 m. Using the Dubins circle method, the total cost is 394.4 m. A comparison of the three methods reveals that the improved Reeds–Shepp algorithm proposed in this study significantly enhances the efficiency of the multi-AUV system.

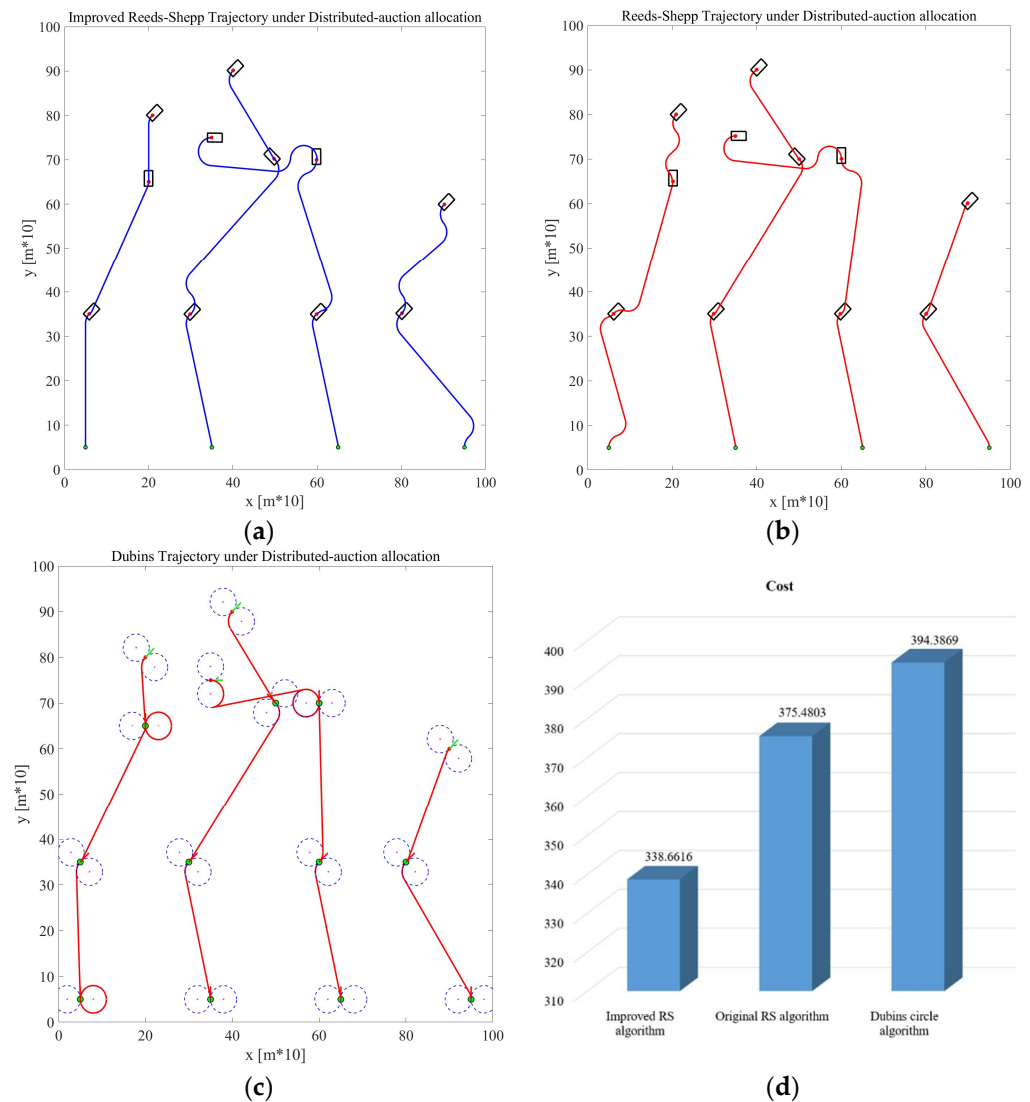


Figure 7. When the amount of tasks is more than the number of AUVs ($k < n$). (a) Improved Reeds–Shepp method. (b) Reeds–Shepp method. (c) Dubins circle method. (d) Cost.

The Dubins algorithm’s fundamental principle is to compute a concise trajectory from the initial position and direction to the target position and direction. This trajectory comprises a straight-line segment and a series of arcs, all adhering to the minimum turning radius constraints. Generally, Dubins curves are categorized into two main types: CSC (curve, straight, curve) and CCC (curve, curve, curve). The Dubins curve algorithm can effectively accommodate the directional requirements of AUV tasks. However, compared to the algorithm proposed in this study, it does not fully utilize the AUV’s motion characteristics, particularly the ability of the AUV to move backward.

From Figure 7c, it can be seen that when AUV₁ performs a task, in order to meet the requirements of the task execution angle, there will be an extra curve when performing a task. As a result, multitasking is costly.

4.3. Monte Carlo Experiments

In the task allocation simulation of multi-AUV systems, Monte Carlo experiments can be used to evaluate the performance of different task allocation strategies, help optimize task allocation strategies, and reduce potential risks.

First, we conducted 50 randomized trials for each of the three allocation methods in Section 4.1, randomly setting the information of 11 tasks, and the position information of the AUV remained unchanged. The results show that the distributed d-auction allocation method proposed in this paper has an efficiency increase of 14.17% and 13.25% compared with the SOM and K-means algorithms, respectively.

Furthermore, to verify the effectiveness of the proposed improved Reeds–Shepp algorithm in planning the detection path, we conducted 50 random experiments on the task amount $n = 5$, $n = 6$, $n = 7$ and randomly set the position point and execution direction angle of the task to ensure the reliability of the simulation results. Based on the random results of 150 experiments, the improved Reeds–Shepp algorithm proposed in this paper saves an average execution distance of 16.74% compared with the original Reeds–Shepp.

4.4. Bridge Inspection Simulation

To ensure that the proposed algorithm can be applied in a real bridge detection environment, a simulated pier environment is established. Figure 8 below shows a comparison of the improved Reeds–Shepp and the original Reeds–Shepp algorithm in a bridge inspection simulation. In Figure 8, the black solid circle is the flat display of the piers, and finally, the four simulated piers are inspected in all directions. Through the algorithm proposed in this paper, the multi-AUV system has a total driving distance of 391.9 m after performing the detection task. Under the original Reeds–Shepp algorithm, the multi-AUV system needs to travel 441.8 m to complete the task. The results show that the algorithm proposed in this paper is more efficient.

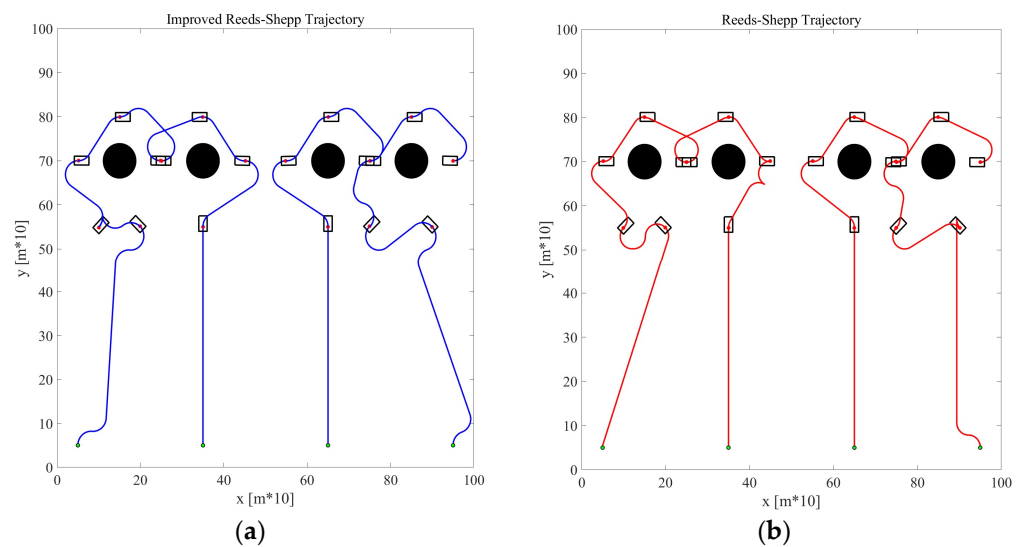


Figure 8. Simulations in bridge inspection environment. (a) Improved Reeds–Shepp algorithm. (b) Original Reeds–Shepp algorithm.

4.5. Simulation under Three-Dimensional Environment

The algorithm proposed in this paper can also be implemented in a three-dimensional environment. Figure 9 is a simulation in a three-dimensional environment. The three AUVs are shown in black, red, and blue curves. According to the simulation results, the proposed algorithm can consider the kinematic model of the AUV in the three-dimensional environment so as to reach the mission point and meet the task angle requirements.

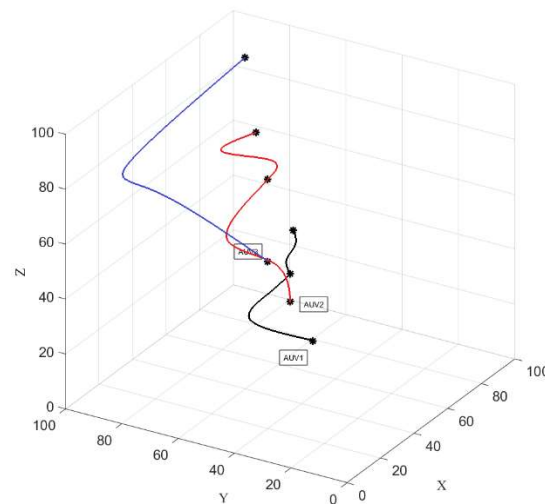


Figure 9. 3D underwater simulation.

5. Conclusions

The proposed improved Reeds–Shepp algorithm in this paper is a task allocation approach designed for multi-AUV systems based on distributed auctions. This algorithm introduces a car-like model, which takes into full account the kinematic model of AUVs and effectively leverages their reverse-driving capabilities, thus satisfying the specific detection angle requirements for AUVs in bridge inspection tasks. The proposed distribution method of distributed auction is more suitable for bridge inspection tasks than SOM and K-means algorithms. This method makes the AUV not vacant, and the specified AUV is executed in each round of detection tasks to achieve the minimum total cost of the system. Unlike the original Reeds–Shepp algorithm, this improved version incorporates constraints on reverse driving, rendering AUVs' path planning more realistic. Additionally, the algorithm employs the particle swarm optimization (PSO) technique to optimize the paths, enabling the entire system to find the optimal path under AUV's kinematic constraints to complete the assigned tasks. Compared to the Dubins curve algorithm, which limits AUVs to a forward motion only, this approach ensures that detection tasks can be accomplished even when AUVs need to move in reverse. This results in a fully utilized multi-AUV system, where detection tasks can be completed with shorter execution distances. Finally, the paper demonstrates the effectiveness of the multi-AUV system in handling multiple bridge inspection tasks.

Although the algorithm proposed in this paper can meet the angle requirements of AUVs well to perform the detection tasks and make the multi-AUV system perform the task with the lowest cost, in the actual path planning, the existence of various obstacles and ocean currents in the environment should be considered. Therefore, in the subsequent work, we will consider these factors to come up with a better solution.

Author Contributions: Conceptualization, H.L.; methodology, H.L.; software, H.L.; formal analysis, H.L.; investigation, M.C.; data curation, H.L. and H.Z.; writing—original draft preparation, H.L.; writing—review and editing, H.L.; supervision, M.C.; project administration, H.L. and T.W.; funding acquisition, M.C. and D.Z. 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 (52371331, 52001195, 62033009), the Creative Activity Plan for Science and Technology Commission of Shanghai (21DZ2293500), and the Natural Science Foundation of Shanghai (20510712300, 20dz1206700).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: Thanks to the National Natural Science Foundation of China and Shanghai Commission of Science and Technology for their support and funding.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Giglioni, V.; Poole, J.; Venanzi, I.; Ubertini, F.; Worden, K. On the use of domain adaptation techniques for bridge damage detection in a changing environment. *Ce/Papers* **2023**, *6*, 975–980. [\[CrossRef\]](#)
2. Wang, L.; Zhu, D.; Pang, W.; Zhang, Y. A survey of underwater search for multi-target using Multi-AUV: Task allocation, path planning, and formation control. *Ocean Eng.* **2023**, *278*, 114393. [\[CrossRef\]](#)
3. Bai, G.; Chen, Y.; Hu, X.; Shi, Y.; Jiang, W.; Zhang, X. Multi-AUV dynamic trajectory optimization and collaborative search combined with task urgency and energy consumption scheduling in 3-D underwater environment with random ocean currents and uncertain obstacles. *Ocean Eng.* **2023**, *275*, 113841. [\[CrossRef\]](#)
4. Sahoo, A.; Dwivedy, S.K.; Robi, P.S. Advancements in the field of autonomous underwater vehicle. *Ocean Eng.* **2019**, *181*, 145–160. [\[CrossRef\]](#)
5. Meng, C.; Zhang, W.; Du, X. Finite-time extended state observer based collision-free leaderless formation control of multiple AUVs via event-triggered control. *Ocean Eng.* **2023**, *268*, 113605. [\[CrossRef\]](#)
6. Cai, C.; Chen, J.; Liu, F. A Task Allocation Method for Multi-AUV Search and Rescue with Possible Target Area. *J. Mar. Sci. Eng.* **2023**, *11*, 804. [\[CrossRef\]](#)
7. Chen, Y.; Yang, D.; Yu, J. Multi-UAV Task Assignment With Parameter and Time-Sensitive Uncertainties Using Modified Two-Part Wolf Pack Search Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2853–2872. [\[CrossRef\]](#)
8. Li, J.; Zhang, K.; Xia, G. Multi-AUV cooperative task allocation based on improved contract network. In Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 6–9 August 2017; pp. 608–613.
9. Chakraa, H.; Guérin, F.; Leclercq, E.; Lefebvre, D. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. *Robot. Auton. Syst.* **2023**, *168*, 104492. [\[CrossRef\]](#)
10. 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, Aberdeen, UK, 19–22 June 2017; pp. 1–6. [\[CrossRef\]](#)
11. Wei, D.; Ma, H.; Yu, H.; Dai, X.; Wang, G.; Peng, B. A Hyperheuristic Algorithm Based on Evolutionary Strategy for Complex Mission Planning of AUVs in Marine Environment. *IEEE J. Ocean Eng.* **2022**, *47*, 936–949. [\[CrossRef\]](#)
12. 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. [\[CrossRef\]](#)
13. Li, X.; Guo, L.; Song, H. A robust auction algorithm for distributed heterogeneous multi-AUV task assignment. *J. Beijing Univ. Aeronaut. Astronaut.* **2022**, *48*, 736–746. [\[CrossRef\]](#)
14. Zlot, R.; Stentz, A. Market-based Multirobot Coordination for Complex Tasks. *Int. J. Robot. Res.* **2006**, *25*, 73–101. [\[CrossRef\]](#)
15. Zhu, D.; Huang, H.; Yang, S.X. Dynamic Task Assignment and Path Planning of Multi-AUV System Based on an Improved Self-Organizing Map and Velocity Synthesis Method in Three-Dimensional Underwater Workspace. *IEEE Trans. Cybern.* **2013**, *43*, 504–514. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Zhu, D.; Liu, Y.; Sun, B. Task Assignment and Path Planning of a Multi-AUV System Based on a Glasius Bio-Inspired Self-Organising Map Algorithm. *J. Navig.* **2018**, *71*, 482–496. [\[CrossRef\]](#)
17. Bertsekas, D.P. Auction algorithms for network flow problems: A tutorial introduction. *Comput. Optim. Appl.* **1992**, *1*, 7–66. [\[CrossRef\]](#)
18. Duan, X.; Liu, H.; Tang, H.; Cai, Q.; Zhang, F.; Han, X. A Novel Hybrid Auction Algorithm for Multi-UAVs Dynamic Task Assignment. *IEEE Access* **2020**, *8*, 86207–86222. [\[CrossRef\]](#)
19. Zavlanos, M.M.; Spesivtsev, L.; Pappas, G.J. A distributed auction algorithm for the assignment problem. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 1212–1217.
20. Zhang, Z.; Wang, J.; Xu, D.; Meng, Y. Task Allocation of Multi-AUVs Based on Innovative Auction Algorithm. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; pp. 83–88.
21. Lee, D.H.; Zaheer, S.A.; Kim, J.H. A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1469–1481. [\[CrossRef\]](#)
22. Cheng, Q.; Yin, D.; Yang, J.; Shen, L. An Auction-Based Multiple Constraints Task Allocation Algorithm for Multi-UAV System. In Proceedings of the 2016 International Conference on Cybernetics, Robotics and Control (CRC), Hong Kong, China, 19–21 August 2016; pp. 1–5.
23. Bin, D.; Rui, Z.; Jiang, W.; Shaodong, C. Distributed coordinated task allocation for heterogeneous UAVs based on capacities. In Proceedings of the 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June 2013; pp. 1927–1932.

24. Haghighi, H.; Asadi, D.; Delahaye, D. Multi-Objective Cooperated Path Planning of Multiple UAVs Based on Revisit Time. *J. Aerosp. Inf. Syst.* **2021**, *18*, 919–932. [[CrossRef](#)]
25. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean Eng.* **2018**, *169*, 187–201. [[CrossRef](#)]
26. Liu, Y.; Han, D.; Wang, L.; Xu, C.-Z. HGHA: Task allocation and path planning for warehouse agents. *Assem. Autom.* **2021**, *41*, 165–173. [[CrossRef](#)]
27. Chow, B. Assigning Closely Spaced Targets to Multiple Autonomous Underwater Vehicles. *J. Ocean Technol.* **2009**, *6*, 57–77.
28. Reeds, J.A.; Shepp, L.A. Optimal paths for a car that goes both forwards and backwards. *Pac. J. Math.* **1990**, *145*, 367–393. [[CrossRef](#)]
29. Mao, Y.; Gao, F.; Zhang, Q.; Yang, Z. An AUV Target-Tracking Method Combining Imitation Learning and Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 383. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.