

## Article

# Research on Unmanned Surface Vessel Aggregation Formation Based on Improved A\* and Dynamic Window Approach Fusion Algorithm

Ge-An Wei \* and Jian-Qiang Zhang

Ordnance Engineering College, Naval University of Engineering, Wuhan 430033, China; w1994706233@126.com  
\* Correspondence: w1017544520@126.com

**Abstract:** The traditional A\* and DWA fusion algorithm has three problems in the task of aggregation formation: one is the lack of meeting coordination strategy, the other is the inability to unify the terminal course, and the third is too many turning points in the global path. To solve the above problems, an improved algorithm is proposed. Firstly, by smoothing the global planning path, the stability of a USV heading in the navigation is improved. Then, by adding the key points of the global path, a guide path is formed to unify the terminal heading range. Finally, by adding an encounter coordination strategy, aggregation efficiency is improved. Simulation experiments were carried out in the Python environment based on this algorithm. The results show that the improved algorithm can improve the navigation obstacle avoidance ability of USVs and guide multiple USVs to finish the task of aggregation formation.

**Keywords:** A\* algorithm; DWA algorithm; USV; formation aggregation



**Citation:** Wei, G.-A.; Zhang, J.-Q. Research on Unmanned Surface Vessel Aggregation Formation Based on Improved A\* and Dynamic Window Approach Fusion Algorithm. *Appl. Sci.* **2023**, *13*, 8625. <https://doi.org/10.3390/app13158625>

Academic Editors: Pingli Lu and Weinan Gao

Received: 27 May 2023  
Revised: 1 July 2023  
Accepted: 24 July 2023  
Published: 26 July 2023



**Copyright:** © 2023 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

In contrast to manned ships, unmanned surface vessels (USVs) have been widely used as a new type of unmanned maritime platform in many fields such as marine observation, hydrographic survey, and accident search and rescue, leading to innovative changes. The unmanned platform at sea has greatly promoted the transition of the ship industry in the direction of informatization and unmanned intelligence. Gradually, USVs are replacing manned ships as the backbone in some fields. As an important development in unmanned technology, USV formation provides advantages in terms of lower cost, higher safety, stronger task adaptability, mutual synergy, and stronger data processing capabilities compared with single-boat manned ships. First, USV formation does not need to carry personnel, which can save expenses related to personnel life support, thereby greatly reducing the cost of the formation. Second, USVs are more flexible in dealing with a variety of complex environments. Under harsh maritime working conditions, the safety of the USV equipment is the only consideration, and there are no additional personnel adaptability and personal safety concerns. Third, the form and number of USV formations can be selected according to the requirements of the task to quickly adapt and respond to different task requirements. Fourth, USV formations can realize cooperative control through intelligent technology, and with higher formation synergy, the efficiency of mission execution can further be improved. Fifth, USV formations can carry more types of sensors and data acquisition equipment, allowing the collection, processing, and transmission of a large amount of data in real time, thereby providing support for more efficient and accurate mission execution.

In USV navigation control formation, path planning and obstacle avoidance technologies are the keys to intelligent navigation. Obstacle avoidance technology is essential in ensuring the safe and efficient navigation of a USV, and path planning technology can generate efficient and feasible trajectories based on the information collected by various

sensors and communication systems, whereby according to a USV's performance and environmental conditions, the USV is guided to reach the target location safely and accurately. Therefore, the usual method of intelligent navigation employs global path planning algorithms to achieve path planning in a known environment, and subsequently uses local path planning algorithms to achieve dynamic obstacle avoidance in a positional environment. The common global path planning algorithms include the reverse encryption algorithm (REA)\* based on a grid map [1], A\*, Dijkstra, and randomly exploring random tree (RRT)\* algorithms based on the sampling method [2], as well as a probabilistic roadmap (PRM) and intelligent AG based on neural networks and ant colony algorithms [3,4]. Local path planning algorithms include the artificial potential field method (PFM), dynamic window approach (DWA), time elastic band (TEB) [5], and vector field histogram (VFH) algorithms [6]. In integrating the above global and local path planning algorithms, the A\* algorithm provides the advantages of easy implementation, wide applicability, and strong scalability, whereas DWA can fully combine the underactuated characteristics of USV and displays stronger adaptability. Therefore, the two were selected as the ideal global and local path planning algorithms for fusion to achieve ideal results in autonomous navigation applications.

In the literature, Chen et al. [7] used the key turning point extraction algorithm to eliminate redundant inflection points and redundant nodes in the global path, optimized the traditional A\* algorithm, and fused it with DWA to optimize path length, smoothness, and safety. Lao et al. [8] improved the key point selection strategy based on the traditional A\* algorithm and fused it with DWA to construct a global optimal path evaluation function to achieve real-time optimal path planning. Li et al. [9] validated and analyzed the A\* and DWA fusion algorithm for a single USV. Zhan et al. [10] introduced the safety distance factor into the heuristic function of the A\* algorithm to improve the safety of the algorithm's planned path and subsequently used the planar structure method to smooth and optimize the global planned path to realize the path planning of the A\* and DWA fusion algorithm. Liu et al. [11] quantified and analyzed the obstacle information in the environment, thereby adjusting the weights of the heuristic function of the A\* algorithm to improve algorithmic efficiency and flexibility. In addition, they improved the path smoothing based on Floyd's algorithm to realize dynamic path planning in complex environments. Lu et al. [12] optimized the predicted cost function and heuristic function of the A\* algorithm to tune the global planning path to the requirements of the task. Zhao et al. [13] reduced the turn angle for path planning in the improved A\* algorithm by treating travel time as the cost and adjusting the weight of the heuristic function according to obstacle information. Bian et al. [14] incorporated the key track points in the global planning path of A\* into the evaluation function of the DWA algorithm to solve the problem of the traditional DWA algorithm which fails to evaluate under a combination of "C"-shaped obstacles, and improved the ability to navigate through dense obstacle zones. Wu et al. [15] proposed a hybrid dynamic path planning algorithm for FAGV based on improved A\* and improved DWA. This algorithm can avoid obstacles dynamically without being far away from the global optimal path. Dan et al. [16] proposed the improved A\* algorithm combined with the greedy algorithm for a multi-objective path planning strategy. This algorithm can achieve a smoother path and reduce the path length by about 5%. Bai et al. [17] proposed a path planning algorithm based on A\* and DWA to achieve global path optimization while satisfying security and speed requirements for unmanned aerial vehicles (UAVs). This algorithm shortens the path length, reduces the planning time, improves the smoothness of the UAV path, and enhances the safety of UAV path obstacle avoidance. R. Han et al. [18] proposed a distributed approach for multi-robot navigation which combines the concept of reciprocal velocity obstacle (RVO) and the scheme of deep reinforcement learning (DRL) to solve the reciprocal collision avoidance problem with limited information.

The above research indicates various application directions for the A\* and DWA fusion algorithm, mostly focusing on the path planning of a single intelligent body, and less on aggregation formation-related problems such as collaborative path planning of multi-

intelligent bodies and cross-encounters. In USV aggregation formation, the traditional A\* and DWA fusion algorithm is not applicable to this scenario for the following reasons. First, the global path ramp of the traditional A\* algorithm has numerous turning points, which causes the USVs to jitter in the local path planning stage. Second, the USVs are under-driven models, and the traditional fusion algorithm cannot unify the heading range of the USVs when arriving at the end point, which forms a certain obstacle to the cooperative movement after aggregation formation. Third, the intersection of multiple USV paths is not considered because of a lack of an encounter coordination strategy, which can easily lead to collision accidents or reduce obstacle avoidance efficiency. Therefore, this study improves the traditional A\* and DWA algorithms to address the three aforementioned problems. The algorithm is improved and optimized by smoothing the global planning path, specifying the end position heading range, and adding an encounter coordination strategy to make it suitable for USV aggregation formation.

In this paper, Section 1 mainly analyzes the importance of USV navigation control research and introduces the research status of USV navigation and formation assembly. Section 2.1 mainly expounds on the traditional A\* and DWA fusion algorithm. Section 2.2 mainly introduces the main contents of the improved algorithm. Sections 2.2 and 2.3 mainly introduces the main contents of the improved algorithm. Section 3 mainly carries out simulation experiments and comparative analysis. Section 4 mainly summarizes the findings and future prospects.

## 2. Materials and Methods

### 2.1. Overview of Traditional A\* and DWA Algorithms

#### 2.1.1. Traditional A\* Algorithm

The traditional A\* algorithm, proposed by Hart et al. [19], is a heuristic search algorithm that finds the shortest path to the target location in a graph. This algorithm is optimized on the basis of Dijkstra's algorithm; however, compared to Dijkstra's algorithm, a cost function  $f(n)$  is introduced to optimize the search process by reducing the number of nodes, thereby improving search efficiency. The basic idea of the A\* algorithm is to start from the starting point, calculate for each node the cost to the starting point  $g(n)$  and to the end point  $h(n)$ , and then add the two values to obtain  $f(n)$ . Commencing with the starting point, the node with the smallest value of  $f(n)$  is selected for extended traversal until the end point is reached, or OPENLIST is empty. The cost function is expressed as

$$f(n) = g(n) + h(n) \quad (1)$$

In Equation (1), the heuristic function  $h(n)$  is one of the most important factors affecting the temporal complexity of the A\* algorithm [20]. The specific flow of the traditional A\* algorithm is shown in Figure 1.

#### 2.1.2. Traditional DWA Algorithm

The traditional DWA algorithm, proposed by Fox et al. [21], obtains a feasible velocity combination within a certain time window by combining the limiting velocity and acceleration constraints of the robot's own physical limitations with the obstacle constraints of the object environment  $(v, w)$ , where  $v$  and  $w$  denote linear and angular velocities, respectively. The three types of constraints can be expressed as follows:

$$V_s = \{(v, w) | v_{\min} \leq v \leq v_{\max}, w_{\min} \leq w \leq w_{\max}\} \quad (2)$$

$$V_d = \{(v, w) | v_c - \dot{v}\Delta t \leq v \leq v_c + \dot{v}\Delta t, w_c - \dot{w}\Delta t \leq w \leq w_c + \dot{w}\Delta t\} \quad (3)$$

$$V_a = \left\{ (v, w) | v_a \leq \sqrt{2 \cdot \text{dist}(v, w) \cdot \dot{v}}, w_a \leq \sqrt{2 \cdot \text{dist}(v, w) \cdot \dot{w}} \right\} \quad (4)$$

where  $V_s$  is the self-limiting velocity constraint;  $V_d$  is the self-limiting acceleration constraint;  $V_a$  is the obstacle constraint;  $v_c$  denotes the current linear velocity;  $\dot{v}$  denotes linear acceleration;  $w_c$  is the current angular velocity;  $\dot{w}$  is angular acceleration;  $dist(v, w)$  denotes the distance between the trajectory and the nearest obstacle.

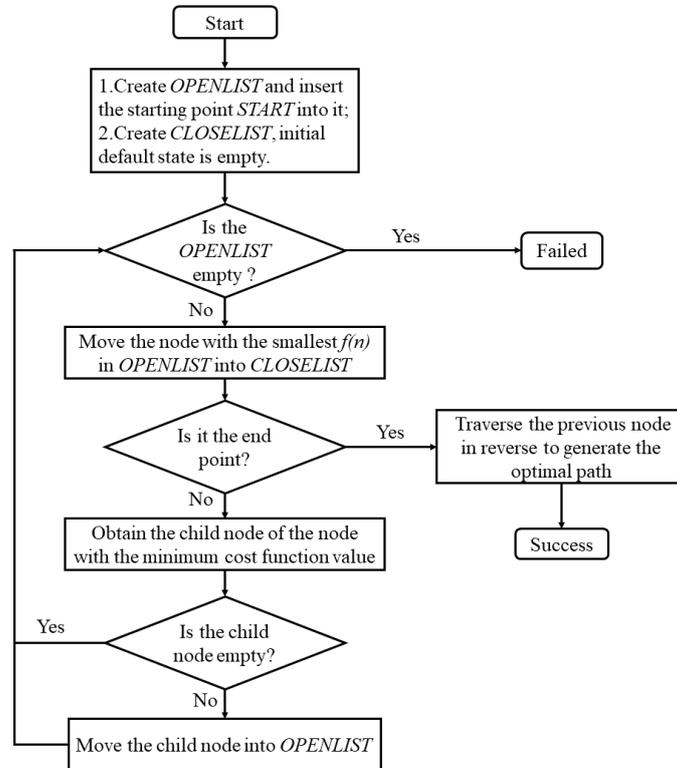


Figure 1. Flowchart of algorithm A\*.

The intersection of the three types of constraints determines the feasible velocity range:

$$V_r = V_s \cap V_d \cap V_a \tag{5}$$

By sampling all feasible velocity combinations according to the specified accuracy  $(v, w)$ , the predicted trajectory is formed and evaluated, and the one with the highest score is considered the ideal trajectory. The evaluation function is defined as follows:

$$E(v, w) = \sigma(\alpha \cdot head(v, w) + \beta \cdot dist(v, w) + \gamma \cdot vel(v, w)) \tag{6}$$

where  $\alpha, \beta, \gamma$  are the weight coefficients; the *head* function refers to the difference between the heading angle at the end of the trajectory and the azimuth angle of the target; the *dist* function calculates the distance between the target and the nearest obstacle; the *vel* function is the linear velocity magnitude;  $\sigma$  denotes the normalization process.

### 2.1.3. Traditional A\* and DWA Fusion Algorithm

The traditional A\* and DWA fusion algorithm can simultaneously consider the global static environment as well as the layout dynamic environment. The USV relies on the map resources pre-loaded into the database and its own sensors to develop situational awareness of the global and local environments. The global optimal path is first obtained according to the A\* algorithm, whereby the optimal path is discretized into a number of point traces. Subsequently, the locally optimal path is generated according to the target point of the DWA algorithm, as determined by the point traces, and combined with the location of surrounding obstacles to generate navigational information to guide the USV. The fusion algorithm can continuously update the heading state of the USV, considering the

navigational capability constraints and environmental information, thus achieving more efficient and accurate path planning. The algorithmic principle is illustrated in Figure 2.

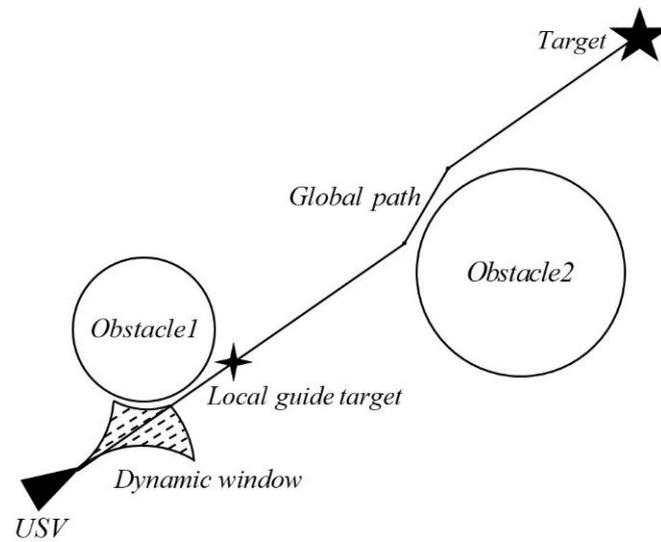


Figure 2. Traditional A\* and dynamic window fusion algorithm.

In Figure 2, the global path is first generated using the traditional A\* algorithm based on the position, size, and other information on the USV target and obstacles in the environment. Then, the local guide target is set according to the distance from the USV and used as the target point for the DWA algorithm. Finally, the traditional DWA algorithm is used to generate the dynamic window and guide the navigation.

### 2.2. Global Path Smoothing Optimization

#### 2.2.1. Problems with the Global Path of the Traditional A\* Algorithm

The traditional A\* algorithm grids the global environment and explores the surrounding grids, starting to traverse from the grid where the target is located. As shown in Figure 3, in the traditional algorithm, there are mainly four or eight directions for traversal search. However, the complexity of the actual global environment is much higher; therefore, even if the eight-direction search algorithm is used, the calculated global path is only composed of continuous line segments in eight directions, making it difficult to satisfy the requirements of the actual environment. When other angular directions appear, the traditional A\* algorithm combines them according to the existing directions, thereby causing additional turning points to appear in the generated paths. This increases the complexity of the global path and affects the stability and accuracy of the navigation phase of the USV.

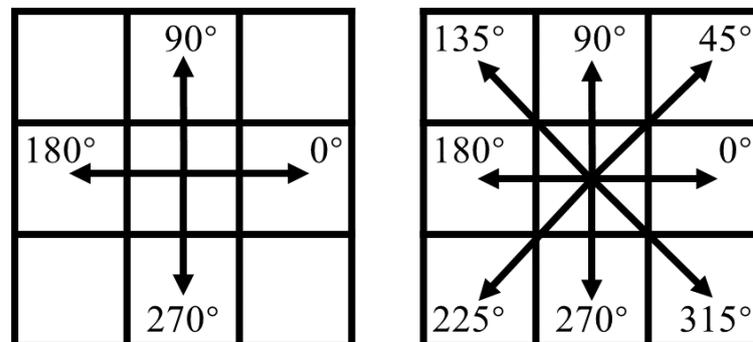


Figure 3. Traversal search of traditional A\* algorithm.

### 2.2.2. Improved Path Smoothing

To eliminate the extra turning points in the paths and achieve simplified global paths, a path smoothing algorithm is proposed in this study. Based on the paths obtained by the traditional A\* algorithm, the non-essential path points are eliminated, and the critical path points are retained to obtain smoothed and optimized critical paths. The specific steps of this algorithm are shown in Figure 4 below. After eliminating the non-essential path points, the number of points in the optimized path set is reduced. In subsequent steps of the fusion algorithm, optimized global paths are collected and added to the path point set according to the unit grid length of the A\* algorithm.

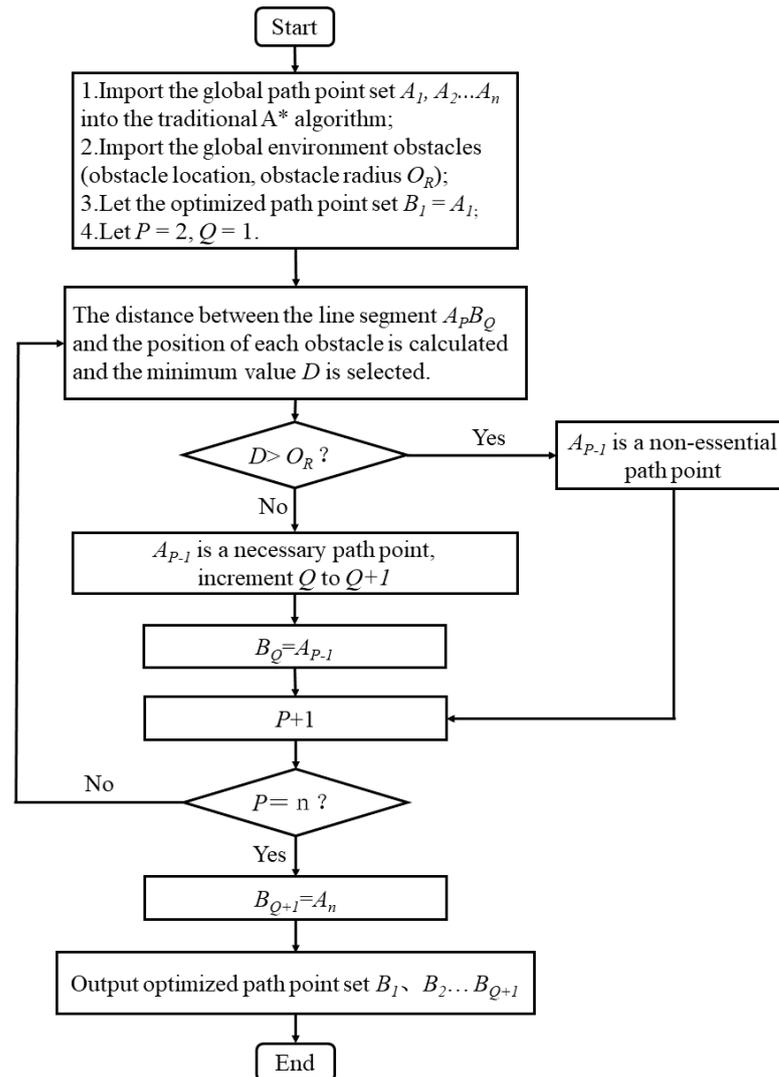


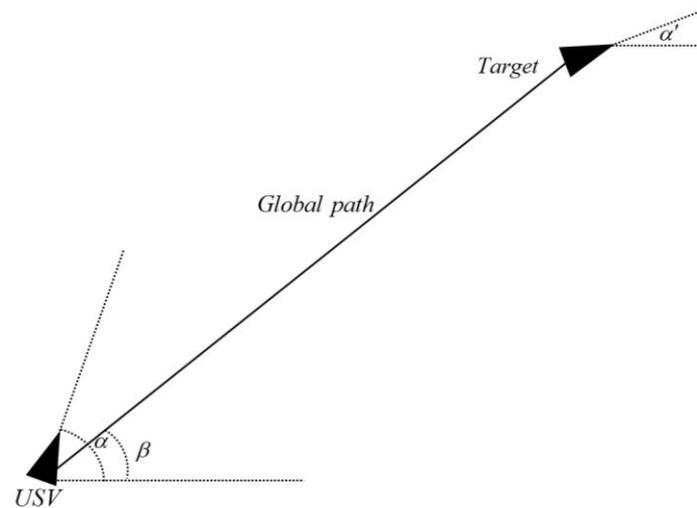
Figure 4. Flowchart of improved path smoothing algorithm.

### 2.3. Designated End Position Heading

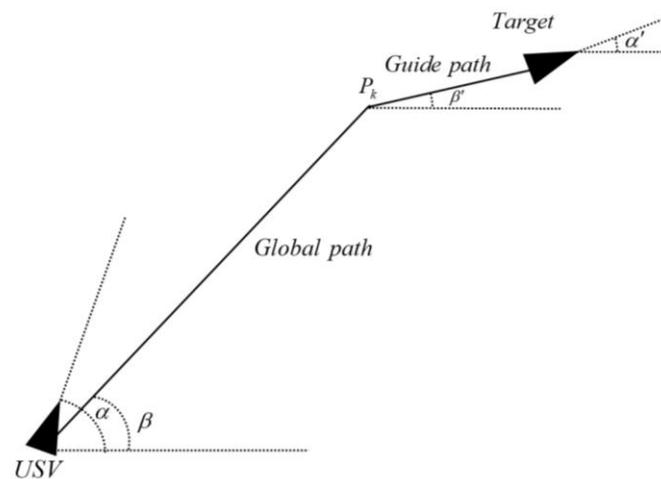
#### 2.3.1. Traditional Fusion Algorithm Endpoint Position Heading

In the traditional fusion algorithm, the heading at the end point of the USV is determined by environmental obstacles and the global path. The A\* algorithm only provides the optimal solution for reaching the end point from the starting point. Moreover, the traditional DWA algorithm cannot realize the function of customizing the end point heading, and the method of modifying the path evaluation index function is complex and difficult to realize. As shown in Figure 5, in an ideal environment without obstacles, regardless of the value of the USV heading angle  $\alpha$  at the starting position, the end point heading  $\alpha'$  of

the USV, determined by the traditional fusion algorithm, always converges to the end point azimuth  $\beta$  of the starting position, and the longer the global path length, the smaller the difference between the two angles  $|\alpha' - \beta|$ . Therefore, the terminal heading of the USV is always determined by the azimuth angle  $\beta$ , which is affected by the global path length.



(a)



(b)

**Figure 5.** (a). USV heading and end point bearing. (b). Global path end point guidance section added.

### 2.3.2. Adding a Global Path End Point Guide Segment

A method is proposed in this study to realize the control of the end point heading of the USV, whereby a key waypoint is set on the basis of the global path obtained by the traditional A\* algorithm to form a global path end point guidance section. The guidance section is the line between the key waypoint  $P_k$  and the target point, and the azimuth of the key waypoint  $P_k$  at the end point  $\beta'$  is the reference value of the end point heading control. Thus, the longer the length of the guidance section, the smaller the error value  $|\alpha' - \beta'|$ . Therefore, the length of the guidance section and the azimuth angle  $\beta'$  can be adjusted to achieve control of the end point heading of the USV.

## 2.4. Synergistic Optimization of Multi-USV Encounters

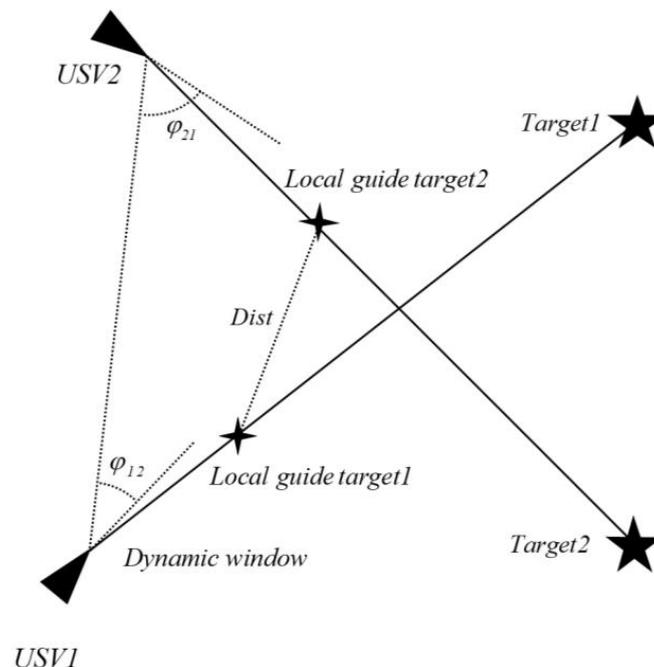
### 2.4.1. Encounter Problems in the Traditional Fusion Algorithm

The traditional fusion algorithm is designed to solve the single USV navigation problem. In a multi-USV environment, the algorithm achieves the desired effect only when the global paths of each USV do not cross and maintain a considerable distance from each other. In scenarios involving encounters, it is difficult to complete safe and efficient navigation tasks using the traditional fusion algorithm.

### 2.4.2. Encounter Coordination Strategy for Multiple USVs

To solve the encounter problem in multi-USV navigation, an encounter coordination strategy is proposed. First, an encounter judgment rule is set such that when the distance between the local guidance points of each USV  $Dist$  is less than or equal to a specified value, the USVs are judged to be in an encounter situation. Second, the priority of USV maneuvering is set to avoid a collision. The USVs considered to be in a mutual encounter situation are grouped, and the outboard angle value  $\varphi$  of the current USV relative to another USV is calculated, whereby the USV with the larger outboard angle value is given higher priority to avoid a collision. Finally, according to the priority level, the local guidance point of each USV is adjusted, only keeping the local guidance point position of the low-priority USV unchanged, and the current position of the other USV is incorporated into the global environment of the high-priority USV as the midpoint of the obstacle. At this time, the low-priority USV gradually decelerates to the local guidance point, and the high-priority USV maneuvers rapidly toward the global target point to ensure navigation safety. The process continues until the distance  $Dist$  between the two local guidance points is greater than the critical value, at which point the encounter situation is circumvented, and the low-priority USV resumes movement from the local target point, continuing to guide the USV while gradually accelerating.

As shown in Figure 6, when the distance  $Dist$  between USV1 and USV2's local guidance point reaches the critical value, the outboard angle values are compared, and if  $\varphi_{12} < \varphi_{21}$ , USV2's collision avoidance priority is higher than that of USV1, whereby USV1 will slow down to the current local guidance point to avoid USV2, and USV2 will sail normally.



**Figure 6.** Encounter coordination strategy for multiple USVs.

### 3. Experiments and Results

To verify that the improved fusion algorithm can further enhance the USV obstacle avoidance synergy and achieve USV aggregation formation, the traditional fusion and improved algorithms were simulated in the Python environment, and the results were analyzed and compared. The parameters of the USV kinematic and DWA algorithms are shown in Tables 1 and 2, respectively.

**Table 1.** USV kinematic parameters.

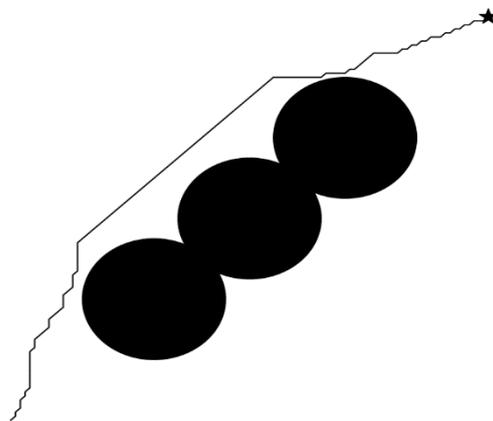
Type	Parameter Value (Unit)
Minimum speed	0 (kn)
Maximum speed	20 (kn)
Minimum angular velocity	15 (deg-s) <sup>-1</sup>
Maximum angular velocity	15 (deg-s) <sup>-1</sup>
Acceleration	2 (kn-s) <sup>-2</sup>
Angular acceleration	5 (deg-s) <sup>-2</sup>

**Table 2.** Fusion algorithm parameters.

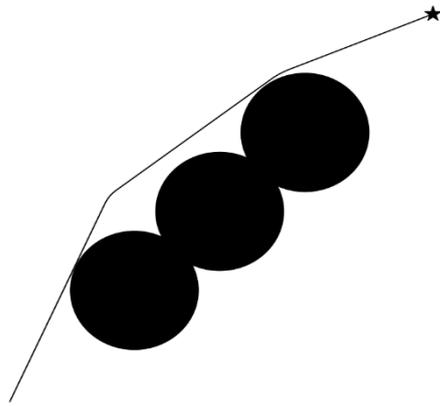
Type	Parameter Value (Unit)
Collision radius	0.25 (nm)
Time period	0.01 (h)
Speed resolution	0.1 (kn)
Angular velocity resolution	1 (deg-s) <sup>-1</sup>
Predicted track time	120 (s)
Global path unit length	0.05 (nm)
$\alpha$	0.1
$\beta$	0.1
$\gamma$	0.7

#### 3.1. Global Path Smoothing Verification Experiment

Based on Tables 1 and 2, a single USV path planning environment containing three obstacles was created, and the traditional A\* algorithm was compared with the smoothing optimization algorithm. In Figures 7 and 8, the global path steering points are significantly reduced after smoothing optimization, from 41 to 2 in this scenario, which is a positive outcome of smoothing optimization.



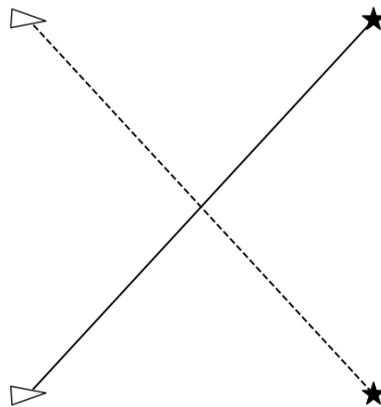
**Figure 7.** Global path of traditional A\* algorithm.



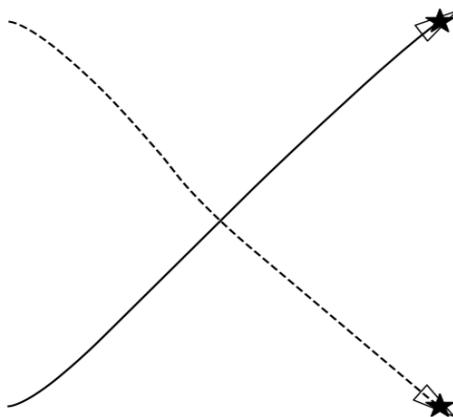
**Figure 8.** Global path after smoothing optimization.

### 3.2. Designated End Position Heading Verification Experiment

The dual-USV path planning environment was created according to Tables 1 and 2, and the heading control reference  $\beta'$  of the optimized algorithm was set to  $0^\circ$  in comparing the traditional fusion algorithm with the optimized algorithm, as shown in Figures 9–12.



**Figure 9.** Global path of traditional A\* algorithm.



**Figure 10.** USV trajectory and end position heading of traditional fusion algorithm.

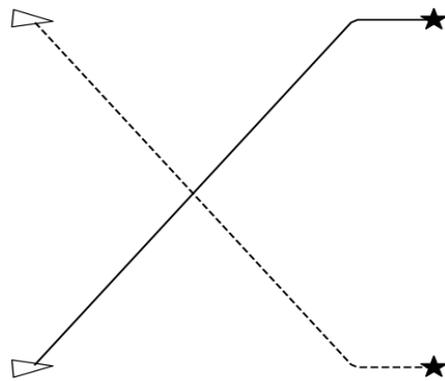


Figure 11. Global path of optimized algorithm.

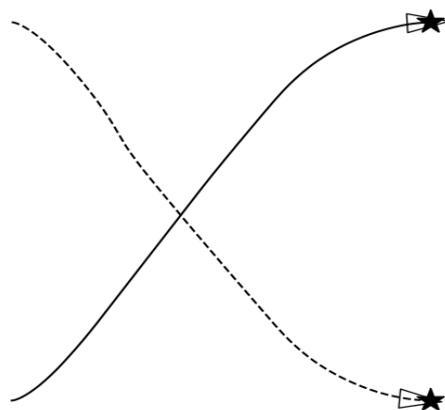


Figure 12. USV trajectory and end position heading of optimized fusion algorithm.

In Figures 9–12, after the optimized algorithm is added to the end guidance segment, the USV adjusts the heading at the end of the range under the influence of the guidance segment, whereby the heading converges towards  $\beta'$  at the end position.

### 3.3. Validation Experiment for Encounter Coordination

Based on Tables 1 and 2, a dual-USV environment was created, and USV head-on encounter and cross-encounter situations were set up to compare the traditional fusion algorithm and the optimized algorithm with an encounter coordination strategy, as shown in Figures 13–16.



Figure 13. Dynamic window during traditional algorithm encounter (head-on encounter situation).

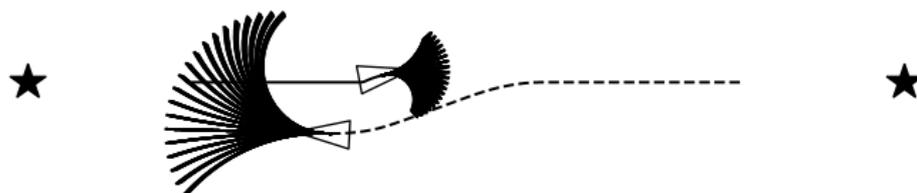


Figure 14. Dynamic window during encounter with optimization algorithm (head-on encounter situation).

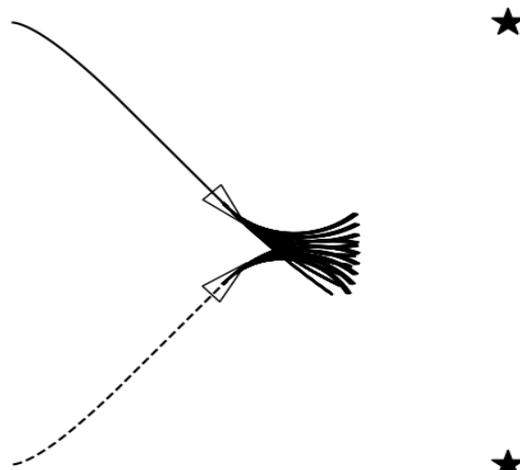


Figure 15. Dynamic window during traditional algorithm encounter (cross-encounter situation).

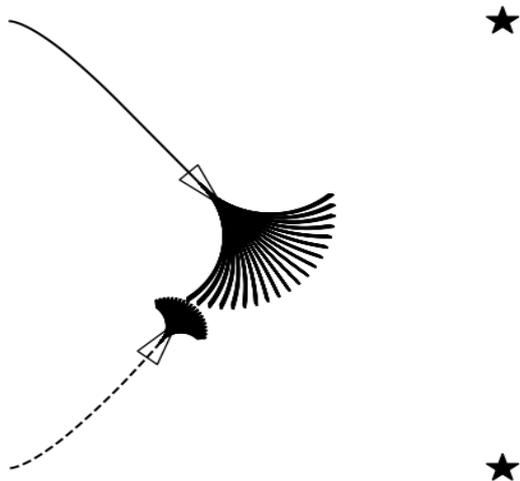
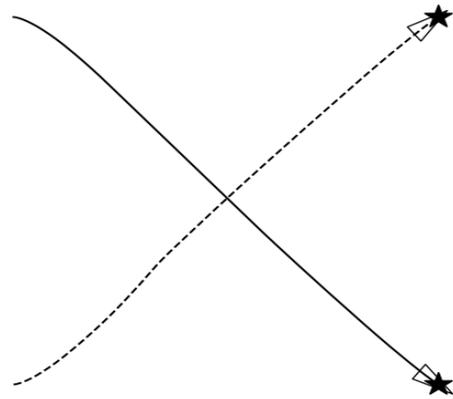


Figure 16. Dynamic window during optimization algorithm encounter (cross-encounter situation).

In Figures 13–16, in the traditional algorithm without an encounter coordination strategy, the use of the DWA algorithm fails to achieve the desired avoidance maneuver in an encounter situation. In the simulation experiment results of Figures 13 and 15, the USVs collide, and the navigation task cannot be completed. In contrast, the optimization algorithm with an encounter coordination strategy provides good safety and applicability by setting the priority of avoidance, thereby allowing one of the USVs to slow down and avoid a collision to achieve safe navigation while also complying with the actual rules of safe maritime navigation. In Figures 17 and 18, under the optimization algorithm, the USV does not take a large angle maneuver during the encounter and can choose an efficient cooperative collision avoidance path with a shorter distance, which allows the USV to economize on fuel in practical situations.



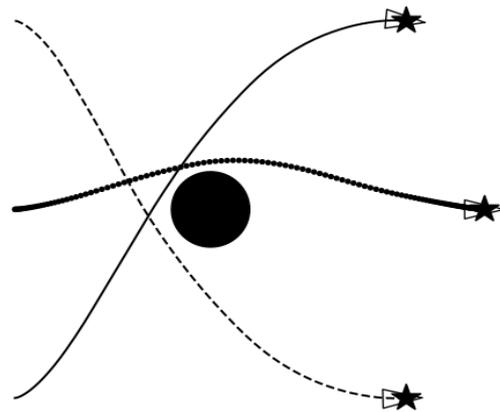
Figure 17. Optimization algorithm encounter trajectories (cross-encounter situation).



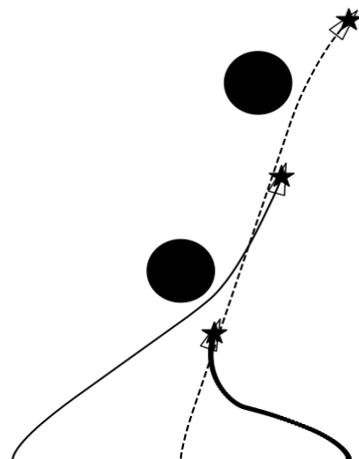
**Figure 18.** Optimization algorithm encounter trajectories (cross-encounter situation).

### 3.4. Multi-USV Aggregation Formation Experiment

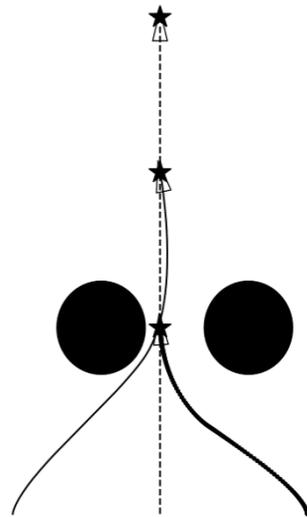
According to Tables 1 and 2, three USV environments were created, and obstacles were added. The reference value of heading control  $\beta'$  of the optimization algorithm  $\beta'$  was set, and a simulation experiment was conducted for USV aggregation formation. The results are shown in Figures 19–21.



**Figure 19.** Simulation of aggregation formation ( $\beta' = 0^\circ$ ).



**Figure 20.** Simulation of aggregation formation ( $\beta' = 60^\circ$ ).



**Figure 21.** Simulation of aggregation formation ( $\beta' = 90^\circ$ ).

In Figures 19–21, it is observed that the optimized algorithm provides significant improvements in obstacle avoidance, encounter, and end course uniformity in the environment of the three USVs, achieving ideal results in the simulation environment, by completing more complicated USV aggregation formation tasks.

Through the above experiments and analysis, the method proposed in Reference [18] is compared with the fusion algorithm in this paper, as shown in Table 3:

**Table 3.** Algorithm comparison.

Algorithm Name	Fusion Algorithm of A* and DWA	Fusion Algorithm of RVO and DRL
Algorithm core method	Fusion algorithm of Improved A* and traditional DWA	Fusion algorithm of RVO and multiple PPO deep reinforcement learning
Number of robots in formation	3	10/16/20
Designated terminal course	Yes	No
Acclimatization	Strong	Weak
Computing power demand	Low	High
Maneuverability requirements of robots	Low	High

Compared with the method proposed in reference [18], the fusion algorithm in this paper has the following advantages:

1. By improving the traditional A\* algorithm, this paper realizes the function of specifying the terminal course, which is more comprehensive in formation control ability;
2. The algorithm in this paper can be successfully implemented in multi-type environments by setting appropriate parameters. However, the deep reinforcement learning algorithm needs many simulated environments for training, and also needs targeted training according to different types of environments to achieve better results. Therefore, the algorithm proposed in this paper is more adaptable to the environment;
3. A deep reinforcement learning algorithm needs a lot of computing power and takes a long time to obtain an ideal training model, which puts higher demands on the equipment configuration involved in the calculation. The algorithm proposed in this paper is based on traditional A\* and DWA algorithms, which have lower complexity and less demand for computing power;
4. The RVO algorithm itself is prone to heading jitter in the navigation process, which puts forward higher requirements for the robot's maneuverability, while the DWA algorithm fully considers the robot's maneuverability and is more suitable for under-actuated systems such as USV.

At the same time, the shortcomings of the algorithm in this paper are that the number of robots in the formation is small, and more complex collision avoidance strategies need to be introduced, which puts forward higher requirements for the efficiency of the algorithm and is more difficult to realize.

#### 4. Conclusions

In this paper, we analyzed the traditional A\* and DWA fusion algorithm to investigate the problems in the predicted trajectory window during collision avoidance. By smoothing the global planning path, the stability of the USV heading in the navigation stage was improved. Subsequently, global path key points were added to form the end point guidance segment, which can specify the end point location heading range. For the multi-USV encounter coordination problem, the aggregation efficiency and safety were improved by introducing an encounter coordination strategy. The simulation results show that the improved fusion algorithm can better improve the control ability and navigation efficiency of USVs and complete the task of USV aggregation formation.

At the same time, the description of the USV motion model in this algorithm is relatively simple, with a certain divergence from the actual USV motion situation. Moreover, the model of the simulation environment is not rich enough, and there is a lack of a more comprehensive analysis of the efficiency and applicable environment of the algorithm. The above problems will be further improved in the follow-up algorithm improvement and real-scene testing research to obtain more ideal experimental results.

**Author Contributions:** Conceptualization, G.-A.W. and J.-Q.Z.; validation, G.-A.W. and J.-Q.Z.; writing—original draft preparation, G.-A.W.; writing—review and editing, J.-Q.Z. 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:** Not applicable.

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

#### References

1. Zhang, A.; Li, C.; Bi, W. Rectangle expansion A\* pathfinding for grid maps. *Chin. J. Aeronaut.* **2016**, *29*, 1385–1396. [[CrossRef](#)]
2. Qureshi, A.H.; Ayaz, Y. Potential functions based sampling heuristic for optimal path planning. *Auton. Robot.* **2016**, *40*, 1079–1093. [[CrossRef](#)]
3. Liu, C.-A.; Yan, X.-H.; Liu, C.-Y.; Wu, H. A dynamic path planning method for mobile robots based on improved ant colony algorithm. *J. Electron.* **2011**, *39*, 1220–1224.
4. Xiaoyan, W.; Yang, L.; Zhang, Y.; Meng, S. Robot path planning based on improved potential field ant colony algorithm. *Control. Decis. Mak.* **2018**, *33*, 1775–1781. [[CrossRef](#)]
5. Roesmann, C.; Feiten, W.; Woesch, T.; Hoffmann, F.; Bertram, T. Trajectory Modification Considering Dynamic Constraints of Autonomous Robots. In Proceedings of the 7th German Conference on Robotics (ROBOTIK 2012), Munich, Germany, 21–22 May 2012; pp. 1–6.
6. Statheros, T.; Howells, G.; McDonald-Maier, K. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *J. Navig.* **2008**, *61*, 129–142. [[CrossRef](#)]
7. Chen, J.; Xu, L.; Chen, J.; Liu, Q. Improved A\* and dynamic window method for mobile robot path planning. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 1650–1658. [[CrossRef](#)]
8. Lao, C.; Li, P.; Feng, Y. Greenhouse robot path planning based on improved A\* and DWA algorithm fusion. *J. Agric. Mach.* **2021**, *52*, 14–22.
9. Li, W.G.; Wang, L.; Fang, D.X.; Li, Y.W.; Huang, X. Improved A\* and dynamic window method for path planning algorithm. *Syst. Eng. Electron. Technol.* **2021**, *43*, 3694–3702.
10. Zhan, W.; Huang, Y.Q. Robot path planning by fusing secure A\* algorithm with dynamic windowing method. *Comput. Eng.* **2022**, *48*, 105–112+120. [[CrossRef](#)]
11. Liu, J.; Xue, L.; Zhang, H.; Liu, Z. Fusing improved A\* and DWA algorithms for dynamic path planning of robots. *Comput. Eng. Appl.* **2021**, *57*, 73–81.

12. Lu, X.; Wu, C.; Yang, G.; Zhang, B.; Chen, Z. Improved A\* and DWA Algorithms for Path Planning of Orchard Spraying Robots. *Computer Engineering and Applications* 1–8. Available online: <http://kns.cnki.net/kcms/detail/11.2127.TP.20221125.1958.036.html> (accessed on 18 May 2023).
13. Zhao, J.; Zhao, T.; Feng, Y.; Cao, Z.; Zhong, Y. Research on robot path planning based on improved A\* algorithm and DWA fusion. *Exp. Technol. Manag.* **2023**, *40*, 87–92. [[CrossRef](#)]
14. Bian, Y.M.; Ji, P.C.; Zhou, Y.H.; Yang, M. Improved DWA-based obstacle avoidance path planning for mobile robots. *Chin. J. Eng. Mach.* **2021**, *19*, 44–49. [[CrossRef](#)]
15. Wu, B.; Chi, X.; Zhao, C.; Zhang, W.; Lu, Y.; Jiang, D. Dynamic Path Planning for Forklift AGV Based on Smoothing A\* and Improved DWA Hybrid Algorithm. *Sensors* **2022**, *22*, 7079. [[CrossRef](#)] [[PubMed](#)]
16. Xiang, D.; Lin, H.; Ouyang, J.; Huang, D. Combined improved A\* and greedy algorithm for path planning of multi-objective mobile robot. *Sci. Rep.* **2022**, *12*, 13273. [[CrossRef](#)] [[PubMed](#)]
17. Bai, X.; Jiang, H.; Cui, J.; Lu, K.; Chen, P.; Zhang, M. UAV Path Planning Based on Improved A\* and DWA Algorithms. *Int. Aerosp. Eng.* **2021**, *2021*, 4511252. [[CrossRef](#)]
18. Han, R.; Chen, S.; Wang, S.; Zhang, Z.; Gao, R.; Hao, Q.; Pan, J. Reinforcement Learned Distributed Multi-Robot Navigation With Reciprocal Velocity Obstacle Shaped Rewards. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5896–5903. [[CrossRef](#)]
19. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [[CrossRef](#)]
20. Feng, K.; Ji, X.; Yang, X. Application of A\* algorithm in self-driving vehicle path planning. *Automot. Pract. Technol.* **2020**, *45*, 25–28. [[CrossRef](#)]
21. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [[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.