Journal of *Marine Science and Engineering*

MDPI

Article

An Exact Algorithm for Task Allocation of Multiple Unmanned Surface Vehicles with Minimum Task Time

Kai Xue, Zhiqin Huang *, Ping Wang and Zeyu Xu

College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, China; xuekai@hrbeu.edu.cn (K.X.); wangping@hrbeu.edu.cn (P.W.); xuzy19951217@hrbeu.edu.cn (Z.X.)
* Correspondence: huangzhiqin@hrbeu.edu.cn

Abstract: Task allocation of unmanned surface vehicles (USVs) with low task cost is an important research area which assigns USVs from starting points to different target points to complete tasks. Most of the research lines of task allocation are using heuristic algorithms to obtain suboptimal solutions to reduce both the max task cost and total task cost. In practice, reducing the maximum is more important to task time, which is from the departure of USVs to the last USV arriving at the designated position. In this paper, an exact algorithm is proposed to minimize the max task time and reduce the total task time based on the Hungarian algorithm. In this algorithm, task time is composed of the travel time along the planned path and the turning time at initial and target points. The fast marching square method (FMS) is used to plan the travel path with obstacle avoidance. The effectiveness and practicability of the proposed algorithm are verified by comparing it with the Hungarian algorithm (HA), the auction algorithm (AA), the genetic algorithm (GA) and the ant colony optimization algorithm (ACO). The results of path planning and task allocation are displayed in the simulation.

Keywords: exact algorithm; minimum task time; task allocation; unmanned surface vehicle

check for updates

Citation: Xue, K.; Huang, Z.; Wang, P.; Xu, Z. An Exact Algorithm for Task Allocation of Multiple Unmanned Surface Vehicles with Minimum Task Time. *J. Mar. Sci. Eng.* **2021**, *9*, 907. https://doi.org/10.3390/jmse9080907

Academic Editor: Alessandro Ridolfi

Received: 22 July 2021
Accepted: 18 August 2021
Published: 22 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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

The research on USVs has attracted increasing interest in the past decade. USVs have the advantages of low costs and large loads. They have become more and more valuable and important in many applications including environmental monitoring [1–3], unmanned cargo vessels [4], disaster relief [5] and military missions [6]. Multiple USVs can cooperate intelligently to perform tasks that individuals cannot complete in the wide sea. For commercial needs and national security, in-depth research on the multiple USVs is required, and task allocation is one of the challenges. Task allocation of multiple USVs is that a certain number of USVs at the initial points are allocated to the target points. Its optimization direction is to minimize the task cost.

Extensive research on task allocation has focused on heuristic algorithms, which are suitable for a large amount of data and are effective in satisfying different requirements at same time, such as reducing the total task cost and the max task cost. Rauniyar et al. [7] proposed a random immigrants genetic algorithm (RIGA) and an elitism-based immigrants genetic algorithm (EIGA) to optimize task allocation as the number of tasks increases in the multi-robot coalition formation problem. Liu et al. [8] proposed an intelligent hybrid multi-task allocation and path-planning algorithm. An adaptive artificial repulsive force field was constructed and integrated into the self-organizing map (SOM) to achieve collision avoidance. On this basis, Liu et al. [9] proposed a novel energy coordination scheme as well as a task prioritizing method to specifically address the two critical issues of energy consumption and communication range in the task allocation of a multi-USV system. Raboin et al. [10] presented a contract-based algorithm for allocating tasks to the USVs, which allowed decentralized and cooperative task negotiation among neighboring USVs. Nam et al. [11] employed a simulated annealing algorithm (SA) and a randomized

algorithm to reduce communication and centralized computation expense during execution by using a prior model of cost change and performing upfront computation of task allocations. Chopra et al. [12] proposed a distributed version of the Hungarian algorithm to find online suboptimal routes cooperatively. The given global criterion (e.g., the total distance traveled) was optimized within a finite set of local computations and communications. Chen et al. [13] proposed a bi-objective ant colony optimization (ACO)-based memetic algorithm to optimize the maximum traveled distance and the total traveled distance. Zhang et al. [14] proposed a vitality-driven genetic task allocation algorithm (VGTA) for multi-robot task allocations based on grid maps, which chooses the robot with the shorter time-path and reduces the waiting time. Shriyam et al. [15] proposed heuristic algorithms to effectively solve the task allocation problem between many different agents. The algorithm integrated the available information about missions and contingencies, along with the resource constraints to plan the task execution to reduce mission completion time. Nunes et al. [16] presented an auction-based method for a team of robots to allocate and execute tasks that have temporal and precedence constraints.

The optimal solution can be obtained at a certain computing time by exact algorithms. They mainly focus on reducing the total task cost, such as the Hungarian algorithm [17]. Nam et al. [18] completed a general model of the multiple-choice allocation problem and studied the multiple-choice Hungarian Algorithm by using penalization functions. Shi et al. [19] proposed a dynamic auction approach for differentiated tasks under cost rigidities (DAACR) which can obtain optimal results in the task allocation of rescue robots with minimum total costs. Lusk et al. [20] presented a distributed task allocation solution to deconflict vehicles based on the auction algorithm, which can solve the problem of the gridlock caused by known distributed collision avoidance strategies.

In practice, reducing the maximum is more important to task time which is from the departure of USVs to the last USV arriving at the designated position. In this paper, the main contributions are as follows:

A. An exact algorithm for task allocation of multiple USVs is proposed, which minimizes the max task time and reduces the total task time. It considers multi-USVs as real working applications, which are characterized by known maps, task time and collision avoidance.

B. The time through the planned path and turning time at the initial points and the target points are used to build the cost matrix of task allocation.

In the rest of this paper: Section 2 introduces the model of task allocation and the optimization direction of the algorithm. Section 3 introduces the composition of the cost matrix and provides the steps of the exact algorithm. Section 4 provides the experiment results to validate and evaluate the proposed algorithm. Section 5 concludes the paper and discusses future work.

2. Model

The task allocation problem of USVs is that n USVs at initial points are allocated to n target points. i is the serial number of the USV and j is the serial number of the target point. $\{x_{ij}\}_{n\times n}$ as an n-order square matrix of 0–1 decision variables represents the relationship between the USVs and the target points [13]. $x_{ij} = 1$ means that the USV i is allocated to the target point j. $x_{ij} = 0$ means that the USV i is not allocated to the target point j.

$$x_{ij} = 0 \text{ or } 1, i, j \in N = \{1, \ldots, n\} \tag{1}$$

In the task allocation, a certain number of USVs are allocated to the same number of target points, so it is required that each USV can only be assigned to one target point and each target point corresponds to only one USV. As shown in (2) and (3), the sum of elements in each row and each column of $\{x_{ij}\}_{n\times n}$ is equal to 1 [13].

$$\sum_{i=1}^{n} x_{ij} = 1, (j = 1, \ldots, n) \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, (i = 1, \ldots, n) \tag{3}$$

The total cost T_{tc} is the sum of the cost of each USV in an allocation plan [20], as shown in (4). c_{ij} represents the cost of the USV i to the target point j. If the USV i cannot reach the target point j, then the value of c_{ij} is positive infinity. As shown in (5), x_{mtc} is the allocation plan with the minimum total cost in all allocation plan $\{T_{tc_1}, T_{tc_2}, \ldots, T_{tc_m}\}$ [20].

$$T_{tc} = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4}$$

$$x_{mtc} = arg\,min(\{T_{tc_1}, T_{tc_2}, \ldots, T_{tc_m}\}), m = n! \tag{5}$$

The max cost T_{mc} is the largest in the cost of each USV in an allocation plan [14], as shown in (6). When the task cost refers to task time, the max time cost is from the departure of USVs to the last USV arriving at the designated position. More details of c_{ij} are given in Section 3.1. As shown in (7), x_{mmc} is the allocation plan with the minimum max cost in all allocation plan $\{T_{mc_1}, T_{mc_2}, \ldots, T_{mc_m}\}$ [14].

$$T_{mc} = \max_{i,j} \{c_{ij} x_{ij}\} \tag{6}$$

$$x_{mmc} = arg\,min(\{T_{mc_1}, T_{mc_2}, \ldots, T_{mc_m}\}), m = n! \tag{7}$$

The optimization direction of this paper is to minimize the max cost and reduce the total cost by adding (5) as a constraint based on (7). The result of task allocation with only (7) as the constraint condition may have multiple solutions, because it can only determine the relationship between one USV and one target point corresponding to the element of the max cost in the allocation plan. There is no constraint on other USVs and target points, except that their cost cannot exceed the max cost. Adding (5) as a constraint can reduce the total cost and an optimal solution can be obtained.

3. Method

3.1. Cost Matrix

In the task allocation algorithm, the cost matrix is first established, which contains the time cost from each USV to each target point. Each element composed of three parts of time in the cost matrix is given by

$$c_{ij} = T(s_{ij}) + T(\theta_{iji}) + T(\theta_{ijj}). \tag{8}$$

In (8), s_{ij} is the length of the planned path from the initial point i to the target point j. $T(s_{ij})$ is related to the speed of the USV and the length of the planned path. It is given by

$$T(s) = \sum_{a=1}^{b} \frac{||P_{a+1} - P_a||}{V_a} \tag{9}$$

where b is the number of the line segments of the planned path, P_a is the position of end points, and V_a is the speed of the USV in every line segment, which is affected by wind, wave and current.

In Figure 1, D_{ii} is the initial direction of the USV i at the initial point P_i. D_{ip} is the initial direction of the planning path at the initial point. θ_{iji} is the angle between D_{ii} and D_{ip}, and $T(\theta_{iji})$ is the adjustment time of it. D_{jr} is the requirement direction at the target point P_j. D_{ip} is the final direction of the planning path at the target point. In (8), θ_{ijj} is the angle between D_{jr} and D_{ip}, and $T(\theta_{ijj})$ is the adjustment time of it. $T(\theta)$ related to four factors is given by (10).

$$T(\theta) = \frac{p_e p_c \theta T_t}{180} \tag{10}$$

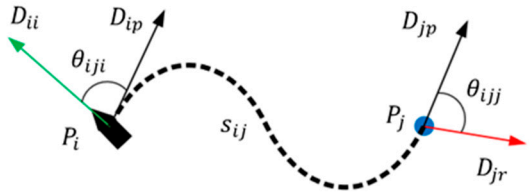

Figure 1. The time cost is affected by the length s_{ij} of the planned path, the angle θ_{iji} at the initial point and the angle θ_{ijj} at the target point.

θ is the angle of the direction change. T_t is the turning time from the start of the steering to the time when the course of USVs turns 180 degrees in the turning test. p_e is the environmental parameter related to wind, waves and currents. Their different directions and sizes affect the rotation of USVs with a low speed near the initial points and the target points. p_c is related to the motion control method of USVs. Different control methods take different amounts of time to eliminate the course deviation and the lateral deviation.

3.2. Exact Algorithm

The following steps shown in Algorithm 1 can be used to find the solution with the minimum max time and low total time.

Algorithm 1 Minimize the max cost and reduce the total cost

Input: The cost matrix $\left\{c_{ij}\right\}_{n\times n}$ based on Section 3.1.
Output: the best solution.

Step 1	Each element in the cost matrix is added with different small multi-digit random decimals which do not affect the ascending order of the elements.
Step 2	In ascending order, each element of the cost matrix is filled in the zero matrix with the same order as the cost matrix. The filling position is the same as the original position in the cost matrix.
Step 3	The rank of the matrix is detected when each element is filled into the matrix. When the matrix is just full rank, stop filling the elements, replace the zero elements in the matrix with positive infinity and obtain the solution by the Hungarian algorithm.

Remark 1. *When detecting whether the matrix is full rank, zero elements represent that they will not be selected. However, in the Hungarian algorithm, zero elements represent no cost and should be selected first. Therefore, in Step 1, the elements that represent no cost are replaced with different small random numbers which do not affect the ascending order of elements. The elements that cannot be selected are replaced by zero elements in Step 2, which are positive infinity in the calculation of the Hungarian algorithm in Step 3.*

Corollary 1. *The full rank of the matrix is a sufficient and unnecessary condition for the Hungarian algorithm to have a solution.*

Proof. Both the Hungarian algorithm and task allocation with solutions mean that there exists a solution satisfying (2) and (3). In Step 2, some elements of the cost matrix cannot be selected and there may be no solution because of the reduction in elements. The full rank of the matrix is a sufficient and unnecessary condition for the Hungarian algorithm to have a solution, because the cost matrix with full rank must have an allocation plan satisfying (2) and (3). □

Remark 2. *In Step 1, different small multi-digit random decimals which do not affect the ascending order of elements are added to each matrix element, in order to solve the influence of the same elements and the elements that can be eliminated by elementary transformation on the rank of the matrix. Adding different random decimals will affect the cost of the allocation plan. Some suboptimal solutions whose cost are close to the optimal solution probably become the optimal solution, so the added random numbers need to be small enough to reduce the impact. Therefore, Step 1 can make*

the full rank matrix approximately a necessary and sufficient condition for the Hungarian algorithm to have a solution, based on Corollary 1.

Example 1. *In Table 1, the matrix A is a full rank matrix. In Step 2, matrix A becomes matrix B in the process of gradually filling data. Matrix B is not full rank because of the influence of the same elements and the elements that can be eliminated by elementary transformation on the rank of the matrix, but the Hungarian algorithm and task allocation of matrix B have solutions at this time. The third-order identity matrix is a solution of the Hungarian algorithm for matrix B. After adding different small random numbers, matrix A becomes matrix C which is full rank, and the Hungarian algorithm and task allocation also have solutions.*

Table 1. The matrices used in Example 1.

$\begin{bmatrix} 1 & 2 & 100 \\ 1 & 2 & 101 \\ 98 & 99 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} 1.036 & 2.014 & 0 \\ 1.078 & 2.012 & 0 \\ 0 & 0 & 5.045 \end{bmatrix}$
(A)	(B)	(C)

Corollary 2. *Algorithm 1 can be used to find the solution of the minimum max cost.*

Proof. In Steps 2 and 3 of Algorithm 1, the Hungarian algorithm has no solution at first. With the increase in added elements, the Hungarian algorithm just has a solution when adding an element to the matrix. The last element added is the largest, because the elements are added to the matrix in ascending order. Additionally, it has to use the fewest elements to make the algorithm solvable, and the last added element must be selected. Therefore, the last element added to the matrix is the minimum max element. □

Remark 3. *In Step 3, the Hungarian algorithm for task allocation applies the following theorem to a given $n \times n$ cost matrix to find an optimal assignment to minimize the total cost, as shown in Algorithm 2.*

Theorem 1. *If a number is added to or subtracted from all of the elements of any one row or column of a cost matrix, then the optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.*

Algorithm 2 The Hungarian algorithm for task allocation

Step 1	Subtract the smallest element in each row from all the elements of its row.
Step 2	Subtract the smallest element in each column from all the elements of its column.
Step 3	Draw lines through appropriate rows and columns so that all the zero elements of the cost matrix are covered, and the minimum number of such lines is used.
Step 4	Test for optimality: (i) if the minimum number of covering lines is n, an optimal assignment of zeros is possible and the algorithm is finished. (ii) If the minimum number of covering lines is less than n, an optional assignment of zeros is not yet possible. In that case, proceed to Step 5.
Step 5	Determine the smallest element not covered by any line. Subtract this entry from each uncovered row, and then add it to each covered column. Return to Step 3.

Remark 4. *For the interior of the multi-USV system, check whether the USVs collide at the path intersection, after Algorithm 1 gives the allocation solution. In the case of possible collision, the USV with less task time will delay departure to avoid collision and reduce the impact on the max task time. Path planning methods and COLREGs are used to avoid collision between USVs and obstacles including other ships.*

4. Experiments and Discussions

The algorithm (MM) we proposed to minimize the max cost and reduce the total cost was compared with the Hungarian algorithm (HA) [17], the auction algorithm (AA) [19], the genetic algorithm (GA) [14] and the ant colony optimization algorithm (ACO) [13] in the task allocation model shown in Section 2. Due to our optimization direction, there were few compared algorithms for simultaneously considering the minimum max cost and the low total cost in multi-USV task allocation, as we know. HA and AA were exact algorithms for minimizing the total cost. GA was a heuristic algorithm for minimizing the max cost. ACO was a heuristic algorithm for reducing the max cost and the total cost at the same time.

All algorithms using in this paper were coded or recoded in MATLAB. All the simulations were run on a single laptop with 8GB RAM and 2.30 GHz i5-6300HQ CPU. In the heuristic algorithms of GA and ACO, population size $PS = 50$ and the generations of solution $G = 200$.

Evaluation criteria based on [14] were as follows. PM is the percentage of the max cost of the solution in (11), and PT is the percentage of the total cost of the solution in (12). SM is the max cost of the solution of these algorithms and ST is the total cost of the solution of these algorithms. $BKSM$ was the best-known solution of minimizing the max cost and $BKST$ was the best-known solution of minimizing the total cost.

$$PM = \frac{SM}{BKSM} \times 100 \tag{11}$$

$$PT = \frac{ST}{BKST} \times 100 \tag{12}$$

The influence of wind, wave and current on the motion model of USVs is complex, so simplified parameters are used in simulation and experiments to reflect the influence of environment and USV motion characteristics on the task time through the random change of parameters in the range. The experimental environment is as follows. The initial directions D_{ii} at the initial points and the required directions D_{jr} at the target points were random, which were clockwise based on due north. The initial points of USVs were random points in A_i and the target points were random points in A_t. The speed of the USVs V_i was 1 ± 0.1 m/s. The rotational speed of the USVs ω_i was 10 ± 1 °/s.

$$\begin{aligned} A_i &= \{(x,y)|0 \le x \le 100,\ 0 \le y \le 100\} \\ A_t &= \{(x,y)|100 \le x \le 200,\ 100 \le y \le 200\} \end{aligned} \tag{13}$$

Figure 2 gave the results PM and PT of HA, AA, GA, ACO and MM in the max cost and the total cost. PM and PT of the algorithms with the number of USVs from 1 to 30 were shown in Figure 2a,c. The results were specially shown in Figure 2b,d where the number of USVs were 10, 20 and 30.

In Figure 2a,b, PM of HA, AA and ACO were relatively large and mainly fluctuate in the range of 120% to 150%. This was because the optimization direction of HA and AA is to reduce the total cost, not the max cost. ACO as a heuristic algorithm reduced the maximum cost and total cost at the same time, and the effect of reducing the maximum cost is not good. The optimization direction of GA and MM was to reduce the maximum cost, and the results were better than other algorithms. PM of GA gradually rose to 120% with the increase in the number of USVs, while MM as an exact algorithm always maintains a low maximum cost, which is the best of these algorithms in PM.

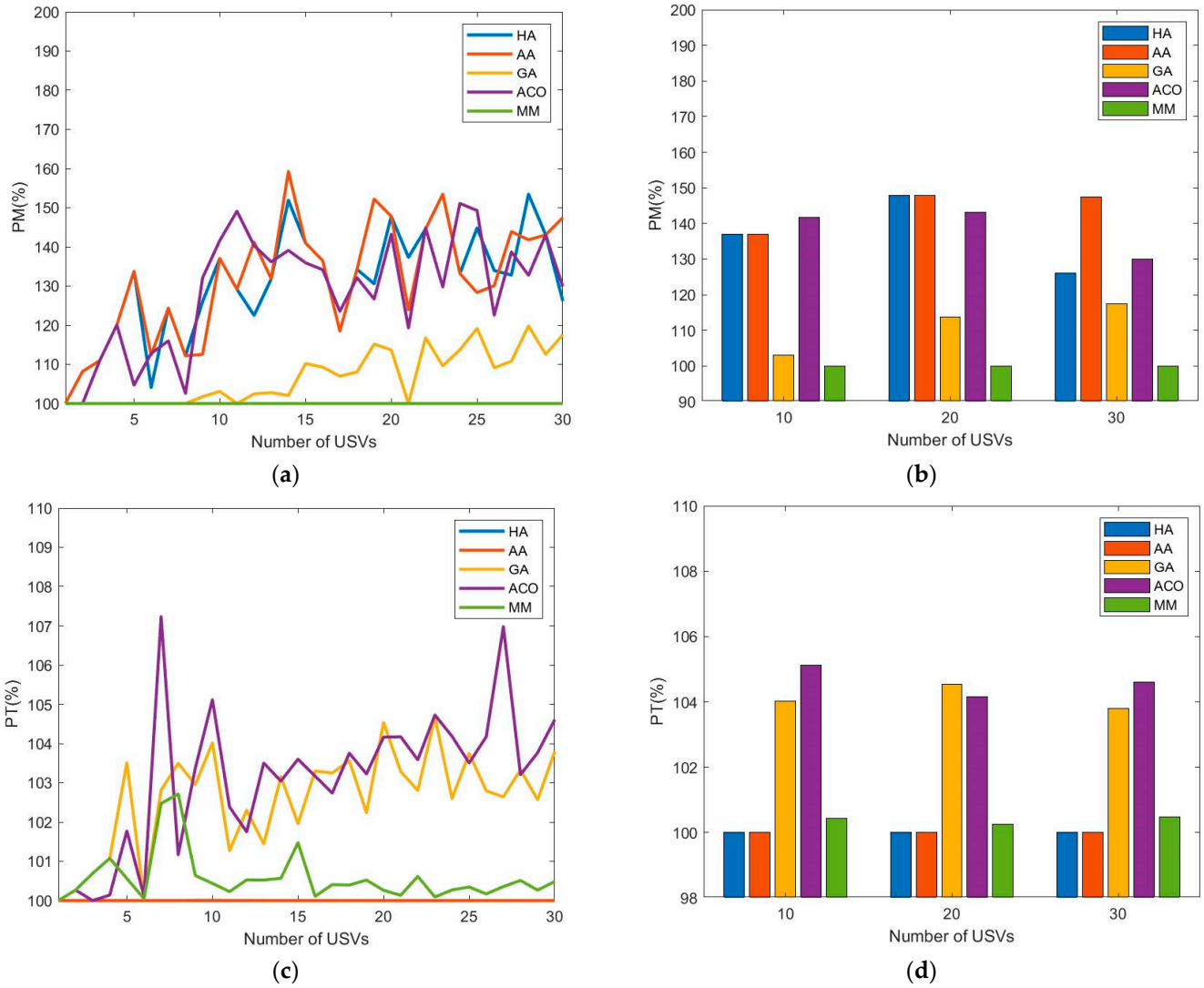

Figure 2. Comparison of algorithms in the max cost and the total cost. (**a**) *PM* of the algorithms with the number of USVs from 1 to 30. (**b**) *PT* of the algorithms with the number of USVs from 1 to 30. (**c**) *PM* of the algorithms where the number of USVs were 10, 20 and 30. (**d**) *PT* of the algorithms where the number of USVs were 10, 20 and 30.

In Figures 1c and 2d, the total cost was the evaluation criterion. HA and AA had the best results as exact algorithms to minimize the total cost. The MM proposed in this paper was based on HA and its *PT* was mainly between 100% and 101%. ACO took into account two optimization directions and the optimization direction of GA was to reduce the max cost, so their *PT* were mainly between 101% and 105%.

From the algorithm effectiveness perspective, MM was the best algorithm and had a large advantage in reducing the max cost. Meanwhile, the total cost of MM was slightly larger than that of the exact algorithm HA and AA whose optimization direction was to minimize the total cost.

To further demonstrate the effectiveness of the proposed algorithm, we used 10,000 cost matrices to detect whether the algorithm can minimize the max cost. The number of USVs were from 2 to 10 and the elements were random with a value from 5 to 10 in these cost matrices. This is to verify the effectiveness of the algorithm in a large number of random data, just as the Hungarian algorithm can calculate the solution of the minimum total cost of any given matrix. Using the limited range of raw data is because the solution of the algorithm proposed in this paper needs to be compared with the best solution and the best solution is obtained by traversing all solutions. The raw data are simplified in order to complete more experiments in a limited time to test the effectiveness of the algorithm. The

minimum max costs of all allocation plans obtained by traversal in the cost matrices were the same as these calculated by the proposed algorithm. Overall, these results showed that the proposed algorithm minimized the max cost and reduced the total cost.

To make readers easily understand the physical meaning of the best solution, we gave the best solution paths of the proposed algorithm in Figure 3. A simulation area of 500 pixels wide and 500 pixels high was shown in Figure 3. The black solid circles were sparse obstacles. The red points were initial points of USVs and the green points were target points. The initial directions D_{ii} at the initial points and the required directions D_{jr} at the target points were random, which were clockwise based on due north. The speed of the USVs V_i was 1 ± 0.1 pixels/s. The rotational speed of the USVs r_i was $10 \pm 1\ ^\circ$/s.

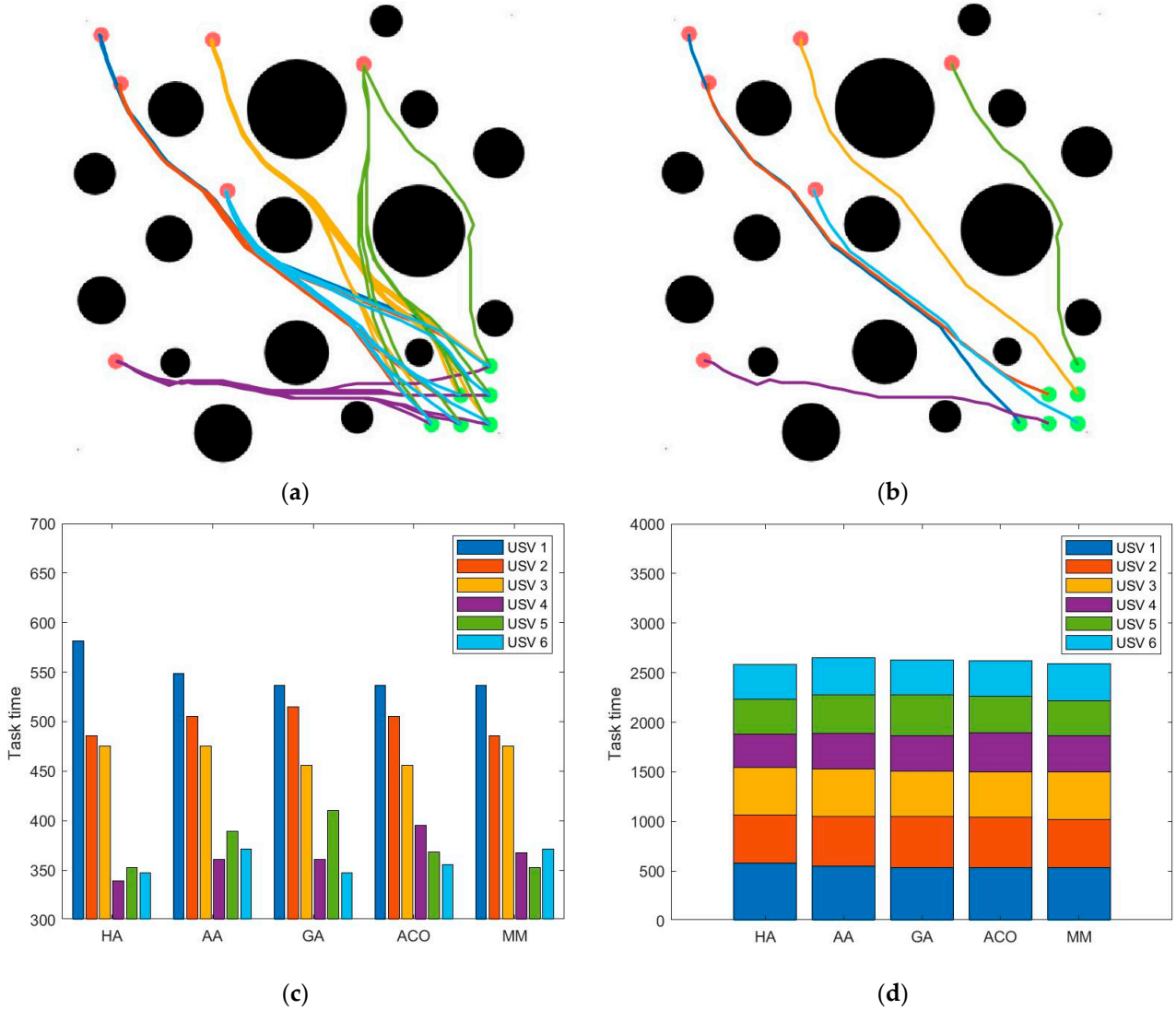

Figure 3. Path planning in task allocation, and the results of algorithms. (**a**) The planned paths from each USV to each target point. (**b**) The selected paths of the proposed algorithm MM. (**c**) The task time of each USVs in the results of algorithms. (**d**) The total task time of each USVs in the results of algorithms.

The paths shown in Figure 3a were from each initial point to each target point and the paths from the same initial point had the same color. The FMS [21] was used to plan these paths from the USVs to the target points, which used the Fast Marching Method [22] (FMM) twice for different purposes. In the FMS [23], the first FMM was used to build the safety potential field from the grid points of all obstacles to the other grid points in the planning space. The second FMM was used to build the distance potential field from

the initial point to the other grid points. Finally, the path with a certain distance from the obstacle was generated on the superimposed potential field. According to the length of the planned path, the time of passing through the path can be calculated.

The selected planned paths of the proposed algorithm were shown in Figure 3b, which also have the corresponding relationship between the USVs and the target points in the calculation result. In Figure 3c,d, these results showed that the proposed algorithm minimizing the max cost and reducing the total cost.

5. Conclusions

In the task allocation of USVs, there is relatively little research that focuses on the exact algorithm of minimizing the max cost and reducing the total cost to obtain the optimal solution. In this paper, we provide steps of the exact algorithm for task allocation. Its optimization direction is to minimize the max cost and reduce the total cost. We test the algorithm by comparing it with other algorithms in the max cost and the total cost to validate and evaluate the proposed algorithm. The algorithm is able to be used in the time-based cooperative working environment. In addition, we provide the method to build the cost matrix.

To further develop the algorithm, we consider adding the formation to task allocation in the future. USVs firstly assemble to form a formation to drive toward the target points, and then perform task allocation near the target points.

Author Contributions: Conceptualization, K.X. and Z.H.; methodology, Z.H.; software, Z.H., P.W. and Z.X.; formal analysis, Z.H., P.W. and Z.X.; investigation, Z.H.; supervision, K.X., P.W. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Sousa, D.; Hernandez, D.; Oliveira, F.; Luís, M.; Sargento, S. A Platform of Unmanned Surface Vehicle Swarms for Real Time Monitoring in Aquaculture Environments. *Sensors* **2019**, *19*, 22. [CrossRef] [PubMed]
2. Luo, S.C.; Singh, Y.; Yang, H.Y.; Bae, J.H.; Dietz, J.E.; Diao, X.; Min, B.-C. Image Processing and Model-Based Spill Coverage Path Planning for Unmanned Surface Vehicles. In Proceedings of the Oceans 2019 Mts/IEEE Seattle, Seattle, WA, USA, 27–31 October 2019; IEEE: New York, NY, USA, 2019.
3. Bae, J.H.; Luo, S.C.; Kannan, S.S.; Singh, Y.; Lee, B.; Voyles, R.M.; Postigo-Malaga, M.; Zenteno, E.G.; Aguilar, L.P.; Min, B.-C. Development of an Unmanned Surface Vehicle for Remote Sediment Sampling with a Van Veen Grab Sampler. In Proceedings of the Oceans 2019 Mts/IEEE Seattle, Seattle, WA, USA, 27–31 October 2019; IEEE: New York, NY, USA, 2019.
4. Felski, A.; Zwolak, K. The Ocean-Going Autonomous Ship-Challenges and Threats. *J. Mar. Sci. Eng.* **2020**, *8*, 16. [CrossRef]
5. Jorge, V.A.M.; Granada, R.; Maidana, R.G.; Jurak, D.A.; Heck, G.; Negreiros, A.P.F.; dos Santos, D.H.; Gonçalves, L.M.G.; Amory, A.M. A Survey on Unmanned Surface Vehicles for Disaster Robotics: Main Challenges and Directions. *Sensors* **2019**, *19*, 44. [CrossRef] [PubMed]
6. Yin, Q.Y.; Lv, Y.Z. Multichannel adaptive deployment and reliable communication design for unmanned surface vessel. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 7. [CrossRef]
7. Rauniyar, A.; Muhuri, P.K. Multi-Robot Coalition Formation Problem: Task Allocation with Adaptive Immigrants Based Genetic Algorithms. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, 9–12 October 2016; pp. 137–142.
8. Liu, Y.C.; Bucknall, R. Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing* **2018**, *275*, 1550–1566. [CrossRef]
9. Liu, Y.C.; Song, R.; Bucknall, R.; Zhang, X. Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (USVs) using self-organising maps and fast marching method. *Inf. Sci.* **2019**, *496*, 180–197. [CrossRef]
10. Raboin, E.; Svec, P.; Nau, D.S.; Gupta, S.K. Model-predictive asset guarding by team of autonomous surface vehicles in environment with civilian boats. *Auton. Robot.* **2015**, *38*, 261–282. [CrossRef]

11. Nam, C.; Shell, D.A. Robots in the Huddle: Upfront Computation to Reduce Global Communication at Run Time in Multirobot Task Allocation. *IEEE Trans. Robot.* **2020**, *36*, 125–141. [CrossRef]
12. Chopra, S.; Notarstefano, G.; Rice, M.; Egerstedt, M. A Distributed Version of the Hungarian Method for Multirobot Assignment. *IEEE Trans. Robot.* **2017**, *33*, 932–947. [CrossRef]
13. Chen, X.; Zhang, P.; Du, G.; Li, F. Ant Colony Optimization Based Memetic Algorithm to Solve Bi-Objective Multiple Traveling Salesmen Problem for Multi-Robot Systems. *IEEE Access* **2018**, *6*, 21745–21757. [CrossRef]
14. Zhang, H.G.; Luo, H.; Wang, Z.; Liu, Y.; Liu, Y. Multi-Robot Cooperative Task Allocation with Definite Path-Conflict-Free Handling. *IEEE Access* **2019**, *7*, 138495–138511. [CrossRef]
15. Shriyam, S.; Gupta, S.K. Incorporation of Contingency Tasks in Task Allocation for Multirobot Teams. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 809–822. [CrossRef]
16. Nunes, E.; McIntire, M.; Gini, M. Decentralized multi-robot allocation of tasks with temporal and precedence constraints. *Adv. Robot.* **2017**, *31*, 1193–1207. [CrossRef]
17. Kuhn, H.W. The Hungarian Method for the assignment problem. *Nav. Res. Logist.* **2005**, *52*, 7–21. [CrossRef]
18. Nam, C.; Shell, D.A. Assignment Algorithms for Modeling Resource Contention in Multirobot Task Allocation. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 889–900. [CrossRef]
19. Shi, J.K.; Yang, Z.; Zhu, J.W. An auction-based rescue task allocation approach for heterogeneous multi-robot system. *Multimed. Tools Appl.* **2020**, *79*, 14529–14538. [CrossRef]
20. Lusk, P.C.; Cai, X.Y.; Wadhwania, S.; Paris, A.; Fathian, K.; How, J.P. A Distributed Pipeline for Scalable, Deconflicted Formation Flying. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5213–5220. [CrossRef]
21. Gomez, J.V.; Lumbier, A.; Garrido, S.; Moreno, L. Planning robot formations with fast marching square including uncertainty conditions. *Robot. Auton. Syst.* **2013**, *61*, 137–152. [CrossRef]
22. Sethian, J.A. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA* **1996**, *93*, 1591–1595. [CrossRef] [PubMed]
23. Liu, Y.C.; Bucknall, R. The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. *Appl. Ocean. Res.* **2016**, *59*, 327–344. [CrossRef]