mathematics

MDPI

Article

Solving Heterogeneous USV Scheduling Problems by Problem-Specific Knowledge Based Meta-Heuristics with Q-Learning

Zhenfang Ma, Kaizhou Gao *, Hui Yu and Naiqi Wu

Macau Institute of System Engineering, Macau University of Science and Technology, Avenida Wai Long, Taipa 999078, Macau; 2220017692@student.must.edu.mo (Z.M.); 3220006100@student.must.edu.mo (H.Y.); nqwu@must.edu.mo (N.W.)

* Correspondence: kzgao@must.edu.mo

Abstract: This study focuses on the scheduling problem of heterogeneous unmanned surface vehicles (USVs) with obstacle avoidance pretreatment. The goal is to minimize the overall maximum completion time of USVs. First, we develop a mathematical model for the problem. Second, with obstacles, an A* algorithm is employed to generate a path between two points where tasks need to be performed. Third, three meta-heuristics, i.e., simulated annealing (SA), genetic algorithm (GA), and harmony search (HS), are employed and improved to solve the problems. Based on problem-specific knowledge, nine local search operators are designed to improve the performance of the proposed algorithms. In each iteration, three Q-learning strategies are used to select high-quality local search operators. We aim to improve the performance of meta-heuristics by using Q-learning-based local search operators. Finally, 13 instances with different scales are adopted to validate the effectiveness of the proposed strategies. We compare with the classical meta-heuristics and the existing meta-heuristics. The proposed meta-heuristics with Q-learning are overall better than the compared ones. The results and comparisons show that HS with the second Q-learning, HS + QL2, exhibits the strongest competitiveness (the smallest mean rank value 1.00) among 15 algorithms.

Keywords: unmanned surface vessel; scheduling; meta-heuristics; Q-learning

MSC: 90B35

Citation: Ma, Z.; Gao, K.; Yu, H.; Wu, N. Solving Heterogeneous USV Scheduling Problems by Problem-Specific Knowledge Based Meta-Heuristics with Q-Learning. *Mathematics* **2024**, *12*, 339. https://doi.org/10.3390/math12020339

Academic Editor: José Antonio Sanz

Received: 21 December 2023
Revised: 16 January 2024
Accepted: 18 January 2024
Published: 19 January 2024

1. Introduction

Oceans cover two-thirds of the earth's area [1]; however, most of the oceans have not been explored. Satellites, submersibles, and vessels are commonly used equipment in ocean exploration, each playing a role in different fields. Satellites are mainly used for ocean resource management and disaster monitoring by collecting large-scale and high-frequency ocean information [2]. However, satellites are affected by many factors, such as clouds, atmosphere, water color, etc., and cannot provide high-resolution and high-accuracy data [3]. Submersibles can directly observe and analyze seabed resources, such that direct data or samples can be obtained. Still, the operations of submersibles are affected by water pressure, temperature, and battery capacity. They cannot last for a long time or travel over a large area, and their execution needs vessels or satellites for support and positioning [4]. Vessels can serve as platforms for ocean exploration and are applied in many fields, such as marine environmental protection, channel measurement, and ocean observation.

Unmanned surface vehicles (USVs) are characterized by unattended operations, high payload capacity, low cost, and high maneuverability [5], and have been widely used in civil and military fields [6]. Xie et al. [7] proposed a hybrid partitioning patrol scheme, which divided the sea surface area into different importance levels and guides USVs to accomplish

patrol tasks. Li et al. [8] addressed the maritime safety issue under severe weather conditions. Inspired by an immune–endocrine short feedback system, they presented a novel approach to make USVs able to fully exploit their strengths and accomplish patrol tasks in complex sea conditions. In terms of marine environment monitoring, Sutton et al. [9] conducted the first autonomous circumnavigation of Antarctica using a USV, measuring sea–air carbon dioxide, wind speed, and surface ocean properties. In [10], high-precision data were collected from the coastal water of Belize by using a USV equipped with pH and pCO2 sensors. Based on a wave-adaptive modular vessel USV, Sinisterra et al. [11] used a stereo vision-based method to track moving target vessels on the sea surface. Benefiting from the communication and carrying capacity of USVs, Shao et al. [12] proposed a collaborative unmanned surface vehicle–unmanned aerial vehicle platform. For collaborating unmanned surface vehicle–autonomous underwater vehicles, Lei Yang et al. [4] proposed three operation modes. Both teams used USVs as platforms to collaborate with UAVs and AUVs, respectively, extending the utilization and capabilities of the devices.

The main contributions of this study are summarized as follows:

(1) a mixed integer linear programming model is established to describe the heterogeneous USV scheduling problems for minimizing the maximum completion time;
(2) problem-specific knowledge-based nine neighborhood search operators are designed to improve the performance of metaheuristics;
(3) three Q-learning strategies are proposed to guide the selection of premium neighborhood search operators during iterations.

The rest of this paper is structured as follows. Section 2 reviews the publications on this topic. Section 3 introduces the mathematical model for the USV scheduling problems. Section 4 presents the proposed algorithms. Section 5 reports the experiential results and comparisons, and finally, Section 6 summarizes the study and points out some future directions.

2. Literature Review

In this section, we present an overview of the published literature in four dimensions: path planning, meta-heuristic, Q-learning, and the problem's heterogeneity.

Path planning of USVs on the sea surface is the basis for completing predetermined task objectives with high speed, safety, and energy efficiency. In [13], path planning problems were divided into two layers: local path planning and global one. In local path planning, robots have limited knowledge of the environment, while in global path planning, they have a complete understanding of the environment [13]. In our study, the known environmental information is relatively complete and remains stable over time, so a static and global path planning method is chosen [14]. Based on an automatic identification system service platform, Kai Yu et al. [14] proposed an improved A* algorithm, which allows ships to change their speed to avoid obstacles during global path planning. Yang et al. [15] proposed a global path planning algorithm based on a double deep Q network for the global path generation. In [16], a parallel evolutionary genetic algorithm (PEGA) was proposed, which was implemented on a chip by a hardware and software collaboration design method. The chip was used for global path planning of autonomous mobile robots. Yin et al. [17] combined an adaptive agent modeling with a rapidly exploring random tree star (RRT*) and proposed a reliability-based path planning algorithm ER-RRT* with better performance.

Meta-heuristics have been widely applied in engineering optimization and scheduling problems, such as traffic signal control [18], flow shop scheduling optimization [19], etc. Compared to other algorithms, meta-heuristics can balance the computational cost and the accuracy of the results [20]. They are also employed for solving task assignment and path planning problems [21]. Gemeinder et al. [22] used GA to design a mobile robot path planning software that focused on energy consumption. A hybrid of a GA and particle swarm optimization algorithm was developed to calculate the optimal path of fixed-wing unmanned aerial vehicles [23]. To improve the quality of the initial path,

Nazarahari et al. [24] developed an enhanced genetic algorithm (EGA), which used five customized crossover and mutation operators for the robots' path planning. An improved SA was developed by integrating two additional operators and path initialization rules in environments with static and dynamic obstacles [25]. Huo et al. [26] addressed UAV path planning problems in natural disaster rescue and battlefield collaborative action scenarios and proposed an improved GA algorithm by integrating simulated annealing to solve them. With the minimum energy consumption, Xiao et al. [27] designed an SA to balance task allocation and path planning for segmented multi-UAV image acquisition tasks. Based on five metaheuristics, Gao et al. [20] designed various heuristic rules and improving strategies to solve the scheduling problems of multiple USVs.

Meta-heuristics are prone to local optimum and have low convergence speed. As one of the most commonly used reinforcement learning algorithms, Q-learning has been employed to improve the performance of meta-heuristics in recent years [28–31]. Ren et al. [32] proposed a variable neighborhood search algorithm with Q-learning to solve disassembly line scheduling problems. In response to the slower learning speed of Q-learning, Low et al. [33] developed a modified Q-learning for path planning by optimizing path smoothness, time consumption, shortest distance, and total distance. Zhao et al. [34] proposed a hyper-heuristic algorithm with Q-learning to solve an energy-efficient distributed blocking flow shop scheduling problem. An efficient Q-learning was designed to solve path planning and obstacle avoidance problems for mobile robots, in which a new reward function was developed to improve the performance of Q-learning [35].

In the domain of unmanned vehicles (UVs) (such as AVs, UAVs, USVs, etc.), the problems related to heterogeneous UVs have attracted much attention. Heterogeneous UVs can be significantly more cost-effective and improve system performance by working cooperatively [36]. In large-scale applications, it is common for heterogeneous UAVs with different capabilities to cooperate [37]. Among the published works on heterogeneous USVs, some studies have explored task allocation [38], coordinated control [39,40], and path planning [41]. However, the scheduling problem for heterogeneous USVs has been less considered.

3. Problem Description

The concerned problem involves the coordination and optimization of multiple heterogeneous USVs with different capabilities and constraints. With obstacles between some points where tasks are to be performed, the distance between them should be calculated in advance. The objective is to minimize the maximum competition time when all USVs finish their tasks.

Figure 1 shows an example with three USVs and 13 tasks, where different tasks are assigned to three vessels of different types. Due to their limited battery capacity, USVs may have to return to the starting point to change a battery and replenish energy during the tasks. In this study, the task points are presented by a two-dimensional plane space with a length and width of 100 units. As shown in Figure 2, the modeling of the environment consists of three parts: boundaries, obstacles, and task points. All task points and obstacles are distributed within the boundaries.

In this problem, we have n tasks with a types, and V_a and K_a, respectively, represent sets consisting of vessels and missions of the type a, $n = n_1 + n_2 + n_3 + \ldots + n_a$; $V_1 = \{v_{11}, v_{12}, \ldots, v_{1n_1}\}$; $V_2 = \{v_{21}, v_{22}, \ldots, v_{2n_2}\}$; $\ldots$; $V_a = \{v_{a1}, v_{a2}, \ldots, v_{an_a}\}$. There are m USVs with b types, $m = m_1 + m_2 + m_3 + \ldots + m_b$; the different types of USVs are denoted as $K_1 = \{k_{11}, k_{12}, \ldots, k_{1m_1}\}$; $K_2 = \{k_{21}, k_{22}, \ldots, k_{2m_2}\}$; $\ldots$; $K_b = \{k_{b1}, k_{b2}, \ldots, k_{bm_b}\}$. All tasks are processed independently, and each USV cannot be interrupted once a task is started until it is completed. Each task can only be performed by one vessel. The ship moving time between task points is included in the total completion time. When all assigned tasks are completed, a USV returns to its departure point.

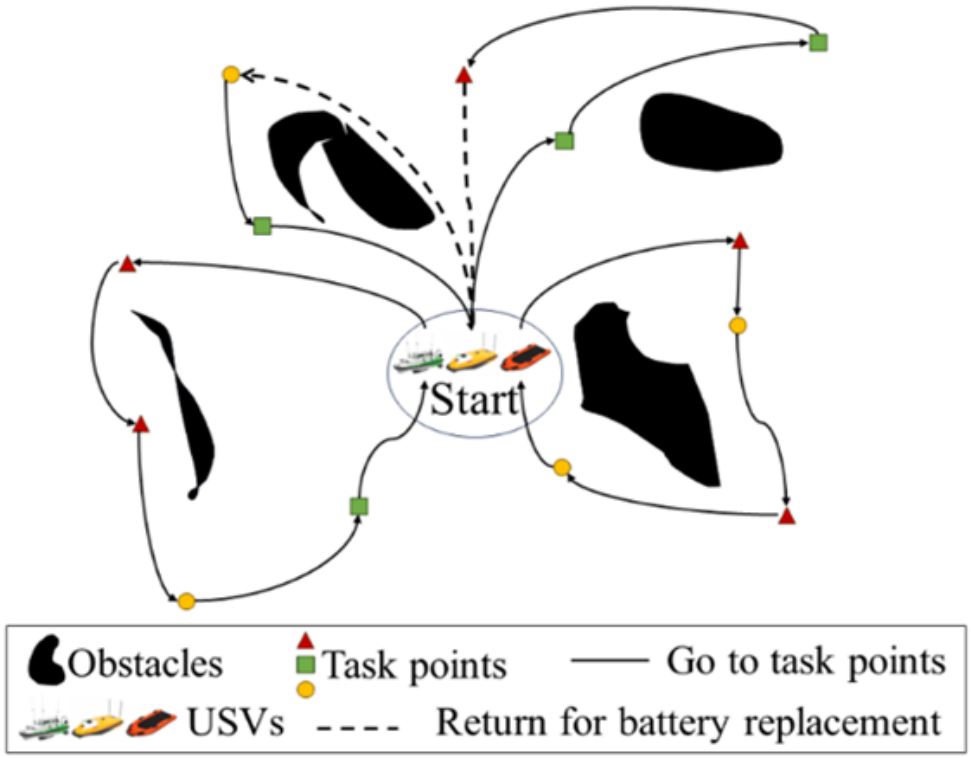

Figure 1. An example of scheduling problems with heterogeneous USVs.

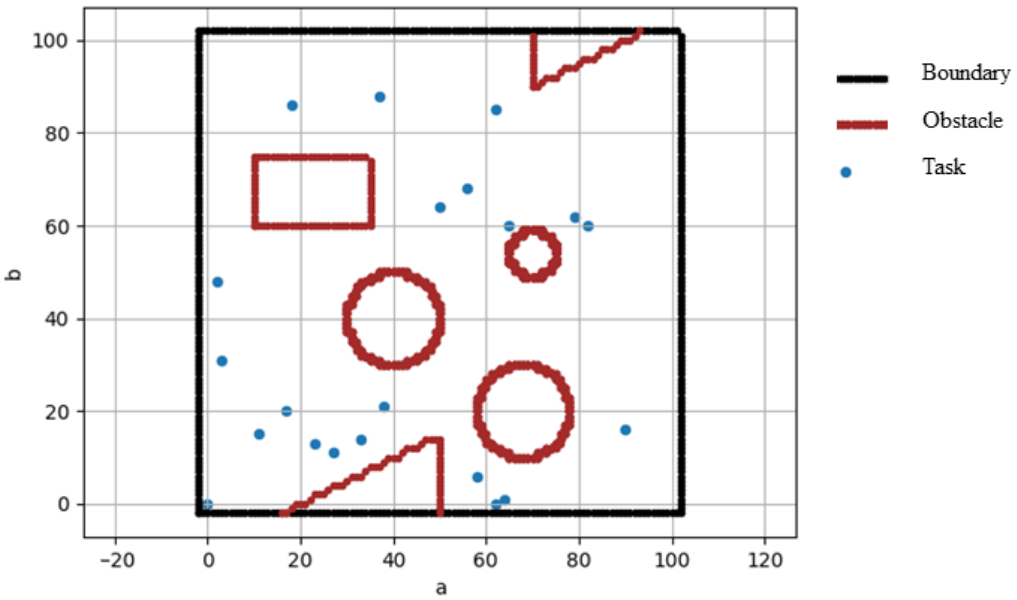

Figure 2. An illustrative example of the environment model.

The used notations in the model are described in Table 1.

In real-life scenarios, many factors are challenging in the global planning stage, such as wind, waves, ocean currents, and emergencies. In this study, triangular fuzzy numbers (TFN) are used to make the model more realistic. The following methods perform addition, rank, and maximization operations for triangular fuzzy numbers.

Table 1. The used notations in the model.

Notation	Description
i, j	Indices of tasks.
k	Index of USVs.
n	Total number of tasks.
m	The number of USVs.
d_{ij}	Length of the path between tasks i and j.
p	Speed of USVs.
t_{ij}	Travel time between tasks i and j.
t_{back}^{ij}	Additional time to travel from task i to the start point and from it to task j.
$\bar{t}_i$	Time required for performing task i.
B_k	Working time after battery replacement.
N_k	The number of return trips to the departure point to replace batteries.
W_k	The total time required for USV k to conduct travel and mapping.
R_k	The required total round-trip time of USV k.
$\mathrm{C_k}$	Completion time for USV k to perform its tasks.
n_{ij}	If the remaining charge after task i is insufficient for the next task j, $n_{ij} = 1$; else $n_{ij} = 0$.
x_{ijk}	If tasks i and j are assigned to USV k and task j is the successor of i, $x_{ijk} = 1$; otherwise, $x_{ijk} = 0$.
y_{ki}	Heterogeneous coefficient of vessel type matching task type.

Addition operation: two task times $\bar{t}_i = (\bar{t}_{i1}, \bar{t}_{i2}, \bar{t}_{i3})$ and $\bar{t}_j = \left(\bar{t}_{j_1}, \bar{t}_{j_2}, \bar{t}_{j_3}\right)$, their addition is as follows.

$$\bar{t}_i + \bar{t}_j = \left(\bar{t}_{i1} + \bar{t}_{j_1},\ \bar{t}_{i2} + \bar{t}_{j_2}, \bar{t}_{i3} + \bar{t}_{j_3}\right). \tag{1}$$

Ranking operation: three criteria are used to compare $\bar{t}_i$ and $\bar{t}_j$.

$$\text{If } \left(\bar{t}_{i1} + 2\bar{t}_{i2} + \bar{t}_{i3}\right)/4 > (<)\left(\bar{t}_{j_1} + 2\bar{t}_{j_2} + \bar{t}_{j_3}\right)/4, \text{ then } \bar{t}_i > (<)\bar{t}_j. \tag{2}$$

$$\text{If } \left(\bar{t}_{i1} + 2\bar{t}_{i2} + \bar{t}_{i3}\right)/4 = \left(\bar{t}_{j_1} + 2\bar{t}_{j_2} + \bar{t}_{j_3}\right)/4, \text{ but } \bar{t}_{i2} > (<)\bar{t}_{j_2}, \text{ then } \bar{t}_i > (<)\bar{t}_j. \tag{3}$$

$$\text{If} \left(\bar{t}_{i1} + 2\bar{t}_{i2} + \bar{t}_{i3}\right)/4 = \left(\bar{t}_{j_1} + 2\bar{t}_{j_2} + \bar{t}_{j_3}\right)/4, \text{ and } \bar{t}_{i2} = \bar{t}_{j_2}, \text{ but } \bar{t}_{i3} - \bar{t}_{i1} > (<)\bar{t}_{j_3} - \bar{t}_{j_1}, \text{ then } \bar{t}_i > (<)\bar{t}_j. \tag{4}$$

Max operation:

$$\text{If } \bar{t}_i > \bar{t}_j, \text{ then } \bar{t}_i \bigvee \bar{t}_j = \bar{t}_i, \text{ otherwise } \bar{t}_i \bigvee \bar{t}_j = \bar{t}_j. \tag{5}$$

The mathematical model is formulated as follows:

$$f = \min\left\{\max[\mathrm{C_1}, \mathrm{C_2}, \mathrm{C_3}, \ldots, \mathrm{C_k}]\right\}. \tag{6}$$

s.t.

$$t_{ij} = d_{ij}/p,\ \forall i, j \in V. \tag{7}$$

$$t_{back}^{ij} = t_{i0} + t_{0j}, \forall i, j \in V. \tag{8}$$

$$W_k = \sum_{i=0}^{n}\sum_{j=0}^{n} x_{ijk}(t_{ij} + y_{ki} * \bar{t}_i), \forall i, j \in V, \forall k \in K. \tag{9}$$

$$R_k = \sum_{i=0}^{n}\sum_{j=0}^{n} n_{ij} * t_{back}^{ij}, \forall i, j \in V. \tag{10}$$

$$\mathrm{C_k} = W_k + R_k\ , \forall k \in K. \tag{11}$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n} x_{ijk} = 1,\ \forall j \in V\backslash\{0\}. \tag{12}$$

$$\sum_{k=1}^{m}\sum_{j=0}^{n} x_{ijk} = 1,\ \forall i \in V\backslash\{0\}. \tag{13}$$

$$\sum_{i=0}^{n} x_{ijk} - \sum_{i=0}^{n} x_{jik} = 0,\ \forall k \in K, j \in V\backslash\{0\}. \tag{14}$$

$$\sum_{i=1}^{n} x_{i0k} - \sum_{j=1}^{n} x_{0jk} = 0,\ \forall k \in K. \tag{15}$$

$$m \geq k. \tag{16}$$

$$x_{ijk} \in \{0,1\}, \forall i,j \in V, i \neq j,\ \forall k \in K. \tag{17}$$

$$\bar{t}_0 = 0. \tag{18}$$

$$(N_k + 1) * B_k \geq W_k + R_k, \forall k \in K. \tag{19}$$

$$V_1 \cup V_2 \cup \ldots \cup V_a = V, a > 1. \tag{20}$$

$$K_1 \cup K_2 \cup \ldots \cup K_b = K, b > 1. \tag{21}$$

As shown in Equation (6), the objective is to minimize the maximum completion time, and the fitness values are expressed using TFN. The rules for calculating and comparing TFN follow Equations (1)–(5). Equation (7) is the travel time from task point i to task point j. Equation (8) expresses the time required for a single battery replacement. The total time for a USV to perform a task and travel to the next task point is given in Equation (9). In Equation (9), when the heterogeneous coefficient y_{ki} is infinity, it means that USVs of the same type with USV k are unable to perform the same type of tasks with task i. Equation (10) denotes the total time required for a USV to replace the battery. Equation (11) represents the completion time of USV k.

Equations (12)–(21) represent the corresponding constraints for the heterogeneous USV scheduling problems. Each task point should be visited once by a USV, which is indicated by constraints (12)–(14). Constraint (15) restricts that all USVs start from the departure point and return there after finishing all the assigned tasks. Constraints (16) and (17) limit the USV amount and index variables. Constraint (18) specifies that the completion time mapped by the start point is 0. Constraint (19) presents a limitation on the total battery capacity of the USVs. Constraints (20) and (21) place the types of tasks and boats.

4. Proposed Algorithms

There are two parts in this section, the first part is the preprocessing of the environment to avoid obstacles using the A* algorithm. The second part is the algorithm design, which includes the coding and decoding of the solution, the meta-heuristics, the local search operators, the Q-learning-based local search, and the framework of the proposed algorithms.

4.1. Path Search

A* algorithms are common for path planning [42] and one of the most effective methods for finding the shortest path in a static environment. They play a crucial role in the field of vehicle navigation. The flowchart of the A* algorithm is shown in Figure 3. When the A* algorithm starts iterating, it will search for the lowest cost grid around the starting point of the grid map. Then, it searches for the next lowest cost grid around this one and repeats the process until it arrives at the end point. The searched grids constitutes the shortest path from the start point to the end point. To solve the stationary obstacle avoidance problem for USVs, this study uses an A* algorithm to calculate the feasible path between task points to obtain the shortest path with obstacles being avoided.

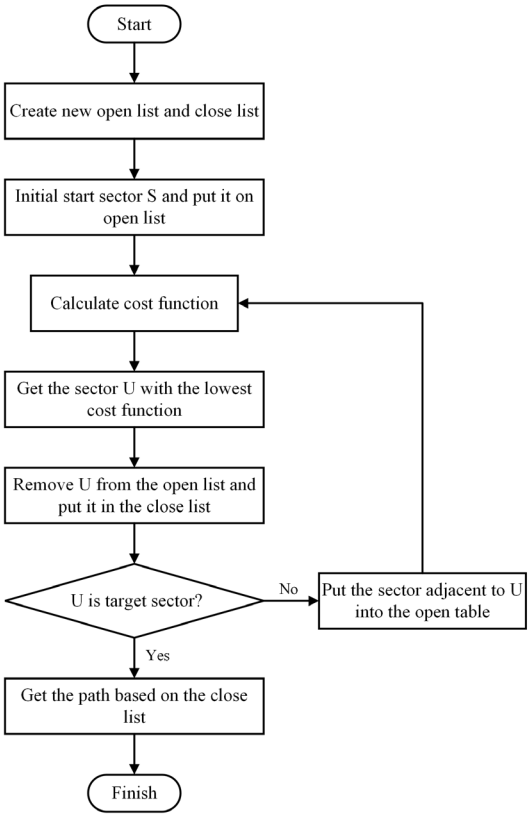

Figure 3. Flowchart of A* algorithm.

An example of the A* algorithm for path searching is shown in Figure 4. The length of the shortest path found by the A* algorithm is used to calculate the travel time between task points S and F.

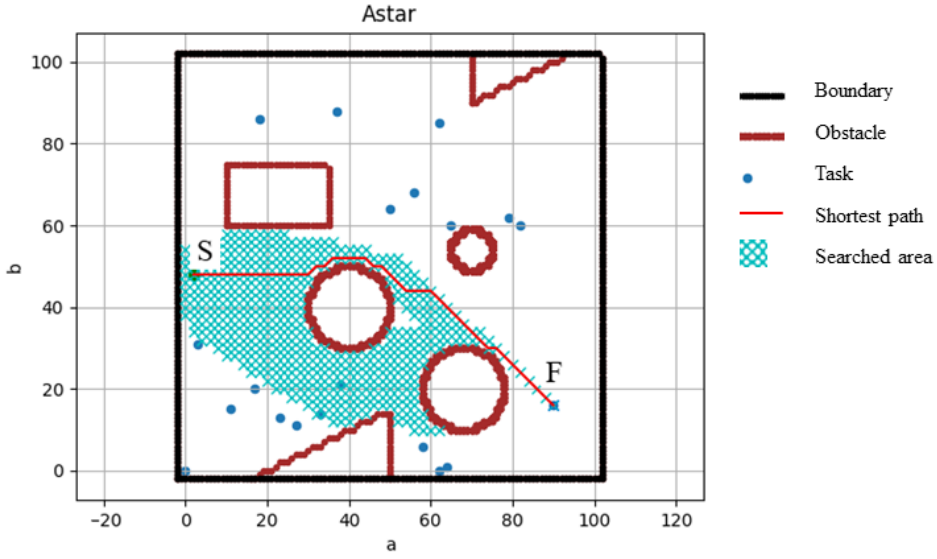

Figure 4. An example of A* Algorithm for path searching.

4.2. Solution Representation

A USV scheduling problem is to assign tasks to USVs and sort the assigned tasks for each USV. According to the characteristics of the problems, we design an encoding strategy to represent a solution. It is in the form of $L = (k_0, S_0, k_1, S_1, \ldots, k_n, S_n)$. A USV with serial number k is assigned to the sequence of tasks $S_k = (v_0, v_1, ..., v_a)$, where k or v is a vector

with three elements denoted as $[\partial_1, \partial_2, \partial_3]$ to present the attributes of USVs or tasks. If ∂_1 has a value of 0, it represents a USV, while if it has a value of 1, it represents a task; ∂_2 gives the type of the USV or task; and ∂_3 indicates its index.

An example for a solution is shown in Figure 5. In this solution, three ships perform seven tasks. For example, element $k_2 = [0, 2, 1]$ represents a USV with serial number 1 of type 2. Similarly,$v_0 = [1, 2, 0]$ represents a task of type 2 with serial number 0. The solution can be represented as $L = (k_0, v_1, v_0, v_2, k_1, v_3, v_4, v_6, k_2, v_5)$, and the sequence of tasks for each vessel can be represented as $S_1 = (v_1, v_0, v_2)$, $S_2 = (v_3, v_4, v_6)$, $S_3 = (v_5)$.

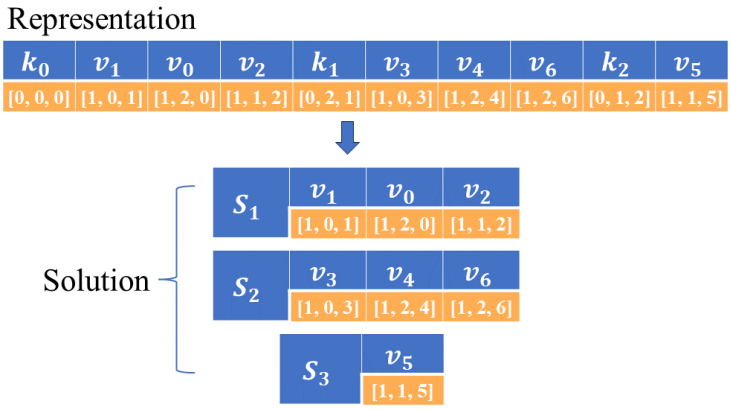

Figure 5. Representation of a solution.

4.3. Meta-Heuristics

This study employs three meta-heuristics, GA, SA, and HS, and proposes their variants by applying Q-learning-based local search operators. The three algorithms are widely used for solving various optimization and scheduling problems [43,44]. Figure 6 shows the flowchart of the meta-heuristics.

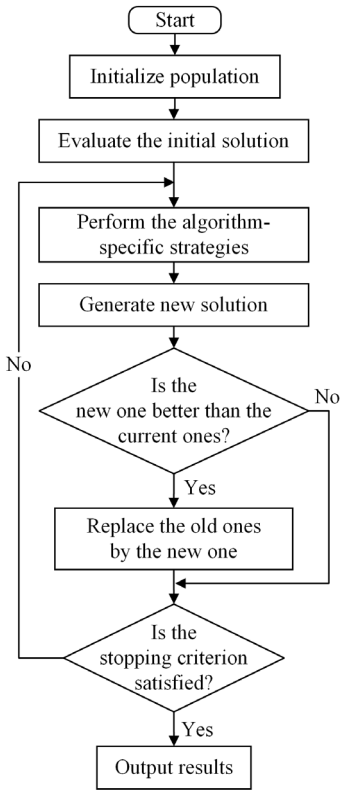

Figure 6. Flowchart of meta-heuristics.

4.4. Local Search

Classical meta-heuristics are characterized by low convergence efficiency and easily fall into local optimum for combinatorial optimization and scheduling problems. Based on problem-specific knowledge, we design nine local search (LS) operators to improve the convergence and solutions' quality and they are categorized into three types. The LS operators in the first type are for task assignment and sequencing of the same type of USVs, while the LSs of the second type are designed for different USV types. The third type of LSs are used for searching larger neighborhood space without considering heterogeneity.

Three LS operators, named as LS1, LS2, and LS3 in the first type, are designed to adjust the task allocation and completion order in the same type of USVs to find better neighbor solutions. As shown in Figure 7, LS1 is used to randomly select an element and insert it into another random position. LS2 selects two elements randomly and exchanges their positions. By LS3, a segment of a solution is randomly selected and the order of elements in it is reversed.

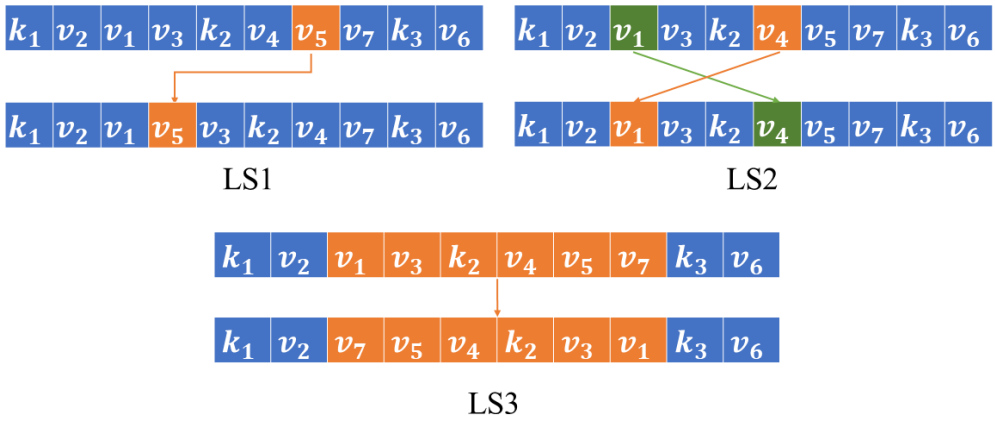

Figure 7. Example of LS1, LS2, and LS3.

The LS operators in the second type consider the heterogeneity of the USVs. LS4 selects an element randomly and inserts it into the sequence of a heterogeneous USV that is also randomly selected. LS5 exchanges the positions of two randomly selected elements from the sequences of two heterogeneous USVs. By LS6, a segment of a solution is randomly selected where the tasks belong to heterogeneous USVs. Reverse operation is executed on these elements and the assigned USV of each task may be changed. The detail for LS6 is shown in Figure 8.

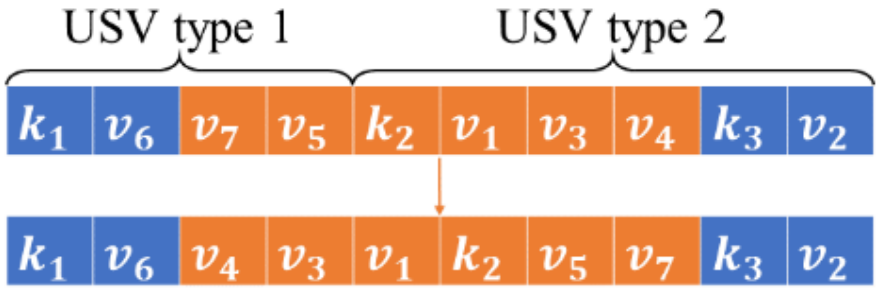

Figure 8. Example of LS6.

To extend the local search space, the third type of LSs are proposed, which do not consider the heterogeneity of USVs and use different neighborhood structures from the previous ones. In the third type, there are three LSs called LS7, LS8, and LS9. As shown in Figure 9, LS7 randomly selects two segments of equal-length sequences from a solution and then mixes them by inserting the elements one by one. LS8 randomly selects and

inserts it into a random position. By LS9, a segment of sequence with a random length is selected, and then a reverse operation is executed on it. After that, it is inserted into a random position of the original solution.

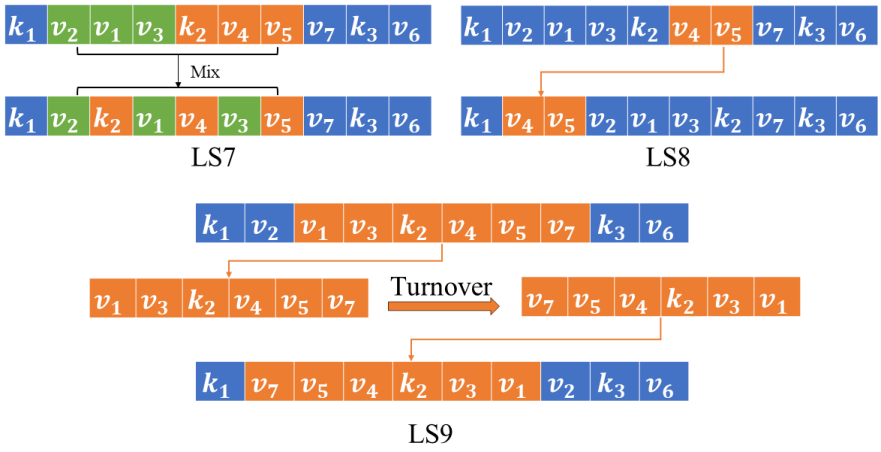

Figure 9. Example of LS7, LS8, and LS9.

4.5. Q-Learning

Q-learning is a kind of reinforcement learning algorithm. As a method of decision-making, it evaluates an agent's behavior based on feedback from the environment and stores the evaluation value of each action. The agent forms experience by constantly interacting with the environment. The agent gains experience, chooses more appropriate actions, and eventually reaches a decision that is closer to the global optimum. In the Q-table, Q-value records the impact of different actions on the long-term reward under different states. The optimal Q-value in the Q-table determines the selected action at each time. The update formula of the Q-value is as follows.

$$Q_{(s_t,a_t)} \leftarrow Q_{(s_t,a_t)} + \alpha\left[R_t + \gamma \max_a Q(s_{t+1},a_{t+1}) - Q_{(s_t,a_t)}\right], \tag{22}$$

where s_t and a_t represent the state and action of an agent at step t; R_t is the reward obtained by the agent executing action a_t; s_{t+1} denotes the state at step $t+1$, while $\max_a Q(s_{t+1},a_{t+1})$ represents the maximum Q-value corresponding to the actions under state s_{t+1}; α means the learning rate, while γ represents the discount factor.

After each interaction with the environment, the agent adjusts the Q-value of the current state-action pair according to the observed reward and the maximum Q-value of the next state. By repeating this process continuously, the agent can gradually approach the optimal Q-value function and choose the optimal behavior accordingly.

4.6. Q-Learning-Based Local Search

4.6.1. The First Q-Learning-Based Local Search (QL1)

By QL1, the states in the Q-table are set to be the task load ratios that match the ship type. A group of four numbers represents the task load ratios of the ship cluster. The task load ratios are divided into zero matching, low proportion matching, medium proportion matching, and high proportion matching. They represent that the number of tasks assigned to a ship that matches the ship type accounts for 0%, 1–33.33%, 33.34–66.66%, and 66.67–100% of their total loads, respectively. As shown in Figure 10, state S = [0, 1, 1, 2] means that the numbers of USVs under four ratio states are 0, 1, 1, 2. The actions in the Q-table are set to nine local search operators, as shown in Table 2.

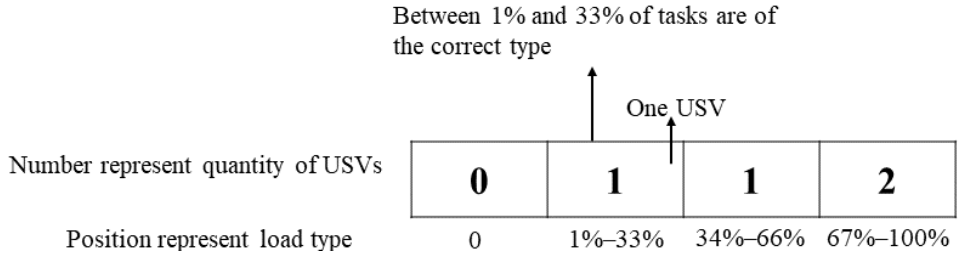

Figure 10. An example of the state for QL1.

Table 2. The Q-table of QL1.

S/A	O_{p1}	O_{p2}	O_{p3}	...	O_{p8}	O_{p9}
SL_1	$Q_{(1,1)}$	$Q_{(1,2)}$	$Q_{(1,3)}$	...	$Q_{(1,8)}$	$Q_{(1,9)}$
SL_2	$Q_{(2,1)}$	$Q_{(2,2)}$	$Q_{(2,3)}$	...	$Q_{(2,8)}$	$Q_{(2,9)}$
SL_3	$Q_{(3,1)}$	$Q_{(3,2)}$	$Q_{(3,3)}$	...	$Q_{(3,8)}$	$Q_{(3,9)}$
SL_4	$Q_{(4,1)}$	$Q_{(4,2)}$	$Q_{(4,3)}$	...	$Q_{(4,8)}$	$Q_{(4,9)}$

In the initial Q-table, all Q-values are set to 100, and all actions have an equal probability of being selected. After executing an action, the Q-value is updated according to the following formula:

$$R = \theta(fitness_{current} - fitness_{new}), \tag{23}$$

where θ is the discount rate, $fitness_{current}$ and $fitness_{new}$ represent the fitness values of the current solution and the new one, respectively.

4.6.2. The Second Q-Learning-Based Local Search (QL2)

By QL2, the states in the Q-table are set to be the loads of USVs in the ship cluster. Similar to QL1, QL2 uses four numbers to represent a state. As shown in Figure 11, each value represents the number of USVs and its position denotes the USVs' workload. The workloads are described as 0–25%, 26–50%, 51–75%, and 76–100%, respectively. In a state S = [2,1,1,0], each value means the number of USVs under the corresponding task load.

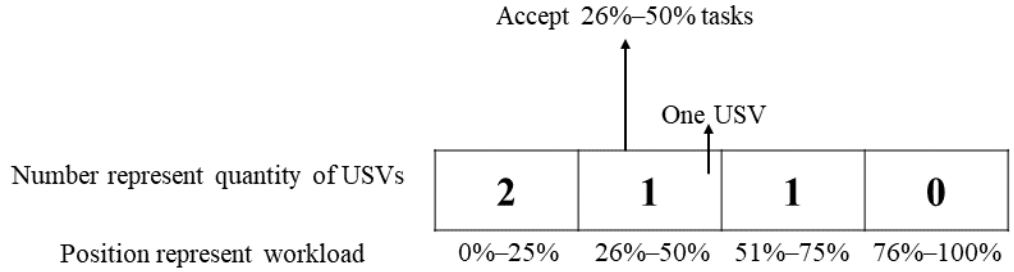

Figure 11. An example of the state for QL2.

Similar to QL1, the actions of QL2 are the nine local search operators. In the initial Q-table, all Q-values are also set to 100, and the initial probability of selecting each local search operator is the same.

4.6.3. The Third Q-Learning-Based Local Search (QL3)

By QL3, a general strategy is adopted without designing a special state identifier for the problem. The nine local search operators are treated as both states and actions. It focuses on optimizing the execution order of neighborhood search operators. As shown in Table 3, both the states and actions are the nine local search operators. For example, if the current state is O_{p1}, an action is executed in the last iteration. If O_{p8} is chosen as an action currently, after executing it, the state is changed to O_{p8}.

Table 3. The Q-table of QL3.

S/A	O_{p1}	O_{p2}	O_{p3}	...	O_{p8}	O_{p9}
O_{p1}	$Q_{(1,1)}$	$Q_{(1,2)}$	$Q_{(1,3)}$	...	$Q_{(1,8)}$	$Q_{(1,9)}$
O_{p2}	$Q_{(2,1)}$	$Q_{(2,2)}$	$Q_{(2,3)}$	...	$Q_{(2,8)}$	$Q_{(2,9)}$
O_{p3}	$Q_{(3,1)}$	$Q_{(3,2)}$	$Q_{(3,3)}$	...	$Q_{(3,8)}$	$Q_{(3,9)}$
...	...	...	...	...	...	...
O_{p8}	$Q_{(8,1)}$	$Q_{(8,2)}$	$Q_{(8,3)}$	...	$Q_{(8,8)}$	$Q_{(8,9)}$
O_{p9}	$Q_{(9,1)}$	$Q_{(9,2)}$	$Q_{(9,3)}$	...	$Q_{(9,8)}$	$Q_{(9,9)}$

4.7. The Framework of the Proposed Algorithms

This study proposes three Q-learning strategies to guide local search selection, which are embedded into three meta-heuristics: GA, SA, and HS. The framework of the proposed algorithms is shown in Figure 12. First, the population is initialized by the solutions generated in Section 4.2, and then the initial solutions are evaluated. Second, new solutions are generated by the algorithm-specific strategies. Then, the QL-based local search strategies in Section 4.6 further optimize the solutions by meta-heuristics. Finally, it repeats the whole iteration process until the termination condition is reached, and then outputs the best results.

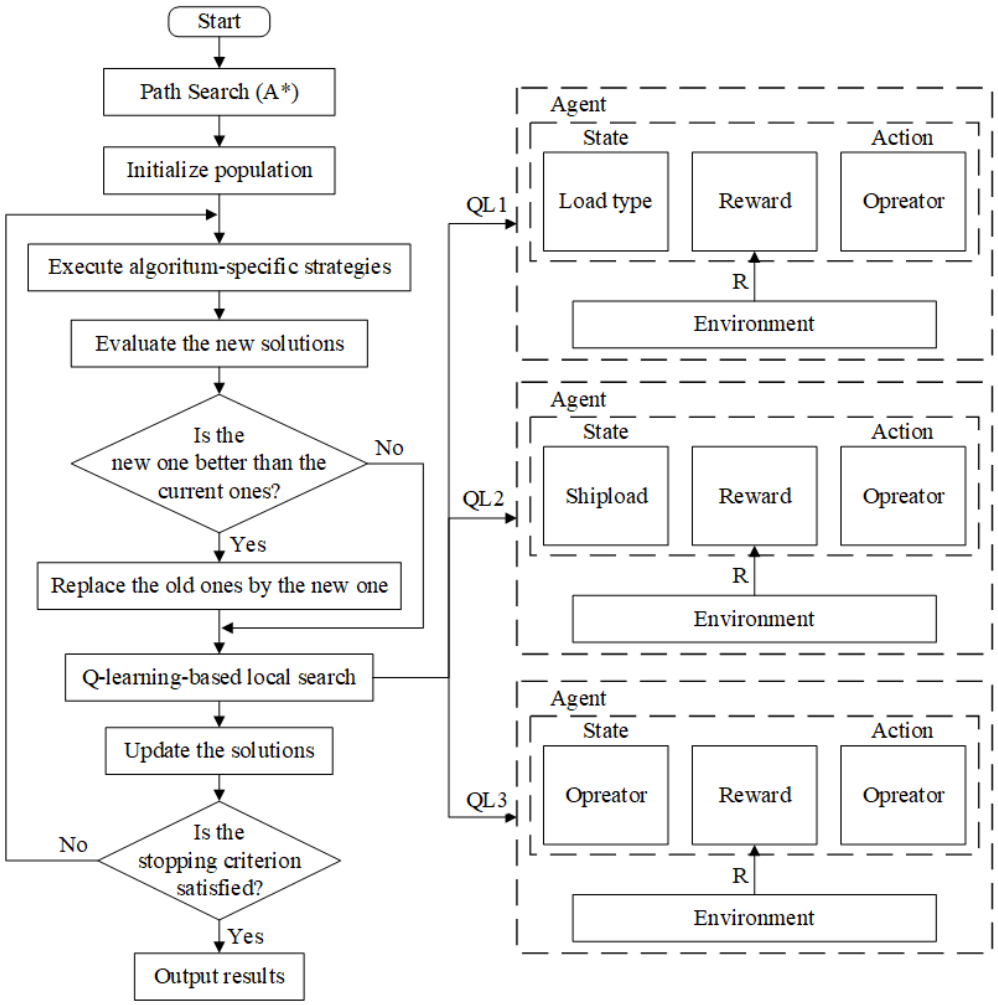

Figure 12. The flow chart of the proposed algorithms.

5. Experiments and Discussion

5.1. Experimental Setup

To evaluate the effectiveness of the improved algorithms, we solve 13 cases with different scales from a USV company in China. The number of USVs is changed from 2 to 8, while the number of tasks is changed from 20 to 120, and there are 3 types of USVs and tasks included in each case. There are few publications used to solve the heterogeneous USVs scheduling problems with obstacle avoidance. In this study, we compare three classical meta-heuristics and their 12 variants, including different local search operators and Q-learning-based local search selection strategies. All algorithms are coded in Python and run on a computer equipped with Intel® Core™-12400 @2.50 GHz with 32.0 GB of memory. To make a fair comparison, each instance is run independently 30 times and the run time is set as follows:

$$Time = 0.45 \times (n_{Task} + 0.4 \times n_{USV}), \tag{24}$$

where n_{Task} is the number of tasks while n_{USV} represents the number of USVs. The Q-tables in the three Q-learning strategies are updated online. The parameters' setting of Q-learning and meta-heuristics are shown in Tables 4 and 5.

Table 4. Parameter setting of the meta-heuristics.

Meta-Heuristics	Parameters	Value
GA	Crossover rate	0.8
	Mutation rate	0.1
SA	Start temp	100
	Temperature drop coefficient	0.96
HS	Harmony memory considering rate	0.7
	Pitch adjusting rate	0.5

Table 5. Parameter setting of Q-learning.

Q-Learning	Value
Penalry learning rate	0.6
Discount rate θ	0.8
Reward learning rate	1

5.2. Effectiveness of Proposed Strategies

In this sub-section, three meta-heuristics (SA, GA, and HS) and their variants based on the local search operators (SA + LS, GA + LS, and HS + LS) and the variants using Q-learning-based strategies to guide the local search (SA + QL1, SA + QL2, SA + QL3, GA + QL1, GA + QL2, GA + QL3, HS + QL1, HS + QL2, and HS + QL3) are evaluated and compared.

Tables 6–8 show the average fitness and CV values obtained by all the compared algorithms for the 13 cases. The CV values are calculated using the following formula:

$$\mathrm{CV} = \frac{S}{M} \times 100\%, \tag{25}$$

where S is the standard deviation and M is the mean of the results obtained in 30 runs of a case by one algorithm. The best results are shown in bold. We find that the meta-heuristics with Q-learning-based local search strategies perform better than others in their respective groups.

Table 6. Statistical results by SA and its variants.

Instance	SA		SA + LS		SA + QL1		SA + QL2		SA + QL3	
	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV
2 × 20	682.56	2.01	686.38	1.79	685.04	1.72	688.43	1.66	**682.05**	2.15
2 × 40	1377.62	1.18	1378.15	1.18	1376.47	0.91	**1375.11**	1.00	1376.43	1.21
2 × 80	3054.90	1.22	3070.32	0.69	3062.77	1.17	3055.89	1.13	**3054.50**	0.88
4 × 20	354.29	2.17	359.43	1.81	359.62	2.01	358.14	2.38	**353.59**	1.62
4 × 40	706.55	1.75	708.31	1.97	707.71	1.22	709.55	1.44	**695.10**	1.61
4 × 80	1244.87	1.33	1241.23	1.61	1245.89	1.47	1245.72	1.30	**1221.78**	1.07
6 × 20	213.75	2.88	214.82	4.42	216.54	3.07	214.51	2.90	**211.37**	3.38
6 × 40	477.15	3.50	469.68	2.59	471.73	3.36	477.29	2.43	**456.72**	3.70
6 × 80	844.30	3.36	842.46	3.14	837.26	2.47	848.27	2.30	**805.19**	2.07
8 × 20	132.22	2.46	131.85	4.57	132.02	2.91	132.72	2.69	**128.50**	3.66
8 × 40	387.19	2.60	383.98	3.70	383.00	3.84	378.66	3.89	**364.40**	4.35
8 × 80	681.51	3.17	673.10	3.75	672.00	3.54	676.35	2.73	**642.21**	3.50
8 × 120	1026.49	2.51	1005.34	2.66	998.29	4.31	1008.36	3.89	**937.87**	3.35

Table 7. Statistical results by GA and its variants.

Instance	GA		GA + LS		GA + QL1		GA + QL2		GA + QL3	
	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV
2 × 20	**681.31**	1.53	681.81	1.75	682.33	1.89	686.36	1.49	683.02	1.59
2 × 40	1396.66	1.73	**1391.86**	1.66	1402.06	1.65	1393.81	1.56	1395.31	1.65
2 × 80	3100.41	0.91	3107.03	0.76	3099.52	1.07	**3087.55**	1.15	3097.92	1.22
4 × 20	367.87	6.15	353.59	2.20	356.22	2.10	353.95	2.77	**350.81**	2.31
4 × 40	707.01	3.63	699.45	1.39	698.34	1.44	**697.18**	1.49	698.73	1.56
4 × 80	1259.42	4.92	1239.40	1.03	1234.65	1.40	1237.09	1.31	**1229.75**	1.38
6 × 20	223.21	3.75	208.39	2.64	207.92	2.76	209.18	2.65	209.21	2.80
6 × 40	483.34	3.78	**460.23**	3.60	464.08	2.34	463.06	2.49	461.32	3.10
6 × 80	831.63	3.29	826.40	1.91	824.48	2.65	830.01	2.27	**822.23**	2.51
8 × 20	133.54	2.66	127.32	3.50	128.15	2.91	127.52	3.18	**126.67**	3.43
8 × 40	386.96	4.10	370.47	2.23	**365.92**	2.36	366.31	2.93	369.11	2.69
8 × 80	656.56	4.87	655.87	2.44	651.78	1.81	**647.76**	2.80	650.32	2.98
8 × 120	985.92	3.52	977.55	2.21	968.87	2.05	972.27	2.58	**963.55**	2.73

Table 8. Statistical results by HS and its variants.

Instance	HS		HS + LS		HS + QL1		HS + QL2		HS + QL3	
	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV	Avg.	CV
2 × 20	702.96	2.36	684.33	1.70	**663.20**	1.96	677.04	1.95	669.04	2.25
2 × 40	1398.19	1.38	1373.60	0.79	**1336.33**	0.94	1349.45	1.28	1341.78	1.36
2 × 80	3102.60	1.24	3045.14	1.28	**2949.68**	1.23	2989.51	1.16	2977.13	1.07
4 × 20	373.17	2.97	352.02	2.28	343.97	2.09	341.99	2.48	**341.05**	1.74
4 × 40	721.68	2.88	688.38	1.47	673.16	2.05	673.01	1.42	**672.82**	1.86
4 × 80	1270.32	2.07	1212.95	0.98	**1177.59**	1.54	1177.71	1.41	1177.81	1.36
6 × 20	226.63	3.64	205.36	3.18	202.59	3.27	**197.89**	2.45	198.62	1.72
6 × 40	493.85	3.48	440.35	2.92	434.87	3.04	**419.50**	2.16	421.35	2.72
6 × 80	865.68	3.53	796.13	1.70	771.60	1.28	**757.34**	1.57	759.34	1.56
8 × 20	138.32	4.59	125.50	3.26	123.95	3.73	**117.81**	4.64	120.54	3.41
8 × 40	402.64	4.76	353.35	2.77	345.56	2.71	**327.55**	2.29	331.19	2.38
8 × 80	697.09	4.25	612.07	3.12	596.13	1.98	580.05	1.62	**579.33**	1.75
8 × 120	1043.27	4.67	907.46	1.35	860.34	2.47	**844.76**	1.95	845.74	1.48

To further validate the effectiveness of the local search operators and Q-learning strategies, we execute the Friedman test on meta-heuristics (SA, GA, and HS) and their variants, respectively. The results are presented in Figures A1–A7 in Appendix A. After

analyzing the results, we can conclude that the three algorithms, SA + QL3, GA + QL1, and HS + QL2 achieved better performance than their peers in the respective groups. It can be concluded that the proposed Q-learning guided local search strategies can improve the performance of the basic meta-heuristics and their variants with randomized local search strategy.

5.3. Statistical Test

To further analyze the performance difference among SA + QL3, GA + QL3, and HS + QL2. The Friedman test is executed on their results for 13 cases, as shown in Figure 13. The asymptotic significance (Asympt. Sig.) is much less than 0.05. This means that there is a significant performance difference among the three algorithms. The three algorithms are ranked using the Nemenyi post hoc test, and the results are shown in Figure 14. In this test, the algorithms with smaller average ranking values have better performance. The average ranking value (1.00) of HS + QL2 is lower compared to the ones by SA + QL3 and GA + QL3. The two-way analysis of variance by rank for the three algorithms is given in Figure 15. This test shows the distribution of rankings achieved by the algorithms for thirteen examples. It can be observed that the HS + QL2 achieves the best results for most cases. Hence, we can conclude that the HS + QL2 is more competitive than its peers.

Test Statistics

N	13
Chi-Squre	21.385
df	2
Asymp.Sig.	0.000

Figure 13. The test statistics of the three algorithms.

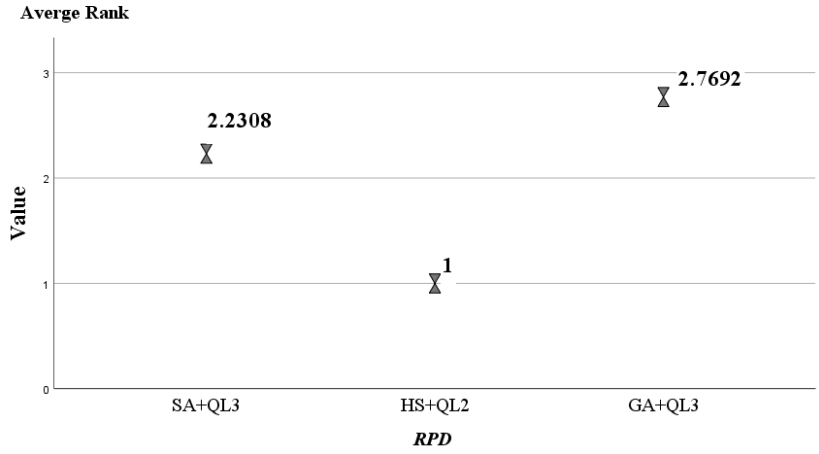

Figure 14. The Nemenyi post hoc test of the three algorithms.

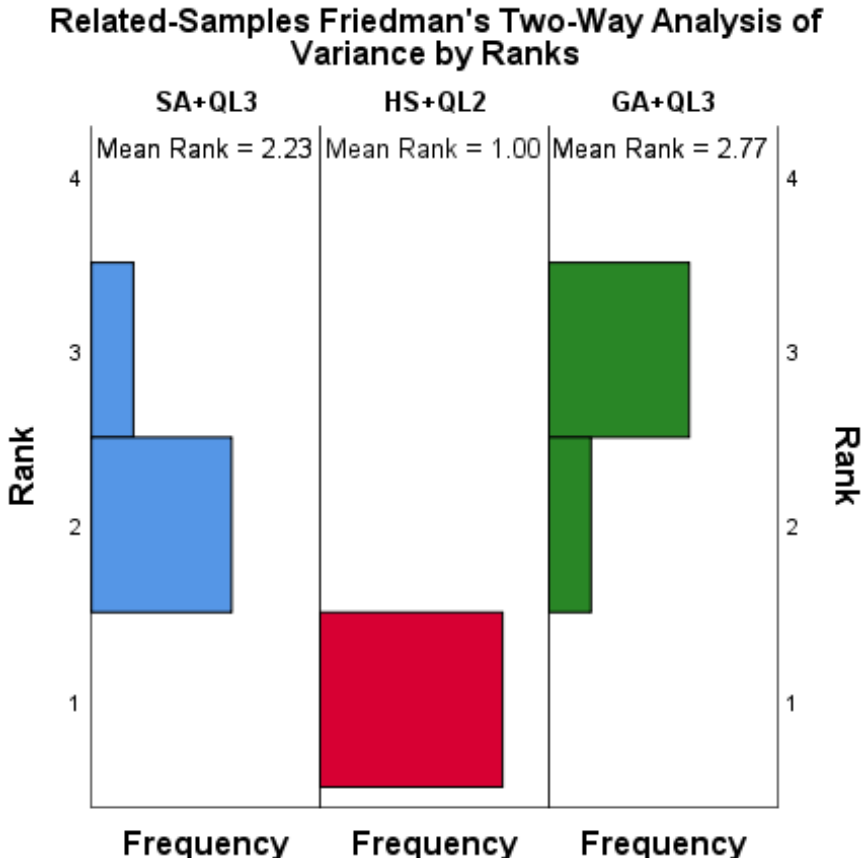

Figure 15. The Rank distribution of the three algorithms.

To present the convergence of the three algorithms clearly, Figure 16 shows the convergence curves of them for the largest instance, "8 × 120". The convergence curves of another case are shown in Figures A8–A11 in Appendix B. It can be seen from the figure that the convergence of HS + QL2 performs the best among the three algorithms, and its final result is also better than those of its peers.

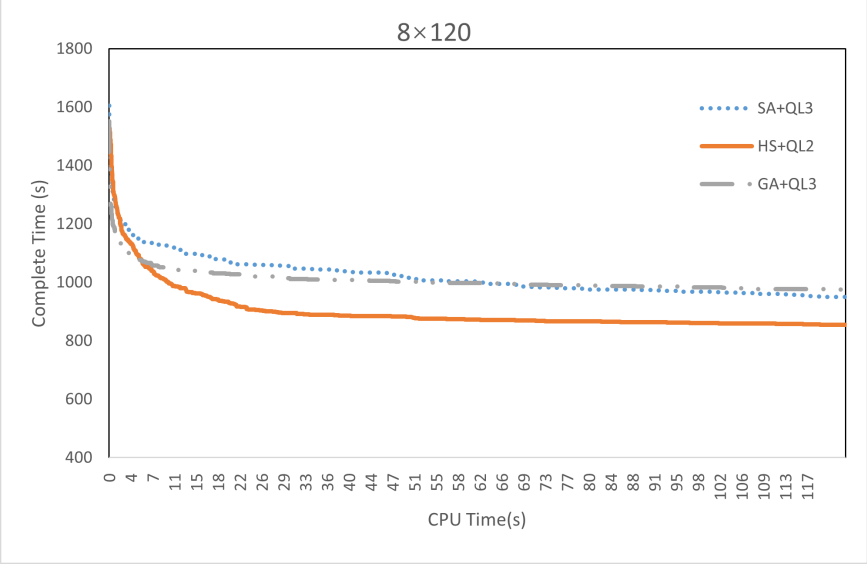

Figure 16. The convergence curves of the three algorithms for the largest case.

5.4. Compare with Existing Algorithms

The SA + QL3, GA + QL3, and HS + QL2 are compare with PSO_LS and PSO_QL in [20]. Table 9 shows the average fitness values obtained by the five algorithms. From Table 9, it can be observed that the results by SA + QL3, GA + QL3, and HS + QL2 are better than those by PSO_LS and PSO_QL. The HS + QL2 obtains the best average values for all instances. It can be concluded that HS + QL2 has the best performance for solving the heterogeneous USV scheduling problems.

Table 9. Comparison of average fitness values for five meta-heuristics.

Instance	PSO_LS Avg.	PSO_QL Avg.	SA + QL3 Avg.	GA + QL3 Avg.	HS + QL2 Avg.
2 × 20	748.50	792.39	682.05	683.02	**677.04**
2 × 40	1466.23	1548.17	1376.43	1395.31	**1349.45**
2 × 80	3076.76	3063.13	3054.50	3097.92	**2989.51**
4 × 20	447.30	411.03	353.59	350.81	**341.99**
4 × 40	720.39	735.50	695.10	698.73	**673.01**
4 × 80	1382.46	1278.32	1221.78	1229.75	**1177.71**
8 × 20	170.70	153.98	128.50	126.67	**117.81**
8 × 40	528.98	582.02	364.40	369.11	**327.55**
8 × 80	973.89	1074.39	642.21	650.32	**580.05**
8 × 120	1562.90	1579.68	937.87	963.55	**844.76**

6. Conclusions and Future Work

In this study, Q-learning is employed to guide three meta-heuristics to select the appropriate local search operators for solving the heterogeneous USV scheduling problems. Based on the nature of the problem, we propose nine local search operators and design three different Q-learning strategies. The performance of the proposed local search operators and the Q-learning strategies are verified by solving 13 instances with different scales. We have compared the proposed meta-heuristics with Q-learning strategies to the existing algorithms. The experimental results and analysis suggest that the proposed algorithms perform better than the peers.

The future research directions are as follows:

(1) consider more objectives such as energy consumption, carbon emission, and safety;
(2) design more approaches to integrate meta-heuristics and reinforcement learning algorithms;
(3) extend the algorithms to solve more USV scheduling and optimization problems.

Author Contributions: Conceptualization, Z.M.; methodology, Z.M., K.G. and H.Y.; writing—original draft preparation, Z.M. and H.Y.; writing, review, and editing, K.G. and N.W.; supervision, K.G. All authors have read and agreed to the published version of the manuscript.

Funding: This study is partially supported by the Zhuhai Industry–University–Research Project with Hongkong and Macao under Grant ZH22017002210014PWC, the National Natural Science Foundation of China under Grant 62173356, the Science and Technology Development Fund (FDCT), Macau SAR, under Grant 0019/2021/A, the Guangdong Basic and Applied Basic Research Foundation (2023A1515011531), and research on the Key Technologies for Scheduling and Optimization of Complex Distributed Manufacturing Systems (22JR10KA007).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

Appendix A

The Friedman test is performed to verify the improvement of the new strategies over the meta-heuristics. From Figure A1, it can be seen that the asymptotic significance

(Asympt. Sig.) for the three basic algorithms with their variants is much less than 0.05. It means that the basic algorithms are significantly different from their improved variants.

These algorithms were ranked using the Nemenyi post hoc test, and the resulting rankings are shown in Figures A2–A4. In the three groups, the three algorithms, SA + QL3, GA + QL3, and HS + QL2 have the smallest average ranking values (1.08, 2.15, and 1.77), respectively. Friedmann's two-way analysis of variance by rank gives a clearer presentation of the instance distribution for the compared algorithms, as shown in Figures A5–A7. It can be seen that SA + QL3, GA + QL3, and HS + QL2 achieve the best results in most cases.

Test Statistics

N	13
Chi-Squre	24.677
df	4
Asymp.Sig.	0.000

SA and its variants

Test Statistics

N	13
Chi-Squre	17.108
df	4
Asymp.Sig.	0.002

GA and its variants

Test Statistics

N	13
Chi-Squre	42.769
df	4
Asymp.Sig.	0.000

HS and its variants

Figure A1. Friedman Test of meta-heuristics and their variants.

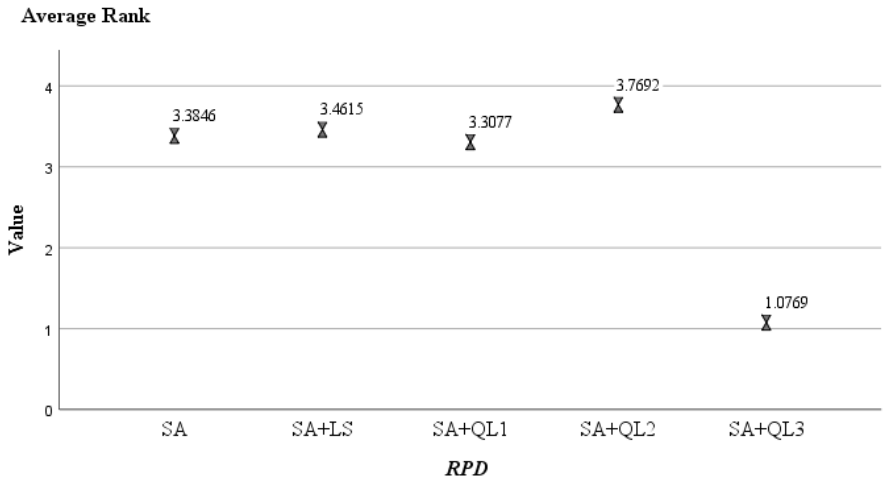

Figure A2. The Nemenyi post hoc test of SA and their variants.

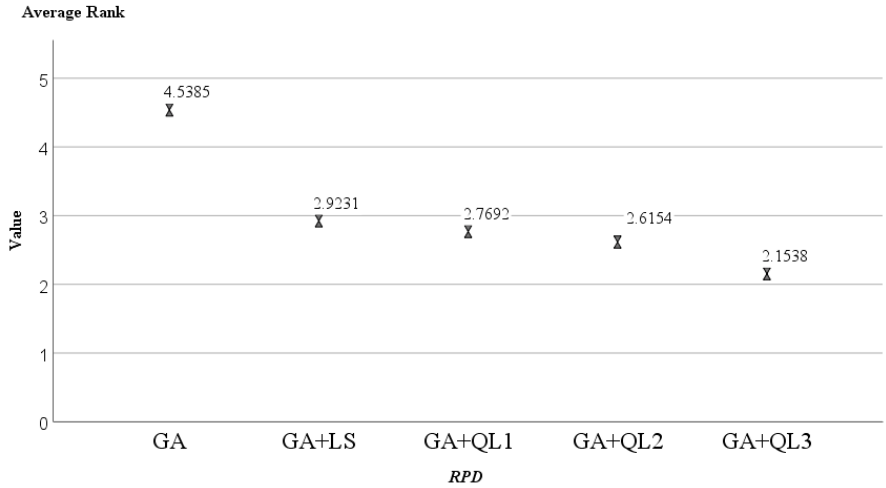

Figure A3. The Nemenyi post hoc test of GA and their variants.

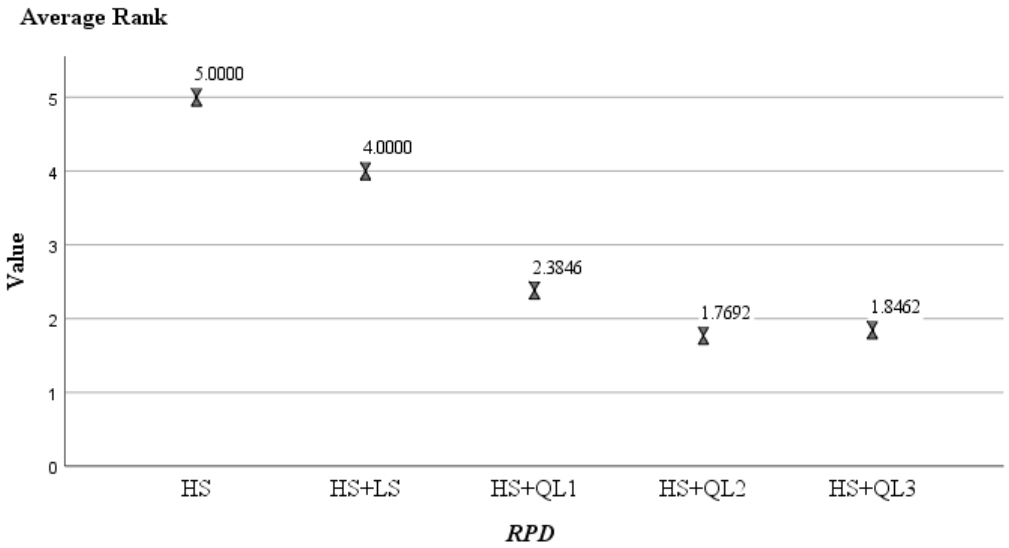

Figure A4. The Nemenyi post hoc test of HS and their variants.

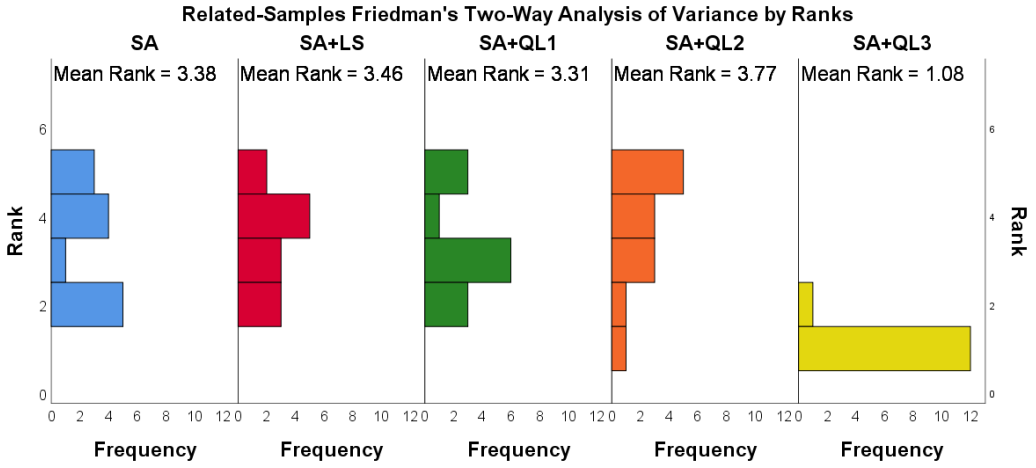

Figure A5. The Rank distribution of SA and their variants.

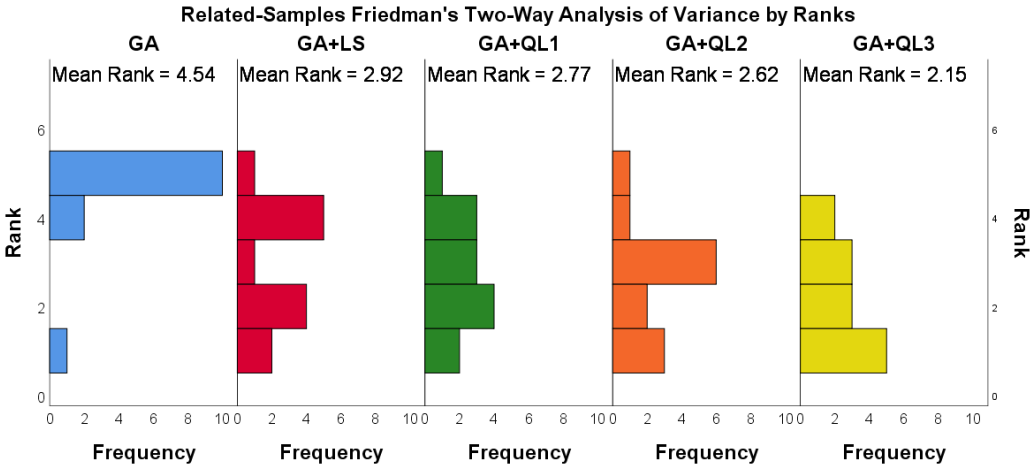

Figure A6. The Rank distribution of GA and their variants.

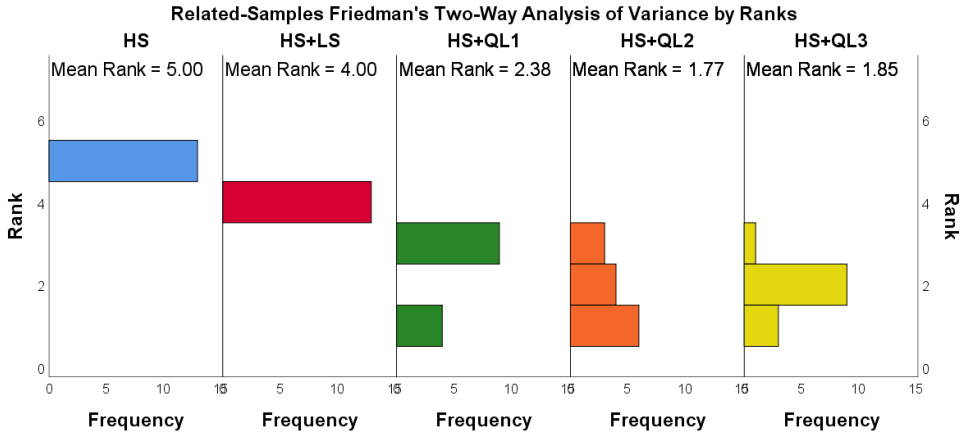

Figure A7. The Rank distribution of HS and their variants.

Appendix B

As shown in Figures A8–A11, we provide the convergence curves for the cases corresponding to the maximum number of tasks with two-USVs, four-USVs, six-USVs, and eight-USVs.

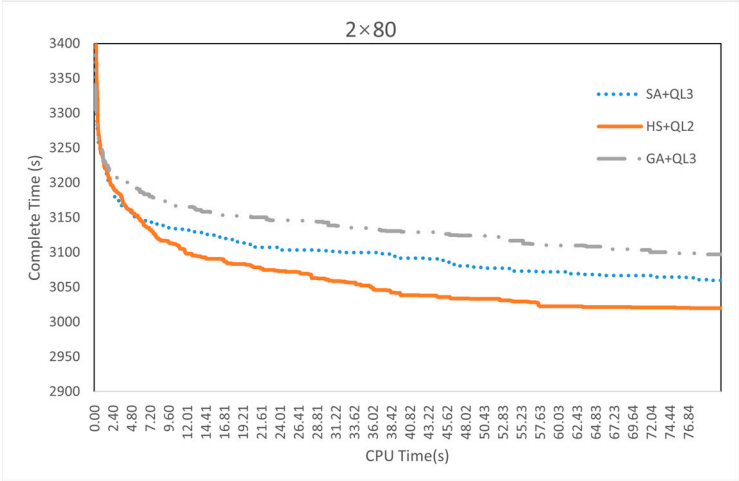

Figure A8. The convergence curves of the three algorithms for case "2 × 80".

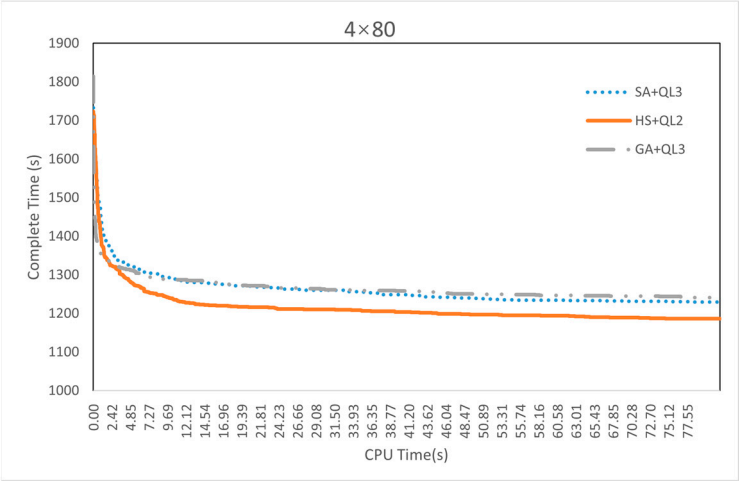

Figure A9. The convergence curves of the three algorithms for case "4 × 80".

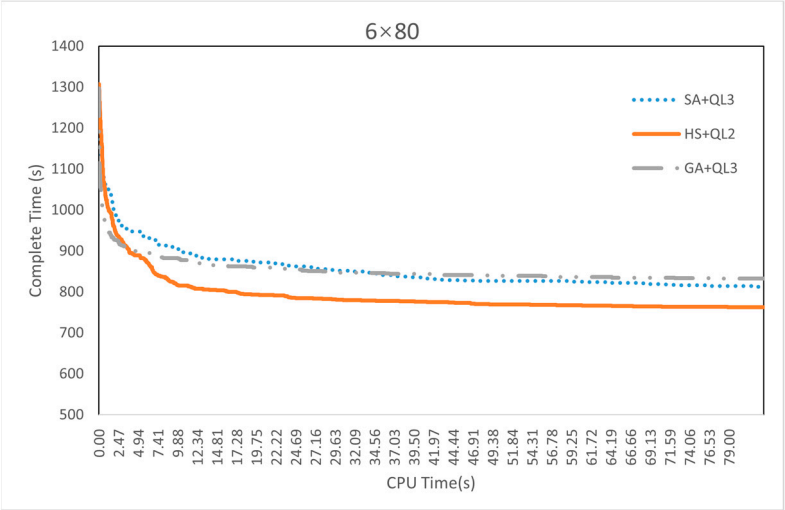

Figure A10. The convergence curves of the three algorithms for case "6 × 80".

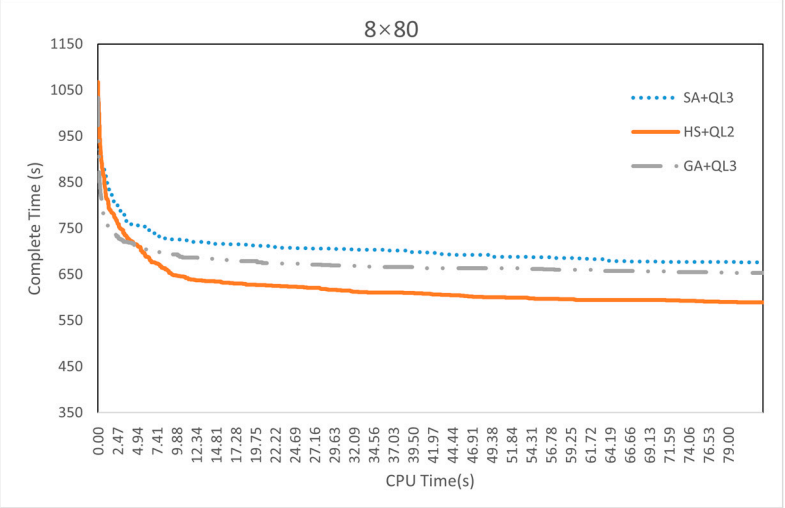

Figure A11. The convergence curves of the three algorithms for case "8 × 80".

References

1. Yuh, J.; Marani, G.; Blidberg, D.R. Applications of Marine Robotic Vehicles. *Intel. Serv. Robot.* **2011**, *4*, 221–231. [CrossRef]
2. Xingfa, G.; Xudong, T. Overview of China Earth Observation Satellite Programs [Space Agencies]. *IEEE Geosci. Remote Sens. Mag.* **2015**, *3*, 113–129. [CrossRef]
3. McCarthy, M.J.; Colna, K.E.; El-Mezayen, M.M.; Laureano-Rosario, A.E.; Méndez-Lázaro, P.; Otis, D.B.; Toro-Farmer, G.; Vega-Rodriguez, M.; Muller-Karger, F.E. Satellite Remote Sensing for Coastal Management: A Review of Successful Applications. *Environ. Manag.* **2017**, *60*, 323–339. [CrossRef] [PubMed]
4. Yang, L.; Zhao, S.; Wang, X.; Shen, P.; Zhang, T. Deep-Sea Underwater Cooperative Operation of Manned/Unmanned Submersible and Surface Vehicles for Different Application Scenarios. *JMSE* **2022**, *10*, 909. [CrossRef]
5. Sotelo-Torres, F.; Alvarez, L.V.; Roberts, R.C. An Unmanned Surface Vehicle (USV): Development of an Autonomous Boat with a Sensor Integration System for Bathymetric Surveys. *Sensors* **2023**, *23*, 4420. [CrossRef] [PubMed]
6. Ren, R.-Y.; Zou, Z.-J.; Wang, Y.-D.; Wang, X.-G. Adaptive Nomoto Model Used in the Path Following Problem of Ships. *J. Mar. Sci. Technol.* **2018**, *23*, 888–898. [CrossRef]
7. Xie, J.; Zhou, R.; Luo, J.; Peng, Y.; Liu, Y.; Xie, S.; Pu, H. Hybrid Partition-Based Patrolling Scheme for Maritime Area Patrol with Multiple Cooperative Unmanned Surface Vehicles. *J. Mar. Sci. Eng.* **2020**, *8*, 936. [CrossRef]
8. Huang, L.; Zhou, M.; Hao, K. Non-Dominated Immune-Endocrine Short Feedback Algorithm for Multi-Robot Maritime Patrolling. *IEEE Trans. Intell. Transport. Syst.* **2020**, *21*, 362–373. [CrossRef]
9. Sutton, A.J.; Williams, N.L.; Tilbrook, B. Constraining Southern Ocean CO_2 Flux Uncertainty Using Uncrewed Surface Vehicle Observations. *Geophys. Res. Lett.* **2021**, *48*, e2020GL091748. [CrossRef]

10. Cryer, S.; Carvalho, F.; Wood, T.; Strong, J.A.; Brown, P.; Loucaides, S.; Young, A.; Sanders, R.; Evans, C. Evaluating the Sensor-Equipped Autonomous Surface Vehicle C-Worker 4 as a Tool for Identifying Coastal Ocean Acidification and Changes in Carbonate Chemistry. *J. Mar. Sci. Eng.* **2020**, *8*, 939. [CrossRef]
11. Sinisterra, A.J.; Dhanak, M.R.; Von Ellenrieder, K. Stereovision-Based Target Tracking System for USV Operations. *Ocean Eng.* **2017**, *133*, 197–214. [CrossRef]
12. Shao, G.; Ma, Y.; Malekian, R.; Yan, X.; Li, Z. A Novel Cooperative Platform Design for Coupled USV–UAV Systems. *IEEE Trans. Ind. Inf.* **2019**, *15*, 4913–4922. [CrossRef]
13. Zafar, M.N.; Mohanta, J.C. Methodology for Path Planning and Optimization of Mobile Robots: A Review. *Procedia Comput. Sci.* **2018**, *133*, 141–152. [CrossRef]
14. Yu, K.; Liang, X.; Li, M.; Chen, Z.; Yao, Y.; Li, X.; Zhao, Z.; Teng, Y. USV Path Planning Method with Velocity Variation and Global Optimisation Based on AIS Service Platform. *Ocean Eng.* **2021**, *236*, 109560. [CrossRef]
15. Xiaofei, Y.; Yilun, S.; Wei, L.; Hui, Y.; Weibo, Z.; Zhengrong, X. Global Path Planning Algorithm Based on Double DQN for Multi-Tasks Amphibious Unmanned Surface Vehicle. *Ocean Eng.* **2022**, *266*, 112809. [CrossRef]
16. Tsai, C.-C.; Huang, H.-C.; Chan, C.-K. Parallel Elite Genetic Algorithm and Its Application to Global Path Planning for Autonomous Robot Navigation. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4813–4821. [CrossRef]
17. Yin, J.; Hu, Z.; Mourelatos, Z.P.; Gorsich, D.; Singh, A.; Tau, S. Efficient Reliability-Based Path Planning of Off-Road Autonomous Ground Vehicles Through the Coupling of Surrogate Modeling and RRT. *IEEE Trans. Intell. Transport. Syst.* **2023**, *43*, 15035–15050. [CrossRef]
18. Gao, K.; Zhang, Y.; Su, R.; Yang, F.; Suganthan, P.N.; Zhou, M. Solving Traffic Signal Scheduling Problems in Heterogeneous Traffic Network by Using Meta-Heuristics. *IEEE Trans. Intell. Transport. Syst.* **2019**, *20*, 3272–3282. [CrossRef]
19. Kuo, I.-H.; Horng, S.-J.; Kao, T.-W.; Lin, T.-L.; Lee, C.-L.; Terano, T.; Pan, Y. An Efficient Flow-Shop Scheduling Algorithm Based on a Hybrid Particle Swarm Optimization Model. *Expert Syst. Appl.* **2009**, *36*, 7027–7032. [CrossRef]
20. Gao, M.; Gao, K.; Ma, Z.; Tang, W. Ensemble Meta-Heuristics and Q-Learning for Solving Unmanned Surface Vessels Scheduling Problems. *Swarm Evol. Comput.* **2023**, *82*, 101358. [CrossRef]
21. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path Planning Techniques for Mobile Robots: Review and Prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [CrossRef]
22. Gemeinder, M.; Gerke, M. GA-Based Path Planning for Mobile Robot Systems Employing an Active Search Algorithm. *Appl. Soft Comput.* **2003**, *3*, 149–158. [CrossRef]
23. Roberge, V.; Tarbouchi, M.; Labonte, G. Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning. *IEEE Trans. Ind. Inf.* **2013**, *9*, 132–141. [CrossRef]
24. Nazarahari, M.; Khanmirza, E.; Doostie, S. Multi-Objective Multi-Robot Path Planning in Continuous Environment Using an Enhanced Genetic Algorithm. *Expert Syst. Appl.* **2019**, *115*, 106–120. [CrossRef]
25. Miao, H.; Tian, Y.-C. Dynamic Robot Path Planning Using an Enhanced Simulated Annealing Approach. *Appl. Math. Comput.* **2013**, *222*, 420–437. [CrossRef]
26. Huo, L.; Zhu, J.; Wu, G.; Li, Z. A Novel Simulated Annealing Based Strategy for Balanced UAV Task Assignment and Path Planning. *Sensors* **2020**, *20*, 4769. [CrossRef] [PubMed]
27. Xiao, S.; Tan, X.; Wang, J. A Simulated Annealing Algorithm and Grid Map-Based UAV Coverage Path Planning Method for 3D Reconstruction. *Electronics* **2021**, *10*, 853. [CrossRef]
28. Zhao, F.; Zhou, G.; Wang, L. A Cooperative Scatter Search with Reinforcement Learning Mechanism for the Distributed Permutation Flowshop Scheduling Problem with Sequence-Dependent Setup Times. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 4899–4911. [CrossRef]
29. Zhao, F.; Wang, Q.; Wang, L. An Inverse Reinforcement Learning Framework with the Q-Learning Mechanism for the Metaheuristic Algorithm. *Knowl. Based Syst.* **2023**, *265*, 110368. [CrossRef]
30. Zhao, F.; Wang, Z.; Wang, L. A Reinforcement Learning Driven Artificial Bee Colony Algorithm for Distributed Heterogeneous No-Wait Flowshop Scheduling Problem with Sequence-Dependent Setup Times. *IEEE Trans. Automat. Sci. Eng.* **2022**, *20*, 2305–2320. [CrossRef]
31. Yu, H.; Gao, K.-Z.; Ma, Z.-F.; Pan, Y.-X. Improved Meta-Heuristics with Q-Learning for Solving Distributed Assembly Permutation Flowshop Scheduling Problems. *Swarm Evol. Comput.* **2023**, *80*, 101335. [CrossRef]
32. Ren, Y.; Gao, K.; Fu, Y.; Sang, H.; Li, D.; Luo, Z. A Novel Q-Learning Based Variable Neighborhood Iterative Search Algorithm for Solving Disassembly Line Scheduling Problems. *Swarm Evol. Comput.* **2023**, *80*, 101338. [CrossRef]
33. Low, E.S.; Ong, P.; Low, C.Y.; Omar, R. Modified Q-Learning with Distance Metric and Virtual Target on Path Planning of Mobile Robot. *Expert Syst. Appl.* **2022**, *199*, 117191. [CrossRef]
34. Zhao, F.; Di, S.; Wang, L. A Hyperheuristic With Q-Learning for the Multiobjective Energy-Efficient Distributed Blocking Flow Shop Scheduling Problem. *IEEE Trans. Cybern.* **2023**, *53*, 3337–3350. [CrossRef] [PubMed]
35. Maoudj, A.; Hentout, A. Optimal Path Planning Approach Based on Q-Learning Algorithm for Mobile Robots. *Appl. Soft Comput.* **2020**, *97*, 106796. [CrossRef]
36. Zhao, X.; Zong, Q.; Tian, B.; Zhang, B.; You, M. Fast Task Allocation for Heterogeneous Unmanned Aerial Vehicles through Reinforcement Learning. *Aerosp. Sci. Technol.* **2019**, *92*, 588–594. [CrossRef]

37. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage Path Planning of Heterogeneous Unmanned Aerial Vehicles Based on Ant Colony System. *Swarm Evol. Comput.* **2022**, *69*, 101005. [CrossRef]
38. Tan, G.; Zhuang, J.; Zou, J.; Wan, L. Multi-Type Task Allocation for Multiple Heterogeneous Unmanned Surface Vehicles (USVs) Based on the Self-Organizing Map. *Appl. Ocean Res.* **2022**, *126*, 103262. [CrossRef]
39. Tan, G.; Sun, H.; Du, L.; Zhuang, J.; Zou, J.; Wan, L. Coordinated Control of the Heterogeneous Unmanned Surface Vehicle Swarm Based on the Distributed Null-Space-Based Behavioral Approach. *Ocean Eng.* **2022**, *266*, 112928. [CrossRef]
40. Liu, Y.; Lin, X.; Zhang, C. Affine Formation Maneuver Control for Multi-Heterogeneous Unmanned Surface Vessels in Narrow Channel Environments. *J. Mar. Sci. Eng.* **2023**, *11*, 1811. [CrossRef]
41. Tan, G.; Zhuang, J.; Zou, J.; Wan, L. Adaptive Adjustable Fast Marching Square Method Based Path Planning for the Swarm of Heterogeneous Unmanned Surface Vehicles (USVs). *Ocean Eng.* **2023**, *268*, 113432. [CrossRef]
42. Bell, M.G.H. Hyperstar: A Multi-Path Astar Algorithm for Risk Averse Vehicle Navigation. *Transp. Res. Part B Methodol.* **2009**, *43*, 97–107. [CrossRef]
43. Gao, K.; Huang, Y.; Sadollah, A.; Wang, L. A Review of Energy-Efficient Scheduling in Intelligent Production Systems. *Complex Intell. Syst.* **2020**, *6*, 237–249. [CrossRef]
44. Gao, K.; Cao, Z.; Zhang, L.; Chen, Z.; Han, Y.; Pan, Q. A Review on Swarm Intelligence and Evolutionary Algorithms for Solving Flexible Job Shop Scheduling Problems. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 904–916. [CrossRef]