**To cite this article:** HOU Y Q, TAO H, GONG J B, et al. Cooperative path planning of USV and UAV swarms under multiple constraints[J/OL]. Chinese Journal of Ship Research, 2020, 16(1). http://www.ship-research.com/EN/Y2020/V16/I1/74.

**DOI:** 10.19693/j.issn.1673-3185.02091

# Cooperative path planning of USV and UAV swarms under multiple constraints

*HOU Yueqi*[1], *TAO Hao*[2], *GONG Junbin*[2], *LIANG Xiaolong*[*1], *ZHANG Nuo*[3]

1 Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China

2 China Ship Development and Design Center, Wuhan 430064, China

3 The 31005 Unit of PLA, Beijing 100094, China

**Abstract**: [**Objectives**] For achieving navigational safety and continuous communication link between swarms of unmanned marine vehicle (UMV) during mission execution, the cooperative path planning of unmanned surface vehicle (USV) and unmanned aerial vehicle (UAV) swarms is studied. [**Methods**] Keep-in and keep-out geo-fences are used to carry out scene modeling, and the problems of threat and obstacle avoidance are transformed into geo-fence constraints. Aiming at collision avoidance and continuous communication link between vehicles, we propose a judgment criterion for the constraints of collision avoidance and communication link via time sequence detection. The average travel time (ATT) of the swarm is taken as the path optimization function, and the multiple constraints are transformed into penalty functions. A self-adaptive differential evolution algorithm is adopted to solve the optimization problem. [**Results**] The proposed method can ensure safe navigation and communication link between USV and UAV swarms in the hostile and obstacle-filled environment, and achieve the shortest ATT under multiple constraints. [**Conclusions**] This method has practical value for the off-line path planning of UMV swarms in the hostile and obstacle-filled environment.

**Key words**: unmanned surface vehicles (USV); unmanned aerial vehicles (UAV); swarms of unmanned marine vehicles; cooperative path planning

**CLC number:** U664.82

## 0 Introduction

Since the concept of Airsea Battle was put forward by the US Army in 2011, with the progressive evolution of such concepts as multi-domain operation, warfare has no longer been limited to single-domain operation, but gradually transferred to cross-domain cooperation and multi-domain integration [1]. Under the idea of multi-domain operation, in combination with broad prospects of unmanned combat, a new concept of warfare with unmanned marine vehicle (UMV) swarms has been proposed. At present, the research on UMV swarms in mission planning, path planning, environmental perception, and formation cooperation is still in its infancy, facing many challenges [2].

UMV swarms are composed of unmanned surface vehicle (USV) and unmanned aerial vehicle (UAV) swarms. Cooperative path planning considering cross-domain characteristics of USVs and UAVs is one of the key technologies. At present, most of the references focus on path planning of USVs or UAVs. Combining the advantages of a Voronoi diagram and a visibility graph, Niu et al. [3] solved shortest-path planning of USVs by the Dijkstra's algorithm. Under the same computational effi-

**Received:** 2020-08-28 **Accepted:** 2021-01-15

**Supported by:** National Natural Science Foundation of China (61701471)

**Authors:** HOU Yueqi, male, born in 1995, Ph.D. candidate. Research interest: cooperative control of UAV swarms. E-mail: afeu__hyq@163.com

TAO Hao, male, born in 1987, Ph.D., engineer

GONG Junbin, male, born in 1978, Ph.D., senior engineer

LIANG Xiaolong, male, born in 1981, Ph.D., professor, doctoral supervisor. Research interests: operational application of aviation swarms and intelligent air traffic control. E-mail: afeu__lxl@sina.com

***Corresponding author:** LIANG Xiaolong

ciency, this method yields a shorter path than the traditional Voronoi diagram method does. However, it fails to model a threat area with a non-zero zone. Yang et al. [4] converted satellite thermal images into binary images to provide environmental information for path planning. Considering the turning performance and environmental constraints of a USV, they solved the shortest collision-free path of the USV by using the finite-angle $A^*$ algorithm. With average solving time of about 0.05 s, this method can be used for real-time path planning. However, it is difficult to guarantee the real-time accuracy of satellite images. In terms of path planning for USV formations, Ouyang et al. [5] planned paths of USV formations based on an improved rapidly exploring random tree algorithm. In addition, they proposed a non-strict conformal correction vector, which can make USVs keep formations stable to a maximum extent while avoiding obstacles. Zhou et al. [6] planned a reliable obstacle-avoidance path for a USV formation, based on a deep reinforcement learning method. With two designed reward functions, USVs are trained to learn the two behaviors of keeping a fixed formation shape and flexibly changing formation shapes. This method can yield a good result in an environment with cluttered obstacles. UAVs were studied earlier than USVs in terms of path planning. At present, the latest research mainly focuses on cooperation between UAVs. For example, paths for simultaneous arrival of UAVs are designed to improve the effectiveness of cooperative attacks. In view of multi-UAV path planning for multi-target attacks, Babel [7] proposed a priority algorithm of simultaneous arrival based on the shortest path. With this method, paths satisfying time constraints can be planned for multiple UAVs in the case of threats and obstacles. However, cooperative path planning considering cross-domain characteristics has been rarely studied. With respect to path planning of unmanned underwater vehicles and unmanned aerial-aquatic vehicles for cooperative attacks, Wu [8] modeled constraints in different media and relevant optimization indexes. On this basis, a method of cooperative path planning with the two stages of surface search and underwater attack was proposed, and a particle swarm optimization algorithm was adopted for calculation. Thus, the cross-domain cooperative attack was realized.

From the research in the above references, the existing methods mainly focus on path planning of homogeneous platforms, with little consideration about coupled constraints between cross-domain heterogeneous platforms. The coupled constraints that need to be considered in cooperative path planning of USV and UAV swarms include continuous communication link and arrival time. As direct communication distances of USVs from shore-end command centers are shorter than those of UAVs, communication between USVs is usually relayed through UAVs in order to extend combat radii. Therefore, in cooperative navigation of USVs and UAVs, communication-distance constraints are necessarily retained all the time. In addition, arrival time constraints of USVs and UAVs also need to be considered to ensure overall cooperativity of cross-domain swarms at their arrival in combat zones.

In view of cooperative path planning of USV and UAV swarms, on the basis of the constraints of threat avoidance, maneuverability, and collision avoidance, this paper further considered communication-link constraints based on requirements of cross-domain communication between USVs and UAVs. This aims to make sure USV and UAV swarms can communicate with each other all the time during their navigation. Then, the paper designed a path optimization function, converting multiple constraints into penalty functions, and used a self-adaptive differential evolution algorithm to solve the optimization problem.

# 1 Modeling for cooperative path planning

## 1.1 Problem description

In the initial stage of mission execution, a USV swarm sets sail from a designated water area at the shore end, while a UAV swarm takes off from a designated landing field. After navigation of a certain distance, they safely arrive in their respective mission areas to execute subsequent missions like patrol, search, reconnaissance, positioning, tracking, and attack. The so-called cooperative path planning of USV/UAV swarms refers to the generation of swarm navigation paths of optimal path performance with initial positions of the USV/UAV platforms as start points and entry points of the designated mission areas as finish points. Such paths are generated under comprehensive consideration of threat avoidance, maneuverability, collision avoidance, communication link, and arrival time. Thus, the swarm platforms can reach their mission areas safely and quickly. Fig. 1 shows a schematic dia-

gram of cooperative path planning of USV and UAV swarms.

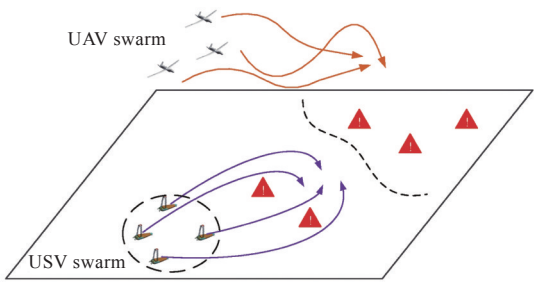


Fig. 1 Diagram of cooperative path planning for USV and UAV swarms

In cooperative path planning, both start and finish points of a path are known. Thus, a path can be generated by inserting a limited number of way points between the start and finish points. Suppose that the number of USVs and UAVs is $N_S$ and $N_A$, respectively. Then, the total number of platforms is $N = N_S + N_A$. All the platforms are numbered 1, 2, …, $N$ in the order from USVs to UAVs. For the platform $U_i$ ($i$ = 1, 2, …, $N$) numbered $i$ in the USV/UAV swarms, it has a given start point $\boldsymbol{p}_0^{(i)}$ and a finish point $\boldsymbol{p}_f^{(i)}$. Thus, the optimal-path set $\boldsymbol{P}^*$ of all the platforms in the cross-domain swarm, obtained through optimization under comprehensive consideration of threat information and multiple constraints, is as follows.

$$\begin{aligned}\boldsymbol{P}^*&=\{\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_N\}\\ \boldsymbol{P}_i&=\left(\boldsymbol{p}_0^{(i)},\boldsymbol{p}_1^{(i)},\boldsymbol{p}_2^{(i)},\cdots,\boldsymbol{p}_{n_i}^{(i)},\boldsymbol{p}_f^{(i)}\right)\end{aligned}\tag{1}$$

where $\boldsymbol{p}_j^{(i)}$ refers to waypoint coordinates, representing the $j^{\text{th}}$ waypoint of the platform $U_i$; $n_i$ is number of waypoints except start and finish points. Different paths may have different numbers of waypoints, which can be set as required. To solve cooperative path planning of USV/UAV swarms, we need to properly model scenes and constraints, construct a path optimization function that meets practical requirements, and solve the optimal-path set $\boldsymbol{P}^*$ through optimization algorithm.

## 1.2 Scene modeling

USV and UAV swarms move in a sea area and airspace, respectively. They need not only to restrict platform motion within designated boundaries of the sea area and airspace, but also to consider avoiding threats and obstacles wherein. It is necessary to establish a unified model to describe environmental space, which is a basis for subsequent cooperative path planning.

Polygonal and circular electronic geo-fences [9] are used in airspace modeling to describe boundary and threat information of airspace. Airspace boundaries are keep-in geo-fences in the air, limiting navigation of UAVs within specified airspace to ensure safety of the UAVs in mission execution. Due to enemy threats and bad weather, there are also no-fly zones for UAVs, namely, keep-out geo-fences in the air. For fuel consumption reduction, UAVs should avoid a high degree of maneuver in the way to their mission areas, and level flight is usually adopted. In this paper, it is assumed that UAVs are flying at the same height. Then, airspace is simplified to polygonal and circular areas at a specified height. First, a polygonal geo-fence is defined, with the following attributes and parameters:

$$\boldsymbol{B}=\{\kappa,h,m,(\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_m)\}\tag{2}$$

where $\kappa$ is the attribute of the geo-fence, $\kappa$=1 means a keep-in geo-fence and $\kappa$=0 means a keep-out geo-fence; $h$ is the height of the geo-fence; $m \geqslant 3$ is the number of polygonal vertices; $\boldsymbol{p}_i$ ($i$ = 1, 2, ..., $m$) refers to vertex coordinates of the geo-fence. Vertexes are usually arranged clockwise or counterclockwise, and the specific arrangement is determined as needed. Similarly, a circular geo-fence can be defined, with the main parameters being its center coordinates and radius, and the number of vertices being $m$ = 1. Airspace $\boldsymbol{E}_A$ is an area composed of one keep-in geo-fence and multiple keep-out geo-fences in the air:

$$\boldsymbol{E}_A=\{\boldsymbol{B}_{A0},\boldsymbol{B}_{A1},\boldsymbol{B}_{A2},\cdots\}\tag{3}$$

where $\boldsymbol{B}_{A0}$ is the keep-in geo-fence in the air; $\boldsymbol{B}_{A1}$, $\boldsymbol{B}_{A2}$, ... are keep-out geo-fences in the air. In practical application, geo-fences are for unified representation of mission-area boundaries, threat zones, and obstacles. During the setting of geo-fences, it is necessary to enlarge or narrow the areas to leave some space redundancy for navigational safety.

A sea area is usually modeled based on an electronic chart [10]. The common electronic chart in the ShapeFile format contains more than 30 layers. Layers with little useful information for mission execution can be ignored. Only those useful for mission execution, navigation, and control are necessarily analyzed and dealt with, including the layers of ocean/land, obstacles, waterways, regional boundaries, and landforms. In a two-dimensional chart, polygons are used to fit shapes of coastlines, islands, peninsulas, and beaches. This is equivalent to using multiple connected line segments to fit plane graphics with irregular edges. This modeling method is the same as that of airspace. A coastline is a

keep-in geo-fence on the sea, while threats and obstacles such as islands, peninsulas, and buoys are keep-out geo-fences. Therefore, a sea area $\boldsymbol{E}_{\mathrm{S}}$ is an area composed of one keep-in geo-fence and multiple keep-out geo-fences on the sea:

$$\boldsymbol{E}_{\mathrm{S}}=\{\boldsymbol{B}_{\mathrm{S0}},\boldsymbol{B}_{\mathrm{S1}},\boldsymbol{B}_{\mathrm{S2}},\cdots\} \tag{4}$$

where $\boldsymbol{B}_{\mathrm{S0}}$ is the keep-in geo-fence on the sea; $\boldsymbol{B}_{\mathrm{S1}}$, $\boldsymbol{B}_{\mathrm{S2}}$, … are keep-out geo-fences on the sea.

An electronic chart has accurate outline information of sets, with an island generally consisting of hundreds of edges and a coastline consisting of up to thousands or even tens of thousands of edges. The huge graphic information consumes a lot of computing resources, greatly reducing algorithm efficiency. Moreover, accurate information on coastline and island contours is of little significance for path planning. Therefore, the electronic chart needs to be preprocessed in practical application. Specifically, coastline and island boundaries are simplified in terms of coarse grains to generate scene models with few edges and simple geometric relationship as much as possible.

# 2 Optimal solution to cooperative path planning

In this section, a path optimization function is established with average travel time (ATT) as the index. In addition, cooperative path planning is converted into a constrained optimization problem under the consideration of multiple constraints like geo-fences, turning maneuverability, collision avoidance, and communication link. Due to its advantages in solving high-dimensional nonlinear optimization problems, a self-adaptive differential evolution (SaDE) algorithm is used to solve this optimization problem.

## 2.1 Path optimization function

Path planning mainly aims to enable cross-domain swarms to reach designated areas safely and quickly. The indexes to evaluate path performance usually include navigation distance, navigation time, and path smoothness. In terms of navigation distances, due to the different navigation speeds of various platforms, optimization of navigation distances is not suitable for heterogeneous platforms with different performances. In contrast, optimization of navigation time is more in line with the purpose of path planning, with a more specific goal. In terms of path smoothness, from the perspective of safe navigation, path smoothness only needs to meet the turning constraints of platforms. High path smoothness means a smaller turning angle of a platform and a shorter navigation time. This is consistent with the optimization goal of navigation time. Therefore, the use of navigation time as the optimization index can meet the requirement of fast arrival at designated areas in path planning. Moreover, navigational safety is realized through the design of constraints. Based on timeliness requirements of missions, considering the overall cooperativity of cross-domain unmanned swarms, this paper used the average navigation time of swarms as the optimization objective. The length of the navigation path of the platform $U_i$ is the sum of distances between adjacent waypoints, which is expressed by $D(\boldsymbol{P}_i)$:

$$D(\boldsymbol{P}_i)=\sum_{j=0}^{n_i}\left\|\boldsymbol{p}_j^{(i)}\boldsymbol{p}_{j+1}^{(i)}\right\| \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm of the vector. The navigation time of the platform $U_i$ is expressed by $T(\boldsymbol{P}_i)$:

$$T(\boldsymbol{P}_i)=\frac{D(\boldsymbol{P}_i)}{v_i} \tag{6}$$

where $v_i$ is the speed of the platform $U_i$. Each platform is usually required to arrive as quickly as possible to ensure overall cooperativity of cross-domain unmanned swarms and improve continuity of subsequent mission execution. Therefore, the average navigation time is taken as the optimization objective:

$$J(\boldsymbol{P})=\frac{1}{N}\sum_{i=1}^{N}T(\boldsymbol{P}_i) \tag{7}$$

where $\boldsymbol{P}=(\boldsymbol{P}_1,\boldsymbol{P}_2,\ldots,\boldsymbol{P}_N)$ is the set of all paths. A smaller $J(\boldsymbol{P})$ means a shorter average navigation time of swarms.

## 2.2 Constraints

In cross-domain cooperative path planning, factors that need to be comprehensively considered include threat/obstacle distribution in mission areas, maneuverability of unmanned platforms, collision avoidance among platforms in swarms, and continuous communication link. The above constraints are important factors affecting navigational safety and must be regarded as essential conditions for path planning. In this section, the above constraints are modeled as geo-fence, turning, collision-avoidance, and communication-link constraints.

### 2.2.1 Geo-fence constraints

In mission execution, all platforms should always

navigate safely within their keep-in geo-fences, and meanwhile, avoid entering keep-out geo-fences. For a circular geo-fence, the position relationship between a platform and the geo-fence can be obtained by only judging the distance between points and the circle center. For a polygonal geo-fence, its relationship with the path $\boldsymbol{P}_i$ of the platform $U_i$ can be judged by the existing classical geometric method[11]. The judgment method is not repeated here but only briefly described.

For a keep-out geo-fence $\boldsymbol{B}$, the given start point $\boldsymbol{p}_0^{(i)}$ and finish point $\boldsymbol{p}_\mathrm{f}^{(i)}$ are definitely not in $\boldsymbol{B}$. Therefore, if every segment $\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}$($k$ = 0, 1, 2, ..., $n_i$) of the path $\boldsymbol{P}_i$ does not intersect with the geo-fence $\boldsymbol{B}$, $\boldsymbol{P}_i$ will not intersect with $\boldsymbol{B}$, denoted as ($\boldsymbol{P}_i$ out $\{\boldsymbol{B}_1, \boldsymbol{B}_2, ...\}$). In order to judge whether a segment $\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}$ intersects with $\boldsymbol{B}$ or not, we only need to judge intersection relationships between all edges of $\boldsymbol{B}$ and $\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}$ one by one.

For a convex polygonal keep-in geo-fence $\boldsymbol{B}_0$, the start point $\boldsymbol{p}_0^{(i)}$ and finish point $\boldsymbol{p}_\mathrm{f}^{(i)}$ are definitely in $\boldsymbol{B}_0$. Therefore, if every waypoint $\boldsymbol{p}_k^{(i)}$ ($k$ = 0, 1, 2, ..., $n_i$) of the path $\boldsymbol{P}_i$ is located in $\boldsymbol{B}_0$, the path $\boldsymbol{P}_i$ will be in $\boldsymbol{B}_0$, denoted as ($\boldsymbol{P}_i$ in $\boldsymbol{B}_0$). By calculating the number of intersections through the ray method, we can judge whether $\boldsymbol{p}_k^{(i)}$ is in $\boldsymbol{B}_0$ or not. Detailed judgment will not be described here. For a concave polygonal keep-in geo-fence $\boldsymbol{B}_0$, the above method is no longer applicable. In such a case, an additional judgment condition is required: Every segment $\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}$ of the path $\boldsymbol{P}_i$ does not intersect with the keep-in geo-fence $\boldsymbol{B}_0$.

A path $\boldsymbol{P}_i$ of a USV is taken as an example. For a sea area $\boldsymbol{E}_\mathrm{S}$ = $\{\boldsymbol{B}_{\mathrm{S}0}, \boldsymbol{B}_{\mathrm{S}1}, \boldsymbol{B}_{\mathrm{S}2}, ...\}$, the geo-fence constraint $S_1(\boldsymbol{P}_i)$ can be expressed as

$$S_1(\boldsymbol{P}_i)=\begin{cases}1, & \text{if } (\boldsymbol{P}_i \text{ in } \boldsymbol{B}_{\mathrm{S}0})\,\&\,(\boldsymbol{P}_i \text{ out}\{\boldsymbol{B}_{\mathrm{S}1}, \boldsymbol{B}_{\mathrm{S}2}, \cdots\})\\ 0, & \text{else}\end{cases} \tag{8}$$

If and only if the path $\boldsymbol{P}_i$ is within the keep-in geo-fence $\boldsymbol{B}_{\mathrm{S}0}$ and without intersections with the keep-out geo-fences $\{\boldsymbol{B}_{\mathrm{S}1}, \boldsymbol{B}_{\mathrm{S}2}, ...\}$, the geo-fence constraint is satisfied, with $S_1(\boldsymbol{P}_i)$ =1; otherwise, the geo-fence constraint is not satisfied, with $S_1(\boldsymbol{P}_i)$= 0.

### 2.2.2 Turning constraints

In path planning, turning constraints of a platform need to be considered in order to generate a feasible path that satisfies the maneuverability of the platform. If three adjacent waypoints are close to each other, with a small angle between relevant segments, the actual turning radius may be less than the minimum turning radius of the platform. Thus, the turning-radius constraint limits the relative positions of three adjacent waypoints. Here, the criterion for judging the turning-radius constraint is given. Three adjacent waypoints $\boldsymbol{p}_{k-1}^{(i)}$, $\boldsymbol{p}_k^{(i)}$, and $\boldsymbol{p}_{k+1}^{(i)}$ are connected to form broken-line segments, and the maximum inscribed circle of the segments is shown in Fig. 2.

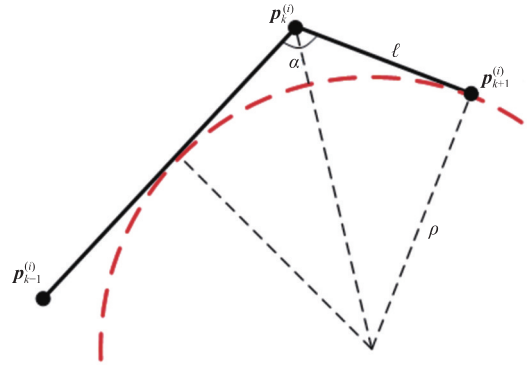


Fig. 2 Diagram of maximum inscribed circle during turning

The angle $\alpha$ between the two segments is given by

$$\alpha=\arccos\frac{\left\|\boldsymbol{p}_{k-1}^{(i)}\boldsymbol{p}_k^{(i)}\right\|^2+\left\|\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}\right\|^2-\left\|\boldsymbol{p}_{k-1}^{(i)}\boldsymbol{p}_{k+1}^{(i)}\right\|^2}{2\left\|\boldsymbol{p}_{k-1}^{(i)}\boldsymbol{p}_k^{(i)}\right\|\cdot\left\|\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}\right\|} \tag{9}$$

The length $\ell$ of the shorter one of the two segments is given by

$$\ell=\min\left\{\left\|\boldsymbol{p}_{k-1}^{(i)}\boldsymbol{p}_k^{(i)}\right\|,\left\|\boldsymbol{p}_k^{(i)}\boldsymbol{p}_{k+1}^{(i)}\right\|\right\} \tag{10}$$

The radius $\rho$ of the maximum inscribed circle is given by

$$\rho\left(\boldsymbol{p}_{k-1}^{(i)},\boldsymbol{p}_k^{(i)},\boldsymbol{p}_{k+1}^{(i)}\right)=\ell\tan\frac{\alpha}{2} \tag{11}$$

The radius of the maximum inscribed circle is the limit turning radius. If the minimum turning radius of a platform is greater than this limit, the platform fails to turn successfully. Suppose that the minimum turning radius of a platform is $R_\mathrm{t}$. Then, the turning constraint can be expressed as follows:

$$S_2\left(\boldsymbol{p}_{k-1}^{(i)},\boldsymbol{p}_k^{(i)},\boldsymbol{p}_{k+1}^{(i)}\right)=\begin{cases}1, & \rho\geqslant R_\mathrm{t}\\ 0, & \rho<R_\mathrm{t}\end{cases} \tag{12}$$

In the case of $S_2\left(\boldsymbol{p}_{k-1}^{(i)},\boldsymbol{p}_k^{(i)},\boldsymbol{p}_{k+1}^{(i)}\right)=1$, the turning-radius constraint is satisfied. In the case of $S_2\left(\boldsymbol{p}_{k-1}^{(i)},\boldsymbol{p}_k^{(i)},\boldsymbol{p}_{k+1}^{(i)}\right)=0$, the turning-radius constraint is not satisfied. In order to determine the turning-radius constraint of a complete path $\boldsymbol{P}_i$, we need to decompose the path into several segments, and then apply the above method to each segment. Finally, with the union of turning-radius constraints of all segments, the turning radius constraint $S_2(\boldsymbol{P}_i)$ of the whole path is obtained as follows.

$$S_2(\boldsymbol{P}_i)=\prod_{j=1}^{n_i} S_2\left(\boldsymbol{p}_{k-1}^{(i)},\boldsymbol{p}_k^{(i)},\boldsymbol{p}_{k+1}^{(i)}\right) \tag{13}$$

where $\Pi(\cdot)$ refers to multiplicative calculation. In other words, as long as the turning-radius constraint of any one segment is not satisfied, we will have $S_2(\boldsymbol{P}_i)=0$, namely that the turning-radius constraint of the whole path is not satisfied. Only when the turning-radius constraints of all the segments are satisfied, we will have $S_2(\boldsymbol{P}_i)=1$, namely that the turning-radius constraint of the whole path is satisfied.

### 2.2.3 Collision-avoidance constraints

Cross-domain cooperative relationships in terms of space and time should be simultaneously considered in the cooperative path planning of USV/UAV swarms. They are reflected as cooperative relationships of swarms in spatial and temporal domains, respectively. Time is coordinated by coordinating departure time of platforms. On this basis, spatial constraints are considered to judge collision-avoidance and communication-link constraints. For cooperative path planning, collision among multiple platforms is a likely problem. The research on collision avoidance focuses on homogeneous platforms. In other words, collision avoidance among USVs and that among UAVs are studied. For collision avoidance, it is necessary to ensure that distances among homogeneous platforms are always greater than the safety radius. However, it is impossible to judge collisions only from the spatial relationship of paths, and it is necessary to detect collisions in terms of spatio-temporal coupling. First, time sequences are determined, and simultaneous arrival of swarms is realized through alignment of arrival time and adjustment of departure time. The departure time $t_i$ of the platform $U_i$ is given by

$$t_i=-T(\boldsymbol{P}_i)+\max_{i=1,2,\cdots,N} T(\boldsymbol{P}_i) \tag{14}$$

Then, the spatial collision constraint is judged on the basis of time unification. A time sequence $k\Delta T$ ($k\geqslant 1$) of collision detection is obtained by time discretization with an interval of $\Delta T$. Fig. 3 shows a diagram of the time sequence.

At a collision-detection time point $k\Delta T$, given the motion velocity $v_i$ of the platform $U_i$, we can easily calculate the position of $U_i$ at $k\Delta T$, which is denoted as $\boldsymbol{P}_i[k]$ (where $k$ is the sequence number of the collision-detection point). For the homogeneous platforms $U_i$ and $U_j$, relative distances between collision-detection points with the same sequence number should meet the following constraint:

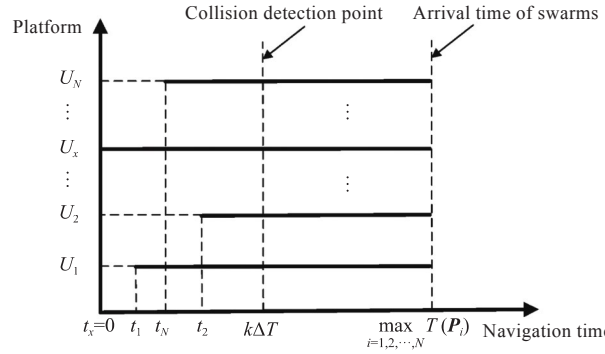


Fig. 3 Time sequence diagram of collision detection

$$\min_{j\neq i}\left\|\boldsymbol{p}_i[k]\boldsymbol{p}_j[k]\right\|\geqslant R_{\mathrm{S}}^{(i)}+R_{\mathrm{S}}^{(j)} \tag{15}$$

where $\boldsymbol{R}_{\mathrm{S}}^{(i)}$ and $\boldsymbol{R}_{\mathrm{S}}^{(j)}$ are anti-collision safety radii of the platforms $U_i$ and $U_j$, respectively; $\boldsymbol{p}_i[k]$ and $\boldsymbol{p}_j[k]$ are coordinates of the $k^{\mathrm{th}}$ collision-detection points of the platforms $U_i$ and $U_j$, respectively. For the paths of all homogeneous platforms, collision-detection points should be checked one by one according to the above-mentioned method. If all time-sequence detection points $\boldsymbol{p}_i[k]$ satisfy the collision-avoidance constraint, the planned path $\boldsymbol{P}_i$ will not collide with paths of other platforms. In this case, the collision-avoidance constraint is denoted as $S_3(\boldsymbol{P}_i)=1$. On the contrary, in the case of $S_3(\boldsymbol{P}_i)=0$, there is a collision among paths.

### 2.2.4 Communication-link constraints

The cooperative mission execution of USV/UAV swarms is based on a reliable communication network. Maintenance of good communication is a prerequisite for completing missions, especially the maintenance of shore-air-sea cross-domain joint communication under complex conditions. Different from homogeneous swarms, both UAV and USV swarms should maintain not only intra-swarm communication but also communication among cross-domain platforms. In other words, UAVs and USVs need to maintain certain spatial relationships to ensure communication links.

Due to the different communication capabilities of platforms, communication distances between each platform and other platforms are different. In view of continuous communication links, each platform is required to remain within the communication range all the time. Thus, stable communication of the whole cross-domain unmanned system can be guaranteed. Let the communication radius between $U_i$ and $U_j$ be $R_{\mathrm{C}}^{(ij)}$. The detection method of the communication-link constraint is the same as that of the collision-avoidance constraint, namely that the distance between detection points with the

same sequence number should be less than the communication radius:

$$\max_{j\neq i}\left\|\boldsymbol{p}_i[k]\boldsymbol{p}_j[k]\right\| \leqslant R_{\mathrm{C}}^{(ij)} \tag{16}$$

Generally, in the mission execution of an unmanned swarm on the sea, it is necessary to maintain good communication among some key nodes. Not all nodes are necessarily required to satisfy the communication-link constraint. If the communication link between $U_i$ and $U_j$ is unnecessary, just set the communication radius to $R_{\mathrm{C}}^{(ij)}=\infty$. When distances among all detection points satisfy the communication-link constraint, stable communication of the whole heterogeneous unmanned swarm can be guaranteed. In such a case, let the communication-link constraint be $S_4(\boldsymbol{P}_i)=1$, and otherwise, let it be $S_4(\boldsymbol{P}_i)=0$.

## 2.3 Optimization based on a SaDE algorithm

In view of the above multi-constraints, the cooperative path planning of a cross-domain unmanned swarm can be transformed into the following optimization problem:

$$\begin{gathered}\boldsymbol{P}^* = \arg\min J(\boldsymbol{P}) \\ \text{s.t. } S_j(\boldsymbol{P}_i)=1 \\ i=1,\cdots,N;\ j=1,\cdots,4\end{gathered} \tag{17}$$

where $\boldsymbol{P}^*$ is a set of optimal paths minimizing $J(\boldsymbol{P})$. The so-called constraint condition means that the paths of all platforms must meet the constraints of geo-fences, turning, collision avoidance, and communication link at the same time. In order for easy calculation, the following optimization problem is obtained by transforming the constraints into penalty functions.

$$\boldsymbol{P}^* = \arg\min\left[J(\boldsymbol{P})+M\sum_{i=1}^{N}\sum_{j=1}^{4}(1-S_j(\boldsymbol{P}_i))\right] \tag{18}$$

where $M$ is a penalty factor, and it is a large positive number.

When all the constraints are satisfied, the penalty term is valued at 0. When all the constraints are not satisfied, the penalty term has a maximum of $4MN$, with the severest punishment. When the constraints are partially satisfied, the penalty item is valued at an integral multiple of $M$. Its value is related to the number of constraints that are not satisfied. The advantage of the above design lies in gradients in the path cost function. As a result, individuals satisfying more constraints can be more likely reserved during optimization, thus gradually producing a feasible solution meeting the constraints. The penalty factor $M$ should be greater than the estimated $J(\boldsymbol{P})$ as far as possible to ensure the effectiveness of the penalty term in the optimization.

Due to a large number of platforms and waypoints, this belongs to a high-dimensional nonlinear optimization problem. In view of its advantages in solving high-dimensional nonlinear optimization problems[12], a SaDE algorithm is adopted in this paper. As a random heuristic search algorithm based on swarm intelligence theory, a differential evolution algorithm realizes optimization through cooperation and competition among individuals in a swarm. Moreover, with its unique memory, the algorithm is able to dynamically track the current search, which is convenient for search strategy adjustment. Thus, this algorithm is of high robustness and global optimization capabilities. The steps for solving the above path planning with the SaDE algorithm are as follows.

Step 1: Code the paths. Real number coding is adopted in the SaDE algorithm. One individual corresponds to one solution. Each individual is composed of several real-number bits, and each bit represents a variable. First, the number of both platforms and waypoints should be given to determine coding length. Suppose that the path of the platform $U_i$ contains $n_i$ waypoints. Given a constant height, the intermediate waypoints are two-dimensional variables. Then, the number of path variables of the platform $U_i$ is $2n_i$. Thus, the total coding length $D$ is the sum of the number of UAV and USV path variables:

$$D=2\left(\sum_{i=1}^{N_{\mathrm{A}}} n_i+\sum_{j=N_{\mathrm{A}}+1}^{N_{\mathrm{A}}+N_{\mathrm{S}}} n_j\right) \tag{19}$$

In the formula, the first term refers to the total number of path variables of $N_{\mathrm{A}}$ UAVs, while the second one refers to that of path variables of $N_{\mathrm{S}}$ USVs. Fig. 4 shows the coding method.

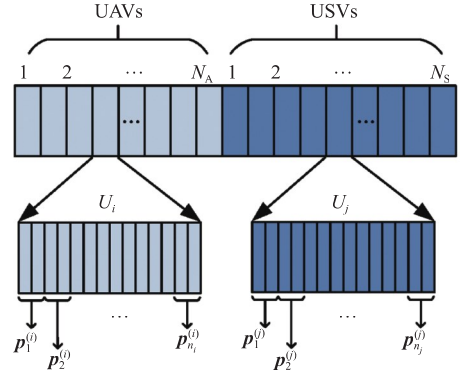


Fig. 4 Real number coding of SaDE algorithm

$\boldsymbol{p}_{n_i}^{(i)}$ is coded a $D$-dimensional vector in terms of

its path, and the dimension of search space increases with the increase in the number of platforms and waypoints. In the case of high environmental complexity, more waypoints are required to avoid obstacles, which will increase calculation time. In the case of low environmental complexity, fewer waypoints are required to shorten the calculation time of planning.

Step 2: Generate an initial population, and initialize the number of evolutionary generations to 1. According to geo-fence information, determine the planning space of waypoints and then upper and lower bounds of each variable. Set the population size to $NP$, and randomly generate $NP$ initial individuals that satisfy the upper and lower bound constraints.

Step 3: Calculate the fitness function. According to the coding method in Step 1, obtain the set $\boldsymbol{P}$ of paths of various platforms by decoding. Then, substitute it into the path cost function to calculate the fitness function value of each individual.

Step 4: Evolve the individuals. Adjust the mutation operator adaptively according to the number of evolutionary generations. Mutate and cross over the population, and process boundary conditions to obtain a temporary population. Conduct a "one-to-one" selection for corresponding individuals in the temporary and the original populations. Evolve the selected optimal individuals into a new population.

Step 5: Terminate the optimization. Judge whether the termination condition is satisfied or the maximum number of evolutionary generations is reached. If so, terminate the evolution. Otherwise, increase the current number of generations by 1, and return to Step 3.

In the optimization of cooperative path planning, the randomly generated initial path usually fails to meet the constraints, with a high fitness function value under the effects of the penalty term. In the initial stage of optimization, with the main purpose of finding "feasible paths" that meet the constraints, a great mutation operator is required to improve the diversity of the population and find as many "feasible paths" as possible to prevent "prematurity". In the later stage of optimization, with the main purpose of finding an optimal path in the "feasible path" set, a small mutation operator is required to improve the accuracy of the global optimal solution. Therefore, the SaDE algorithm with a self-adaptive mutation operator is adopted, and the self-adaptive mutation operator $F$ is designed as follows:

$$F=2^{\lambda}F_0,\ \ \lambda=\mathrm{e}^{1-\frac{G_\mathrm{m}}{G_\mathrm{m}+1-G}} \tag{20}$$

where $F_0$ is a constant mutation operator; $G_\mathrm{m}$ is the maximum number of evolutionary generations; $G$ is the current number of evolutionary generations; $\lambda$ is a parameter adaptively varying with $G$. In the SaDE algorithm, the mutation operator determines mutation amplitude of individuals and fineness of random mutation. In the early stage of evolution, the self-adaptive mutation operator is $2F_0$, which maintains the diversity of individuals. With the continuation of evolution, the mutation operator is gradually reduced to $F_0$ to retain good paths and improve the probability of finding a globally optimal path.

## 3 Simulation analysis

This paper verified the effectiveness of the proposed cooperative path planning method by simulation with a lake testing ground as the mission area, based on the scene model in Section 1.2. The height of the airspace was set to 500 m. The keep-in geo-fence in the air was a rectangular space above the sea area. In addition, two keep-out geo-fences in the air and three on the sea were set randomly. It was supposed that UAVs took off to the specified height in the specified area. Then, the coordinate points of UAVs after their take-offs were set as start points of the UAVs, and the arrangement positions of USVs on the shore were set as start points of the USVs.

In the simulation, seven platforms were set in the cross-domain swarm, denoted as $U_1$, $U_2$, … , $U_7$. Specifically, there were three UAVs denoted as $U_1$−$U_3$, and four USVs denoted as $U_4$−$U_7$. Table 1 and Table 2 show the initial states of UAVs and USVs, respectively. The safety radius among homogeneous platforms was 100 m. The maximum communication distance among UAVs was 12 km, and that among USVs was 5 km. The maximum communication distance between a UAV and a USV was 10 km.

According to the number of platforms and waypoints, the variable length in real number coding of the SaDE algorithm is 20. In the SaDE algorithm, we set the number of individuals in the population to 200, the maximum number of evolutionary gener-

**Table 1 Initial status information of UAV**

| Platform | Coordinates/m | | Speed /(m·s$^{-1}$) | Minimum turning radius/m | Number of waypoints |
|---|---|---|---|---|---|
| | Start point | Finish point | | | |
| $U_1$ | (2 923，14 673) | (12 976，6 012) | 20 | 55 | 1 |
| $U_2$ | (1 429，13 945) | (12 427，7 843) | 23 | 75 | 2 |
| $U_3$ | (2 016，12 864) | (11 789，5 122) | 25 | 100 | 1 |

**Table 2 Initial status information of USV**

| Platform | Coordinates/m | | Speed /kn | Minimum turning radius/m | Number of waypoints |
|---|---|---|---|---|---|
| | Start point | Finish point | | | |
| $U_4$ | (3 317, 4 994) | (9 785, 8 685) | 30 | 200 | 1 |
| $U_5$ | (3 170, 5 415) | (9 035, 8 247) | 25 | 140 | 2 |
| $U_6$ | (2 822, 6 020) | (11 330, 11 900) | 30 | 200 | 1 |
| $U_7$ | (2 794, 7 129) | (10 340, 7 781) | 35 | 300 | 2 |

ations to 500, the constant mutation factor to $F_0$ = 0.3, and the crossover factor to 0.1. In addition, the penalty factor was set to $M = 10^3$, and the time interval for discretization was set to $\Delta T$ =5 s. Figs. 5−9 show the simulation results.

Fig. 5 shows the simulated keep-in and keep-out geo-fences both in the air and on the sea, as well as the planned paths of the USV/UAV swarms. UAVs fly in the airspace at a height of 500 m, while USVs sail on the sea. The start points of all paths are located near the shore on the left side, and the finish points are distributed in the mission area on the far right side. In Fig. 5, black dotted lines refer to keep-in geo-fences in the air and on the sea, and all paths are kept within the geo-fences. Red and blue solid lines refer to keep-out geo-fences in the air and on the sea, respectively. In order for minimum navigation time, some of the paths are close or even almost "tangent" to the keep-out geo-fences. This will not affect navigational safety, due to redundant distances reserved in the design of geo-fences. The simulation verifies the effectiveness of the designed geo-fence constraints.

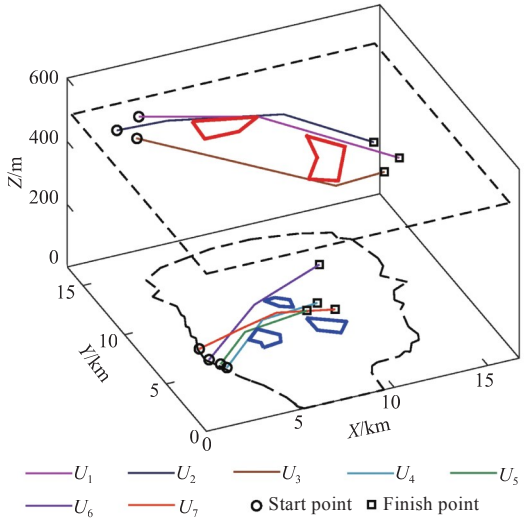


Fig. 5 Results of path planning of USV and UAV swarms

Fig. 6 shows the variation of path optimization function values with the number of iterations. In the initial stage of optimization, under the effects of the penalty function, the function values are high, and all the solutions in such a case fail to meet the constraints. From the first to the 100th generation, the main purpose is to find feasible solutions meeting the constraints. In such cases, the function values are gradually reduced to less than 1 000, and the solutions can meet the constraints. This verifies the reasonableness of the designed penalty function. After the 100th generation, the main purpose is to find the solution minimizing optimization function values. Finally, in the 500th generation, the fitness function value converges to 565.8, namely that the average navigation time of the USV/UAV swarms is 568.5 s.

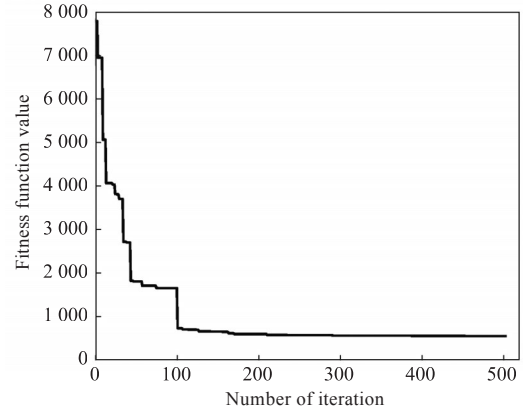


Fig. 6 Optimization process of the path optimization function

Fig. 7 shows the navigation time sequence of USV/UAV swarm platforms. In the figure, the horizontal axis refers to sailing time, while the vertical one refers to platform number. Circles refer to the departure time of platforms, while squares refer to the arrival time of platforms. With the departure time of the platform $U_1$ having the longest sailing time as the reference, the departure time of the platforms $U_2$−$U_7$ is 101, 150.7, 175.5, 154.2, 7.3, and 244.7 s, respectively.

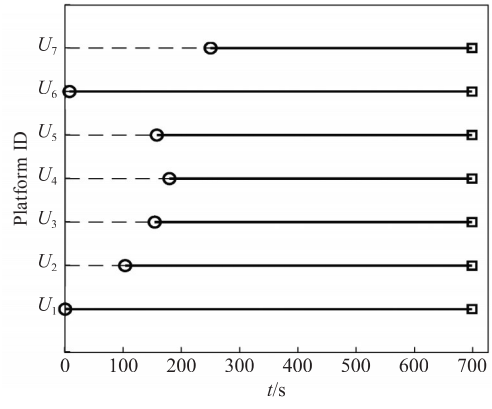


Fig. 7 Navigation sequence diagram of USV/UAV swarms

In order to verify the effectiveness of collision-avoidance and communication-link constraints,

Figs. 8−9 show real-time distance variations among platforms of the USV/UAV swarms. As all safety radii set in this section are 100 m, safe distances are 200 m. From the figure, the distances among the platforms are always greater than the safety value, which verifies the effectiveness of the collision-avoidance constraint.

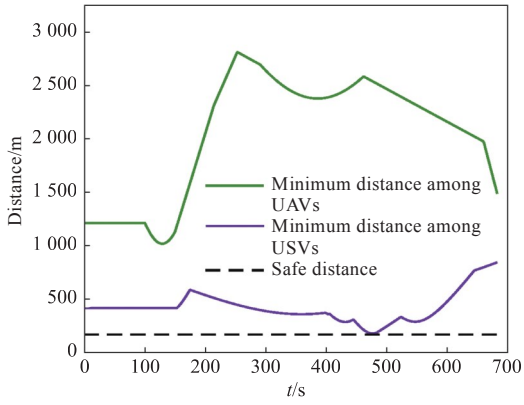


Fig. 8 Variation curves of the minimum distance between platforms with time

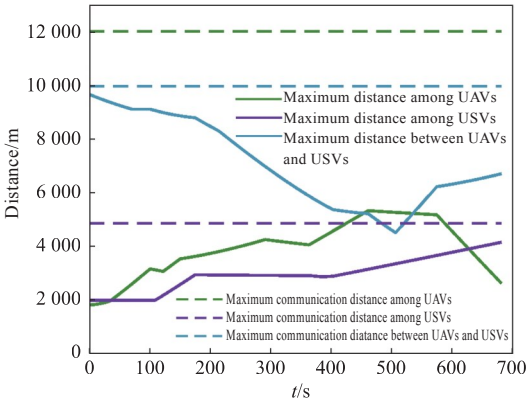


Fig. 9 Variation curves of maximum distance between platforms with time

Fig. 9 shows the maximum distances and relevant upper limits among UAVs, among USVs, and between UAVs and USVs. Specifically, solid lines refer to real-time maximum distances among the platforms, while dotted ones refer to maximum communication distances. From the figure, the real-time distances of all platforms are smaller than their maximum communication distances. This ensures the communication link between the USV/UAV swarms, verifying the effectiveness of the communication-link constraint mentioned above.

## 4 Conclusions

In this paper, a cooperative path planning method of USV and UAV swarms based on a self-adaptive differential evolution algorithm was proposed. This method realizes cooperative path planning under multiple constraints and ensures safe navigation and continuous communication link in an obstacle-filled environment. Thus, it is of certain application value. Obstacle avoidance, platform collision avoidance, and communication link were modeled as multiple constraints through geo-fence and time-sequence methods. In addition, path planning was transformed into an unconstrained optimization problem through the penalty function method. An SaDE algorithm was used for optimization. It can maintain diversity in the initial stage of search and accuracy in the later stage, thus ensuring the solving of an optimal path. However, with the increase in the number of waypoints, the calculation time will continue to increase. Limited by the timeliness of solving, the proposed method is only applicable to offline path planning. Real-time online planning will be deeply studied in future work.

## References

[1] ZHANG W M, HUANG S P, HUANG J C, et al. Analysis on multi-domain operation and its command and control problems [J]. Command Information System and Technology, 2020, 11 (1): 1−6 (in Chinese).

[2] ZHANG W D, LIU X C, HAN P. Progress and challenges of overwater unmanned systems [J]. Acta Automatica Sinica, 2020, 46 (5): 847−857 (in Chinese).

[3] NIU H L, SAVVARIS A, TSOURDOS A, et al. Voronoi-visibility roadmap-based path planning algorithm for unmanned surface vehicles [J]. The Journal of Navigation, 2019, 72 (4): 850−874.

[4] YANG J M, TSENG C M, TSENG P S. Path planning on satellite images for unmanned surface vehicles [J]. International Journal of Naval Architecture and Ocean Engineering, 2015, 7 (1): 87−99.

[5] OUYANG Z L, WANG H D, HUANG Y, et al. Path planning technologies for USV formation based on improved RRT [J]. Chinese Journal of Ship Research, 2020,15 (3): 18−24 (in Chinese).

[6] ZHOU X Y, WU P, ZHANG H F, et al. Learn to navigate: cooperative path planning for unmanned surface vehicles using deep reinforcement learning [J]. IEEE Access, 2019, 7: 165262−165278.

[7] BABEL L. Coordinated target assignment and UAV path planning with timing constraints [J]. Journal of Intelligent & Robotic Systems, 2019, 94 (3/4): 857−869.

[8] WU Y. Coordinated path planning for an unmanned aerial-aquatic vehicle (UAAV) and an autonomous underwater vehicle (AUV)in an underwater target strike mission [J]. Ocean Engineering, 2019, 182: 162−173.

[9] CHO J, YOON Y. How to assess the capacity of urban airspace: a topological approach using keep-in and keep-out geofence [J]. Transportation Research Part C: Emerging Technologies, 2018, 92: 137−149.

[10] ESRI. ESRI shape file technical description-an ESRI white paper[R]. Redlands, CA, USA: Environmental

Systems Research Institute, Inc, 1998.
[11] HORMANN K, AGATHOS A. The point in polygon problem for arbitrary polygons [J]. Computational Geometry, 2001, 20 (3): 131−144.
[12] QIN A K, SUGANTHAN P N. Self-adaptive differential evolution algorithm for numerical optimization [C]//2005 IEEE Congress on Evolutionary Computation. Edinburgh, Scotland, UK: IEEE, 2005.

# 多约束条件下无人艇和无人机集群协同航迹规划

侯岳奇[1], 陶浩[2], 龚俊斌[2], 梁晓龙[*1], 张诺[3]
1 空军工程大学 空管领航学院,陕西 西安 710051
2 中国舰船研究设计中心,湖北 武汉 430064
3 中国人民解放军31005部队,北京 100094

**摘　要**:[**目的**]为实现海上无人集群在执行任务过程中的安全航行和通信保持,开展无人艇(USV)和无人机(UAV)集群协同航迹规划问题的研究。[**方法**]采用禁入和禁出地理围栏进行场景建模,将规避威胁和障碍问题转化为地理围栏约束。针对平台之间的碰撞冲突和通信连接问题,提出基于时序检测的碰撞冲突和通信保持约束判断准则。以集群平均航行时间为航迹优化函数,将多约束条件转化为惩罚函数,采用自适应差分进化算法进行优化求解。[**结果**]仿真结果表明,所提方法能够在威胁和障碍环境中保持无人艇和无人机集群的安全航行和通信连接,并在满足多约束的条件下实现集群平均航行时间最短。[**结论**]该方法可用于海上无人集群面对威胁和障碍环境时的离线航迹规划,具有一定的应用价值。
**关键词**:无人艇;无人机;海上无人集群;协同航迹规划

[*Continued from page* 11]

[19] MALLIPEDDI R, LEE M. An evolving surrogate modelbased differential evolution algorithm [J]. Applied Soft Computing, 2015, 34: 770−787.
[20] STEIN M L. A kernel approximation to the kriging predictor of a spatial process [J]. Annals of the Institute of Statistical Mathematics, 1991, 43 (1): 61−75.

# 基于机器学习的实海域无人艇避碰算法智能演进方法

楼建坤[1,2], 王鸿东[*1,2], 王检耀[1,2], 易宏[1,2]
1 上海交通大学 海洋智能装备与系统教育部重点实验室,上海 200240
2 上海交通大学 海洋工程国家重点实验室,上海 200240

**摘　要**:[**目的**]无人艇(USV)效能是指在给定时间内的特定海域完成指定任务的能力,是多层次多技术节点耦合作用的结果,而针对单一技术节点的传统优化方法,对无人艇效能的提升效果有限。[**方法**]针对无人艇自主系统的特点,从智能算法的角度,提出无人艇智能演进的2种主要形式:一是算法函数;二是算法参数。在此基础上,给出基于机器学习的无人艇智能演进方法,设计一种可演进的无人艇自主系统控制体系架构,并在实海域测试。[**结果**]以无人艇避碰算法为例,基于实海域测试结果,初步验证了所提方法在提升无人艇效能方面的可行性与有效性。[**结论**]基于机器学习的无人艇智能演进方法是持续提升无人艇效能的有效途径,具有较高研究价值和应用意义。
**关键词**:无人艇;智能演进;实海域测试;机器学习