2021 年美国数学建模竞赛电子科技大学模拟赛

承诺书

我们仔细阅读了数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮 件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料(包括网上查到的资料)，必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们的题目编号是（填写：A 或者 B）： B

我们报名**队伍**号是： H090

参赛队员姓名学号（**中文填写打印并签名**）：

1. 贺奕豪 2018011205026
2. 张露丹 2018081304027
3. 王晨昊 2018011202003

指导教师或指导教师组负责人（有的话填写）：

是否愿意参加 2021 年美国赛（是，否）： 是

日期：2021 年 11 月 23 日

Summary

With the popularization of unmanned equipment, unmanned ships have become increasingly widely used. Research on the actions of naval unmanned ships has become very popular, and the penetration and interception of unmanned ships has become an important research topic. This article solves the problem of optimizing the offensive and defensive strategy of the red and blue USVs.

For the first question, we established a **risk-related model** of the blue side, which has 2 risk factors. The **segmented negative exponential function** and **quadratic function** are respectively used to constrain the blue USV y-direction speed. The red team's defense problems can be divided into 2: Red USVs cluster adopts best formations to maximize the defense range; the red team's overall movement strategy. For the formation plan, the **Monte Carlo method** is used to calculate the maximum effective area of the red team that can intercept the blue USV, which is equivalent to a circle with a **radius of 123.5m**. For the red side's movement strategy, a strategy of directly pointing to the blue side was initially adopted. The **state space transition model (S-S model)** is used to simulate and calculate the position and direction angle of the red and blue USV at different moments. The **Infinitesimal method** is used to approximate the motion to a uniform linear motion in each small period of time. Monte Carlo method is used to determine the initial position that makes the blue USV successfully penetrate. The result is that the blue USV begin from the point near the **initial point (7000,3500)** will be intercepted; the blue USV begin from the other positions will be successfully pass through, and the **penetration rate is 0.976.**

For the second question, under the condition that the red team adopts the direct-pointing strategy, the direct search method is used to find **Mmin=1400m**. The **simulated annealing algorithm (S-A Algorithm**) is used to optimize the blue USV's risk-related model parameters, and the fastest passing time of the blue party is **412s**. Then the red movement strategy is improved, and the **y-direction direct alignment method** is adopted, finding **Mmax=7800m** under this strategy of the red side.

For the third problem, we solved the USV quantity allocation model and cluster movement strategy respectively. Based on the latter, we established an **independent direct interception model** and a **regional defense interception model**, respectively discussed the interception effects of the specific conditions of the problem, and explored the limits of their capabilities. Finally got **Mmax=15000m**.

Finally, in the verification part of the model, we simulated the whole process of each stage through MATLAB simulation, adjusted individual problems of the model, solved the problems of boundary judgment and direction change, and made the model practical.

KEYWORDS: Risk-Related Model; Monte Carlo method; S-S Model; S-A Algorithm; Regional Defense Interception Model

Contents

1 Introduction

1.1 Background

In addition to civilian use, the use of machines and unmanned technology in military warfare is also very important. Due to the development of swarm technology, swarms of unmanned equipment can often accomplish the same combat mission at very low cost, so we often discuss the relationship and connection between UAVs and fleets. Among them, the use of clusters of unmanned ships to patrol blocked seas is an important research direction. This model is relatively complicated. In the initial stage of research, we often use a limited individual to replace a large group, which is convenient for simulation with MATLAB. For example, in this question, five red USVs were used to replace a cluster, and the sea area was blocked, and then a blue USV was used to try to break through the blockade and form a confrontation. We have respectively established the best interception model for the red side and the best penetration model for the blue side to describe this process more realistically.

1.2 Restatement of problems

A blue USV hopes to break through the red side's interception and reach the destination on the other side safely within 700s, while two red carriers releases of 10 USVs in two clusters hope to intercept the blue side's infiltration. The maneuvering range of both parties is limited to a rectangular area of length L=10000m and width M. The data of all vehicles on both sides are shown in Table 1-1. The existence of the turning radius makes the trajectory of all motions smooth and steerable. The location information of both sides are exposed to each other, so that both parties can make adjustments to each other's real-time situation. The USV of the red side was carried by the USV carrier. In some cases, the carrier can selectively release different numbers of USV under different conditions. The distance between any two USVs on the red side is not less than 30m, and there are at least two friendly forces within 100m of each USV. The USV carrier must be close to 2000m with at least one USV, and cannot be close to 1000m with the blue USV. The USV formation can be adjusted at any time.

Table1-1 Parameters of both vehicles

Units	Speed	Minimum radius of curvature
Blue USV	25	100
Red USV	20	80
Red transport ship	<16	500

The established mathematical model needs to complete the following tasks:

(1) Determine the state space model of both USVs [1], and clarify the input and output parameters and states.

(2) Clarify the influence of the starting positions of the blue USV and the red mothership on the confrontation process.

(3) Establish the blue's optimal penetration strategy and the red's optimal interception strategy.

(4) Study the optimization and improvement of the model for the Red carrier to release the multi-wave USV at different times and locations.

1.3 Overview of Our Work

In our design project, the **first step** is to build a state space model, in which the input parameters of the blue and red USVs correspond to the position of each other's USV at each moment, and the output parameters correspond to their own movement at the next moment. direction. The **second step** is to design the most scientific and reasonable USV layout plan for the red side USV, and obtain its equivalent simplified model at the macro battlefield scale. The **third step** is to construct the best penetration and interception strategies for the blue and red sides. This part is mainly based on the more practical and simple algorithms obtained from mechanism analysis and empirical reasoning, and discussed with the different input and output conditions at both ends. A more practical strategy. The **fourth step** is for the first two questions of the question, the different initial positions of the two parties are substituted into the calculation and analysis, and several extreme initial position classification situations are obtained.

The **last step** is to extend the previous model to a two-wave USV and establish an independent two-wave interception model and a regional defense model.

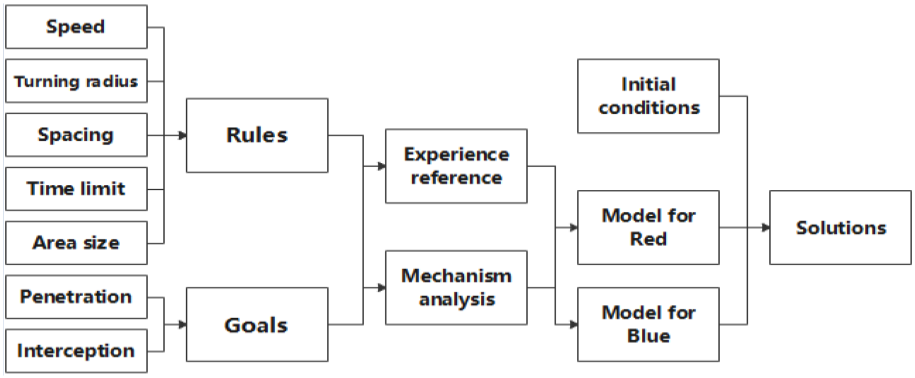


Figure 1-1 Framework of workflow

2 Assumptions

The time for the aircraft carrier to release the USV is negligible.

The size of the internal relative position of the USV cluster is negligible compared to the size of the entire map.

Regardless of the speed of the USV transport ship, it must be limited to a minimum radius of curvature of 500m.

Assuming that the distance between the blue USV and the red carrier is too close, and the distance between the red carrier and the cluster is too far, in both cases it is considered that the blue side successfully passed through

3 Symbols & Notations

Symbols	Notations
D_{th}	Interception distance
D_{BR}	Distance between the two parties
P_D	Probability of discovery
dd	Direction angle change
dn	Direction angle
$X_{ab} or Y_{ab}$	Coordinate difference

kd	Deceleration factor
Dab	Distance
vRc	Speed of the carrier
ke	Border risk factor

4 Model Establishment in Problem 1&2

4.1 State space model of USV

Both the red side and the blue side have determined internal decision-making rules. Since the position information of the red and blue parties are communicated in real time, the change of the state at each time point will affect the new decision of the next state. Therefore, we can build a state-space model for each point in time. In order to facilitate actual model calculations, we use discrete moments to describe changes in time. As long as the sample is relatively small, we can approximate this process as a continuous change. In the following formula, dx/dt is the state difference, x is the state vector, u is the input vector, and y is the output vector. These vectors are multi-dimensional, and each category can have different units.

$$\frac{dx}{dt} = Ax + bu \tag{4-1}$$

$$\mathrm{y} = \mathrm{Cx} + \mathrm{Du} \tag{4-2}$$

First of all, we need to determine what the input variables, state variables, state difference components and output variables are. By comparing the conditions and analyzing the mechanism of the USV movement of the two parties, we can determine that the input of the USV of the blue side is the position of the USV of the red side at the current moment, the state quantity is the position of the USV at the current moment, and the state difference is the current moment and the previous one. The amount of position change at the moment, and the output variable is the direction of the speed at the next moment. Drive this model to find that the internal function output at the next moment is the best avoidance strategy for the blue USV.

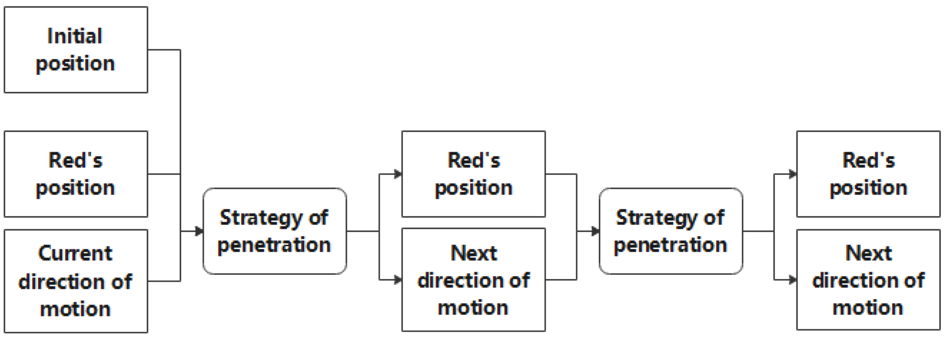


Figure 4-2 Blue USV algorithm

Although the USV parameters of the red side are slightly different from those of the blue side, its state space model also has a similar structure. Its input is the position of the blue USV at the current moment, the state quantity is the position of its own USV at the current moment, the state difference is also the position change between the current moment and the previous moment, and the output variable is also the direction of the speed at the next moment. Among them, the internally calculated output law is the best interception strategy for the red USV.

4.2 The best interception strategy for Red USV

Since the red team's interception is carried out in clusters, we need to determine the layout of the USV in each cluster, and then use each cluster as a whole to study the entire interception process with the cluster's pursuit of the blue USV.

From the perspective of the entire process, the USV layout inside the red cluster is negligible compared to the scale of the entire theater map, so we approximate the establishment of the two models to be independent of each other. Let's first discuss the layout of the Red Side USV.

The best layout inside the USV cluster

The commonly used individual search and interception model is widely used in maritime interception search [2], and it is generally believed that the probability of finding a target is:

$$P_D \leq \frac{2D_{th} \times E_{(S)}}{V_b \times L} \leq \frac{2D_{th}}{L}\sqrt{1+\frac{{V_r}^2}{{V_b}^2}} \tag{4-3}$$

For the round-trip search and discovery probability of turning on the patrol line boundary line:

$$P_D = \frac{2D_{th}}{L}\sqrt{1+\frac{{V_r}^2}{{V_b}^2}} - \frac{{D_{th}}^2}{{D_{BR}}^2}[1+\frac{{V_r}^2}{{V_b}^2}] \tag{4-4}$$

In the discussion of the first and second questions, each USV cluster is composed of five groups. We need to obtain a set of placement methods with the highest interception efficiency. Of course, these placement methods must meet the relationship between the relative positions of the red USV in the question, that is, the distance between any two is greater than 30m, and there are two within 100m. Friendly forces. According to the experience of mathematical set, we can get the common tight patterns of five vertices as follows:

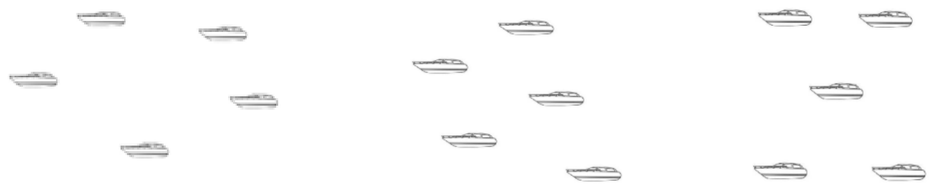


Figure 4-3 Three types placement of 5USVs

To effectively intercept the blue USV, the red side needs to have at least two USVs covering him within a radius of 100. Therefore, the truly effective interception surface is to draw a circle with a radius of 100m at these five points, with at least two layers in it. Overlapping area. Below we respectively deal with the above three placement methods as shown below to show their effective interception area.

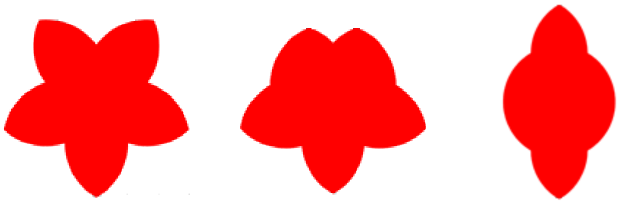


Figure 4-4 Placement with valid area

In the actual situation, the effective interception area is the real part of the interception mission. Therefore, we believe that the best placement method should be the placement with the largest effective interception area. Below we will take the above three typical placements as examples to analyze the effective area calculation.

As mentioned in the previous article, due to the large difference in scale between the actions inside the USV cluster and the whole as the object, the shape in the placement method mentioned here has a negligible effect on the whole, so we need to study in the USV cluster It is always equivalent to a circle in, which makes it much easier to study the constraint conditions of the distance. Take a regular pentagon as an example. The equivalent effect is shown in the figure below:

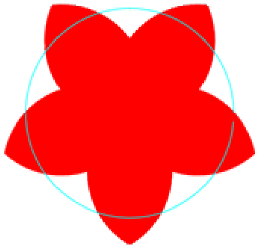


Figure 4-5 Valid area with equivalent radius

Because the above graphics are relatively complicated, it is a very troublesome process to directly obtain the function expressions of these boundary lines to obtain the integral. Therefore, on this issue, we use Monte Carlo method to approximate the effective area. In this process, it is only necessary to judge whether the position of the input point is to be covered by two center points. We substitute the specific values of the constraint conditions in the title into the calculation, and finally calculate the equivalent intercept radius of the typical placement mode in 3 when a cluster is composed of 5 USVs, as shown in the following table:

Table4-1 Valid area of different placement methods

Placement	Regular Pentagon	Olympic Five Rings	Cross
Effective interception radius/m	123.5	119.8	115.6

Based on the above results, we choose to arrange a regular pentagon in the USV cluster to achieve the best, and the equivalent maximum effective radius is 123.5m.

4.2.1 Interception method towards the target

Because it is not easy for the red side to predict the direction of the blue side, a basic interception strategy is that the red side's direction of movement is always facing the blue side's position. However, due to the influence of the turning radius, the angle of each turn cannot be too large, and the direction cannot be accurately aligned with the blue side. The following is to derive the relationship between the minimum turning radius and the turning angle within ***dt***.

See the figure below, the current position coordinate point ***p***, the moving direction angle ***d*** (unit: rad, range: 0-2π). After the time ***dt*** has elapsed, the next position coordinate ***pn***, the direction angle ***dn***. ***d*** is perpendicular to $\overline{pO}$, ***dn*** is perpendicular to the line segment $\overline{\boldsymbol{pnO}}$**.** Direction angle change ***dd***

$$dd = dn - d \tag{4-5}$$

From the vertical angle relationship,

$$dd = \angle pnOp \tag{4-6}$$

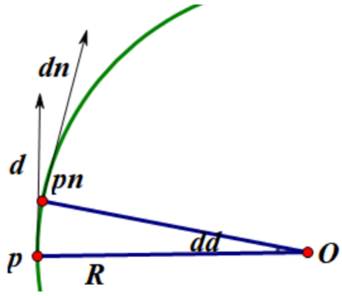


Figure 4-6 Geometric relationship between turning radius and angle

Because ***dt*** is very small, the idea of Infinitesimal method is adopted, and it is considered that ***dt*** is a uniform linear motion, and the displacement after ***dt*** time

$$\overline{ppn} = v \times dt \tag{4-7}$$

and

$$\widehat{ppn} = \angle pn0p \times R \tag{4-8}$$

$$\widehat{ppn} \approx \overline{ppn} \tag{4-9}$$

thus

$$\mathrm{dd} = \angle \mathrm{pn0p} = \frac{v \times dt}{R} \tag{4-10}$$

$$dd_{max} = \frac{v \times dt}{R_{min}} \tag{4-11}$$

The above formula can be used to calculate the maximum angle ***ddmax*** turned within ***dt*** from the speed and the minimum turning radius ***Rmin***

When the target position of the blue team changes too fast and the red team hopes that the direction angle changes greatly, the maximum can only change ***ddmax***. Therefore, the red side's interception effect is limited.

4.2.2 Improved interception algorithm based on direct interception

In the simulation test of the interception method towards the target, we found that the interception effect is not good at certain times. Due to the speed advantage of the blue USV, once the distance in the Y direction is increased, the red USV will have difficulty catching up with the blue. USV. Therefore, we concluded that chasing the specific position of the blue USV during the interception is not so important, and more importantly, we must pay attention to tracking the Y-axis position of the blue USV. This is based on the fact that the blue USV always aims to advance in the x direction, so the intercepting party will have the highest adjustment efficiency in the other's vertical movement direction.

In this algorithm, the red USV group always pays attention to the difference in the Y-direction distance between itself and the blue USV from the beginning. Once the Y-axis distance between them is greater than a certain threshold, the red USV group will approach the enemy along the Y direction. When the distance between them in the Y direction is less than the threshold again, the USV group changes the direction of movement, tending to directly intercept the direction. The judgment formula is as follows:

$$Rn = \begin{cases} \frac{\pi}{2} & Y_{pB} > Y_{pR} + D_{rush} \\ \frac{3\pi}{2} & Y_{pB} < Y_{pR} - D_{rush} \\ \arctan\left(\frac{Y_{pB}-Y_{pR}}{X_{pB}-X_{pR}}\right) & Y_{pR} - D_{rush} < Y_{pB} < Y_{pR} + D_{rush} \end{cases} \quad (4\text{-}12)$$

Of course, as before, the actual direction angle change also depends on the minimum turning radius, that is, the maximum steering angle. The movement direction dRn of the next moment is affected by the movement direction dR of the previous moment:
When

$$|dRn - dR| > ddr_{max} \quad (4\text{-}13)$$

$$dRn = \begin{cases} dR + ddr_{max} & dRn > dR \\ dR - ddr_{max} & dRn < dR \end{cases} \quad (4\text{-}14)$$

To sum up, the red USV has two interception modes. The first mode is to always adjust the position in the Y direction according to the above-mentioned rules, and not rush to prevent the blue USV from breaking through. Until the distance between the two parties in the Y direction is less than a certain threshold, the red USV enters the second mode, that is, the interception method toward the target is adopted, so that the blue USV can be captured as much as possible. These two tracking and blocking modes are not in a sequential relationship, and can be switched with each other. For example, after the red side aligns the Y-direction coordinates of both sides, enter the second mode. This is because the blue USV may extend the distance between the two sides in the Y-direction again due to the avoidance strategy. At this time, the red side's USV should not be used again. I have been chasing, but I need to adjust my y-direction coordinates in time and align it again, so I will switch to the first interception mode again. In the subsequent simulation process, we can find that, due to the alternating of the two interception modes, the interception trajectory of the red USV often has many similar steps.
Such an improved algorithm combines the advantages of the previous algorithm. It not only makes full use of its own moving speed, but also uses its own position advantage more efficiently. To a certain extent, it realizes the prediction of the enemy's USV movement. The system becomes more intelligent.

4.2.3 The mobile strategy model of the Red Mothership

Due to the various constraints of the mothership, it often affects the interception strategy and interception effect of the Red USV to a large extent. Therefore, optimizing a reasonable mothership trajectory can guarantee the smooth progress of the interception to the greatest extent.
In most cases, the requirements for the transport ship are mainly to keep the mothership within 2km of the USV cluster (because the internal size of the cluster is too small, it is approximately the distance to the center of the cluster), so we define the red transport ship's The normal operation mode is to run in the direction of the friendly USV, and its speed is also determined by the speed between the two, which can ensure that the distance between them is as stable as possible within a certain range. The function is expressed as follows:

$$\mathrm{Dr} = |\mathrm{pRc} - \mathrm{pR}| \quad (4\text{-}15)$$

$$\mathrm{vRc} = \frac{vRc_{max} - vRc0}{1 + e^{-\frac{Dr}{100}+5}} + vRc0 \quad (4\text{-}16)$$

pRc is the position coordinate of the red transport ship, pR is the position coordinate of the center point of the red cluster, Dr is the distance between the two, vRcmax is the maximum speed of the red transport ship, and vRc0 is its minimum speed.
We require that as the distance gets closer, the speed of the transport ship needs to decrease, and as the distance gets away, the speed of the transport ship needs to increase. The correlation function is as follows:

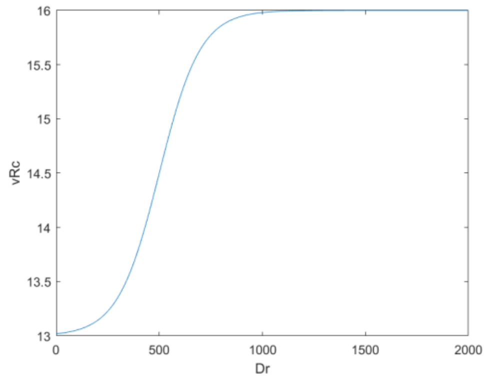


Figure 4-7 Transport ship speed function1

On the other hand, as the blue USV approaches, the red transport ship must maintain a distance of more than 1km from it. Therefore, the mothership needs to design a reasonable avoidance strategy to comprehensively consider the follow-up of the friendly USV and the enemy USV. escape.
When the distance between the blue side and the red side transport ship is less than a certain value, the red side transport ship must move away from the blue side. However, it has been experimentally verified that if it moves away at a greater speed, due to the large minimum turning radius of the red transport ship, the red transport ship will soon move away from the red USV cluster. Therefore, after the red transport ship is closer to the blue side, it will decelerate while moving away, multiplied by a deceleration coefficient kd, and its relationship with Db (the distance between the blue side and the red side transport ship) is shown below.

$$\mathrm{kd} = \begin{cases} \frac{-1}{(Dbth-1000)^2 \times (Db-1000) \times (Db+1000-2Dbth)} & Db < Dbth \\ 1 & Db \geq Dbth \end{cases} \tag{4-17}$$

The correlation function is as shown in the figure below, when Db>Dbt, it does not slow down until Db<Dbth.

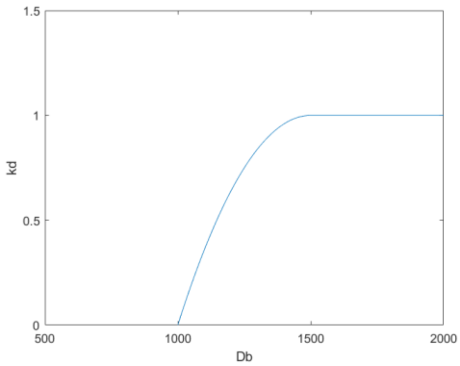


Figure 4-8 Transport ship speed function2

4.3 Blue USV's best avoidance strategy

To deal with the opponent's tight defense, the blue team's best avoidance strategy needs to fully utilize the blue team's USV speed advantage. Based on the mechanism analysis of the dodge process, we have drawn inspiration from the extraordinary actions of football and other ball sports and designed a set A relatively simple evasion technique to use. Avoiding the opponent's interception as a whole is a very complicated process. In specific ways, it is manifested in ways such as breaking through the limit speed or changing directions. We need to establish a relatively universal model to describe this process. The details include the risk factor model. The establishment and movement criteria are switched with the relative distance.

4.3.1 Hazard related models

According to the general idea of blue USV evading enemy interception, when the enemy gets closer to him, the situation becomes more dangerous, and the more he needs to evade. In this question, the movement of the USV in the x direction means forward, and the movement in the y direction means avoidance. Therefore, we need to change the direction of movement of the blue USV with the distance between it and the red. When the distance is closer, it tends to move in the y direction, and when the distance is farther, it tends to move normally in the x direction. go ahead. Since the USV movement speed is constant, its x and y components are bound by the Pythagorean theorem. So according to simple reasoning, we can think that the speed in the y direction increases as the distance between it and the red USV decreases.

Due to the limitation of the turning radius, the acceleration in the y direction will be correspondingly limited, so that he can only turn with the smallest radius of curvature in any case. Therefore, the final result derived from such a model is: if the distance between the two parties is very long, the blue USV will suffer a small "repulsion" and produce a small amount of velocity component in the y direction, but it can basically be ignored, making it basically look like Go along the x direction. As the red USV approaches, the danger of the blue USV increases, and the velocity component in the y direction gradually increases until it reaches a lateral movement perpendicular to the forward direction. If it is not caught up afterwards, the distance between the two will become farther and farther, and the danger of the blue USV will decrease. As the speed in the y direction decreases, it will gradually recover the speed in the x direction and successfully break through the interception.

To achieve such an adjustment effect, we hope to obtain a speed function in the y direction. When the distance is large, the value of the function is small or tends to 0, and the maximum value is 20m/s when the distance is less than a certain value. According to common mathematical knowledge, we can use the segmented negative exponential function to describe such a process, expressed as follows:

$$\mathrm{vy} = \begin{cases} vB \times e^{-a_0 \times (Dbr - D0)} & Dbr > D0 \\ vB & Dbr \le D0 \end{cases} \tag{4-18}$$

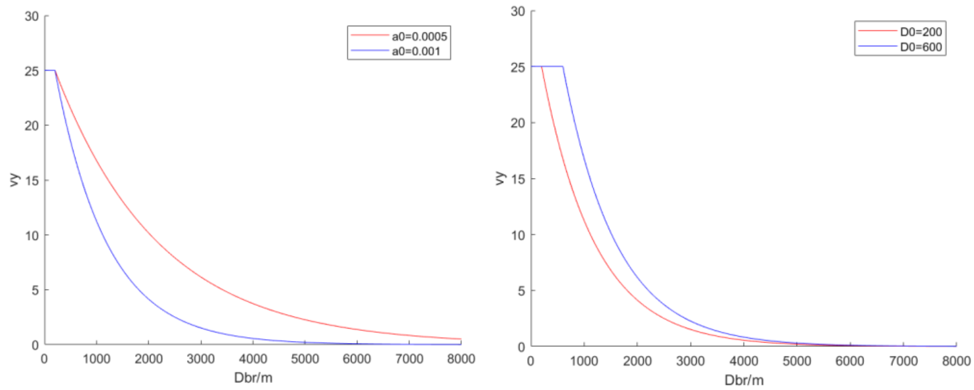


Figure 4-9 Blue USV speed function in Y direction

Analyze the influence of two parameters ***a0*** and ***D0*** on the curve: the larger ***a0***, the steeper the curve, the same ***Dbr*** at ***Dbr>D0***, the larger ***vy***, ***D0*** is the turning point of the curve, less than ***D0***, ***vy*** is always equal to ***vB***.

Of course, the y-direction movement caused by the above function must meet the requirements of the battlefield width required by the problem, so we need to make the following optimizations for the above USV movement strategy:

On the basis of the last obtained ***vy***, multiply the boundary hazard coefficient ***ke***. The relationship between ***ke*** and the distance ***De*** is: when ***De*** is greater than the threshold ***Deth***, ***ke***=1, when ***Deth*** is less than ***Deth***, as ***De*** decreases, ***ke*** decreases, so that ***vy*** decreases. Small to prevent the blue side from moving out of bounds.

$$\mathrm{ke} = \begin{cases} -\frac{3}{\mathrm{Deth}^2}\mathrm{De}(\mathrm{De}-2\mathrm{Deth})-2 & De < Deth \\ 1 & De \geq Deth \end{cases} \tag{4-19}$$

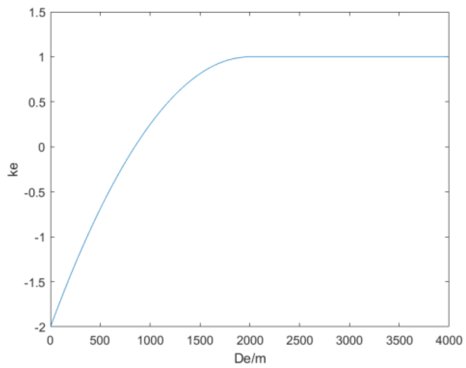


Figure 4-10 Adjustment of blue USV when facing boundary

4.3.2 The fastest crossing strategy based on simulated annealing algorithm

The fastest crossing strategy is required, and the previous model needs to be improved, so that the speed in the y direction is as small as possible and the speed in the x direction is as large as possible to pass as quickly as possible. Therefore, a blue strategy is added to the previous basis. When the blue USV bypasses the red team, the blue USV is at a certain distance from the red side cluster in the x direction after ***Dx*** (the blue USV is no longer in danger), the blue USV quickly moves the speed direction parallel to the x-axis under the condition of meeting the minimum turning radius, and rushes to the end at the fastest horizontal speed.

Then it is necessary to iteratively optimize the parameters of the function model to find the optimal parameters so that the blue square's passing time t is the shortest. There are 4 parameters to be optimized: ***D0***, ***a0***, ***Deth***, ***Dx***. The value range of the parameter is determined by experiment. The blue USV's penetration time ***t*** can only be calculated by computer simulation, and the expression of ***t*** on the parameters cannot be obtained. Therefore, the optimization algorithm based on gradient descent is not applicable. In the experiment, when the fmincon function of matlab is used to solve the problem, the value of ***t*** can hardly decrease almost every time it is calculated. It is speculated that there are many local minima in the solution space.

The simulated annealing algorithm can effectively skip the local extreme points and find the global optimal solution. The advantage of this algorithm is to accept non-optimal solutions with a certain probability. The algorithm has an important parameter: temperature ***T***. With the iteration of the algorithm, ***T*** gradually decreases, using an exponential decline. When ***T*** is larger, there is a greater probability of accepting the non-optimal solution, which has the following formula:

$$\mathrm{p} = \begin{cases} 1 & f(Sn) < f(S) \\ e^{-\frac{f(Sn)-f(S)}{T}} & f(Sn) \geq f(S) \end{cases} \tag{4-20}$$

S is the current optimal solution, ***Sn*** is a new solution generated randomly, ***P*** is the probability of accepting the new solution, ***f(Sn)*** is the objective function value obtained from the new solution, ***f(S)*** is the objective function value obtained from the current optimal solution. The algorithm will randomly select the next possible solution within a certain range, and the range will decrease as ***T*** decreases. Therefore, the optimal solution fluctuates up and down at the beginning, and the result becomes stable later.

4.4 Generative adversarial network algorithm

An optimal strategy is often to find more information from the whole process, so it also becomes very complicated and may even be difficult to interpret. In contrast, the relatively easy-to-implement interception or evasion algorithm we designed does not consider the current estimated speed of the opponent's USV, the statistics of the change trend and other information, and the processing process is relatively rough. Due to the typical confrontational nature of this topic, a most reasonable strategy can be derived from each other, so that the algorithms of both parties iteratively tend to be optimal. Whenever the blue party iterates to a more efficient evasion path algorithm, it can easily break through the outdated red party interception and force the red party to iteratively update its interception strategy, thus making it easy to intercept the blue party's infiltration. Strategies will gradually tend to be optimized. The general principle is shown in the figure below.

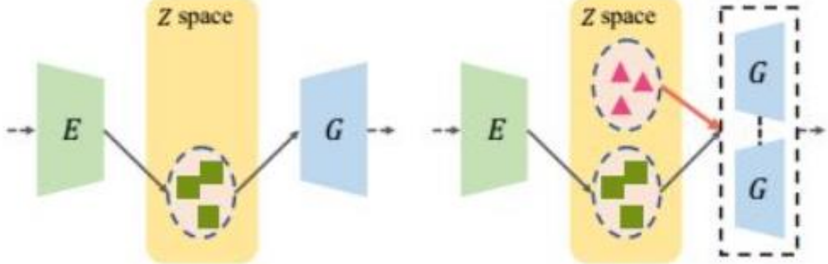


Figure4-11 Principles of GAN

Due to the limitations of conditions and capabilities, we failed to complete the training of the generative confrontation network for this problem, so we propose a solution for discussion. In fact, in the process of our research on this problem, the process of gradually optimizing and obtaining better avoidance and interception algorithms has a certain confrontational nature. Thinking back and forth about the better strategies of both sides, but the efficiency should be much worse.

5 Solution & Result of Problem 1 & 2

5.1 Simulation of interception of the red side towards the target

When we take the initial point of the blue square as the leftmost midpoint, the result is as follows. The blue dot represents the blue side, the red circle represents the red cluster, and the green circle represents the red carrier. See the left picture below. When the blue side approaches the red side, the blue side's speed in the y direction increases, bypassing the red block. The speed in the y direction gradually decreases after the distance from the red square.

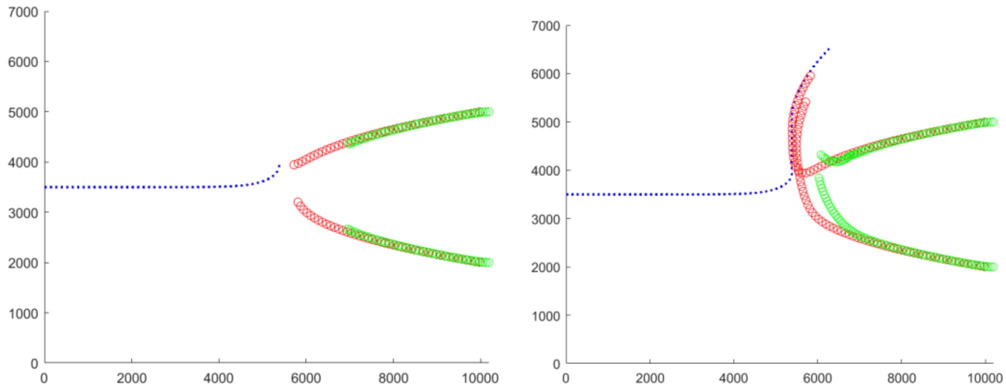


Figure5-1 Simulation of red side towards the target

The picture below shows the complete road map, and the blue team successfully penetrated the defense. When approaching the upper boundary, the speed in the y direction decreases faster.

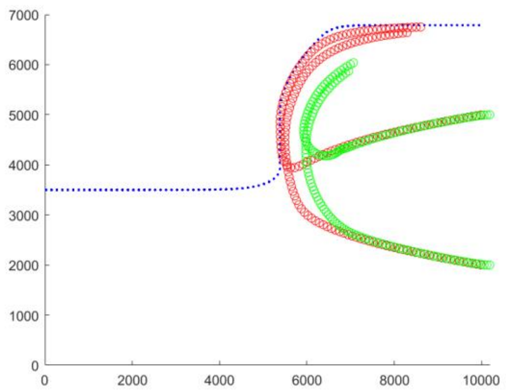


Figure5-2 Complete roadmap of red side towards the target

5.2 Finding the initial position of the blue USV penetration

5.2.1 Monte Carlo statistics solution method

Randomly cast points within the range as the initial position of the blue side. The red defense adopts an interception method that directly faces the target. Use computer simulation to get the result of successful penetration.

5.2.2 Results

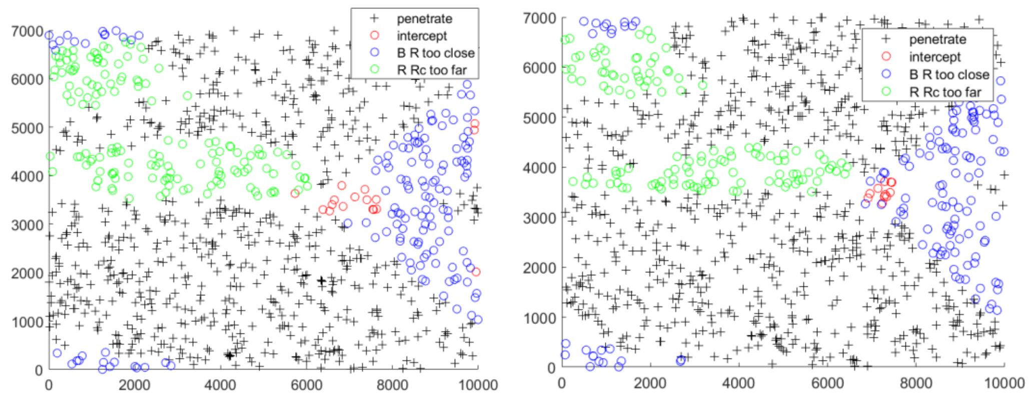


Figure5-3 Monte Carlo statistics solution of initial position

The above figure shows the results of 1000 initial points with ***D0=180(left)*** and ***D0=250(right)***. Only the points near the (7000,3500) coordinates are intercepted, because the blue side and the red side are very close, the battlefield is reduced, and the blue side's escape range is reduced. The point on the right is closer to the red side, and soon the blue side and the red side's carrier are too close, and the blue side wins.

When D0=180, more blue USV are intercepted than when D0=250. It can be seen that the larger the D0, the better the blue team's penetration ability. When D0=180, penetration rate=0.983, When D0=250, penetration rate= 0.9975. On the other hand, the distribution pattern of the overall offensive and defensive results about the initial point remains consistent when the parameters are different, which verifies the stability of the model.

5.3 Search for Mmin and Mmax

The strategy of the blue side: parameters D0=400m, Deth=500m, a0=10/M. The defensive strategy of the red side: adopts direct defense. M traverses from 100m to 15000m, with 100m as the step length, and then carries on the computer simulation in turn to get the following results:

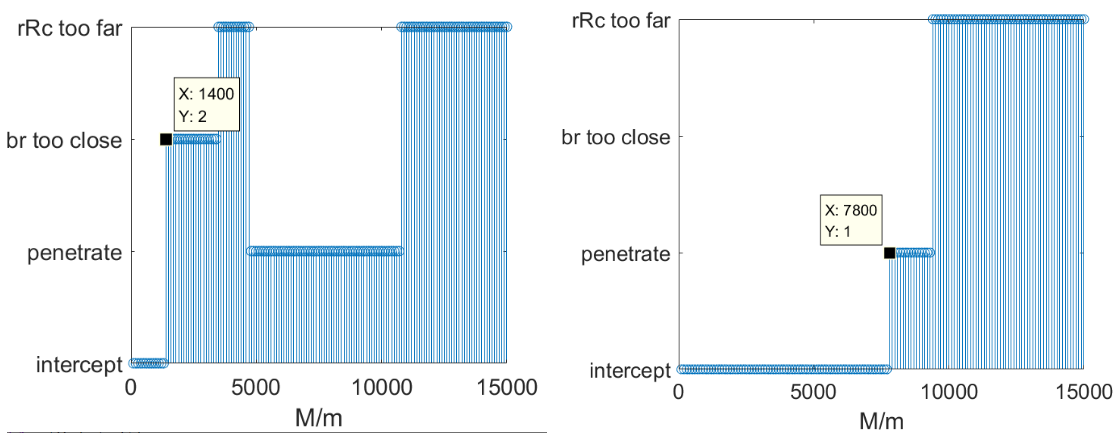


Figure5-5 Solution of Mmin and Mmax

When ***M<1400m***, the blue side will be intercepted, so ***Mmin=1400m***. When ***M>Mmin***, the Blue USV can break through the interception of the Red UAVs cluster.

The blue strategy remains unchanged, the red strategy changes to the improved y-direction alignment method, the M range remains unchanged, computer simulation is performed, and the following results are obtained:

When ***M<7800m***, the Blue USV will be intercept, so ***Mmax=7800m.*** This also verifies that the red team's improved defensive strategy is effective.

The demonstration of the defensive strategy aligned in the Y direction is shown in the figure below, which shows successful interception by the red side(left 2 figure) and a successful penetration of blue USV(right 1 figure).

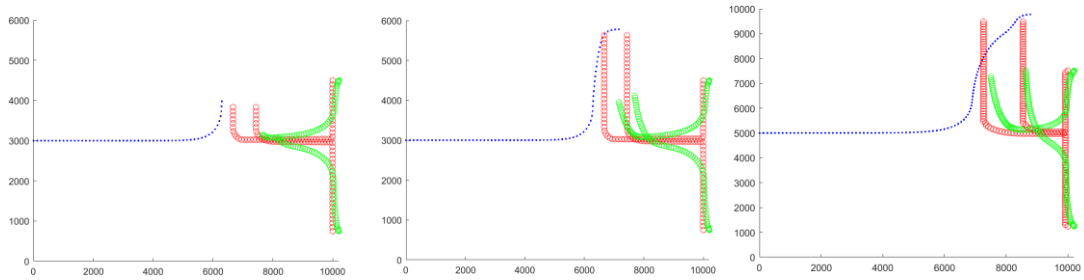


Figure5-6 Defensive strategy demonstration

5.4 Solving the Blue's fastest penetration strategy

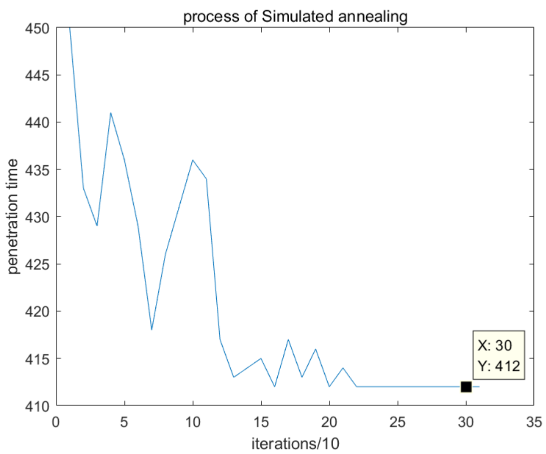


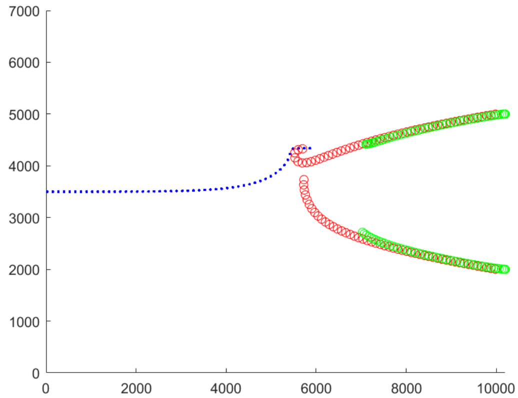
Figure5-8 SA method to get the blue square passing time

The above figure shows the iterative descent process of the blue square through time obtained by the simulated annealing method. It can be seen that the algorithm can jump out of the local minimum and gradually find the global minimum. The shortest time is 412s, and the optimal solution is:

Table5-1 Optimal solution of crossing time

D0	*Deth*	*Dx*	*a0*
186.13m	1316.8m	-377.78m	0.0008

The corresponding motion trajectory diagram is:



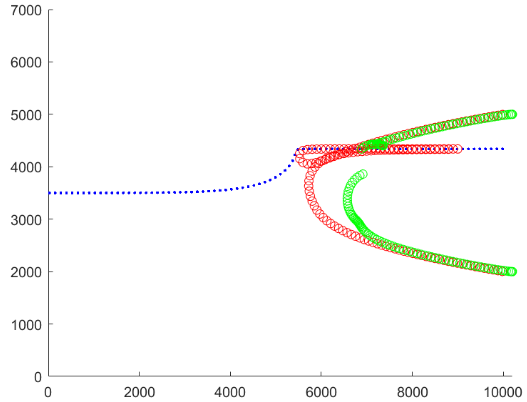


Figure5-9 Blue penetration movement trajectory

It can be seen that the blue side just took advantage of the minimum turning radius of the red side to make a sharp turn near the red cluster, rubbed the edge near the red side, and then moved parallel to the x-axis to the right.

6 Model Establishment of Problem 3

This problem is relatively complicated. The release of the two-wave USV introduces many variables. In order to gradually solve this problem, we can disassemble this big problem into two relatively independent sub-problems. Our decomposition ideas are as follows Picture:

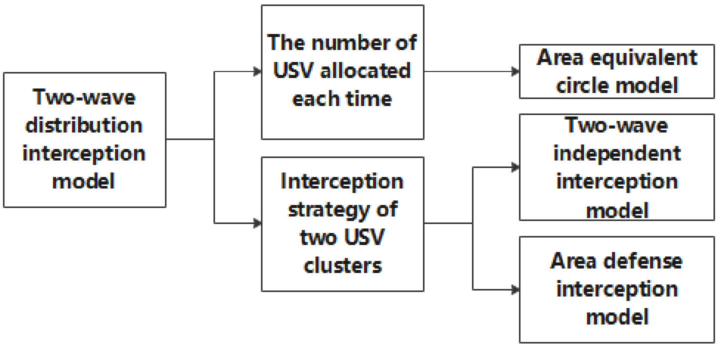


Figure6-1 Break down the thinking framework

6.1 Two-wave number distribution model

According to our model placed inside the cluster, when the number of USVs in the cluster increases, the interception range of the cluster will increase. According to the requirements of the subject, the two-wave allocation scheme for each USV mothership is limited, as shown in the following table:

Table6-1 Distribution plan

Plan	1	2	3	4	5
First dispatch	3	4	5	6	7
Second dispatch	7	6	5	4	3

6.1.1 Overview of the internal layout of the cluster

In order to choose a better way to allocate the number of USVs, it is necessary to study the interception coverage effect of clusters of different numbers of USVs. If the number of internal USVs is 3, only a regular triangle is the most efficient arrangement. If the number of internal USVs is 4, there are two conventional placements, regular rhombus and square placement. Through simple calculations, it can be known that the effective interception area of the square placement is larger. The regular hexagon is also the most efficient way to place 6 USVs. Similarly, when there are 7 USVs, the regular heptagon is the most efficient. Through geometric calculation and demonstration, it can be obtained that in order to achieve the maximum effective intercept area, only regular polygons can be selected to achieve the highest efficiency. The following is a schematic diagram of the effective interception area of these placements:

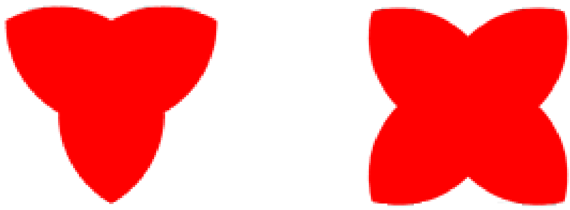


Figure6-2 Valid area for 3&4USVs

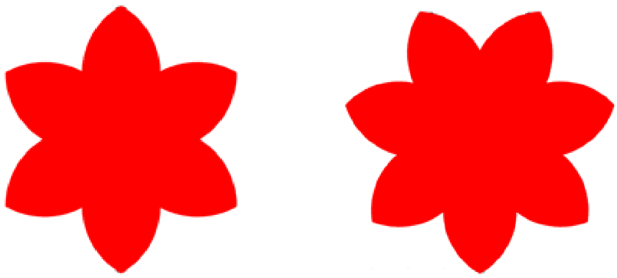


Figure6-3 Valid area for 6&7USVs

Table6-2 Valid radius of different USV numbers

Number of USV	3	4	5	6	7
Valid area	88.0	104.8	123.5	141.5	159.0

By comparing the effective radius sum of the two times, we can calculate that no matter how the allocation is made, the radius sum is always surprisingly similar: approximately equal to 247! Therefore, we cannot look at the distribution of quantity from the area alone, we need to comprehensively analyze the overall effect of the two waves. Through a large number of different demonstrations below, we found that if the red side USV successfully intercepts the blue side's infiltration, it is basically the last interception completed by the later USV, but the harassment and inducement of the previous batch of USVs. Therefore, in the subsequent release allocation strategy, if the two waves of USV have the same role, the allocation plan of 5 for both waves will be adopted. If the roles of the previous and the next two are divided, the first three USVs will be allocated. Seven USVs are allocated at the back.

6.2 Two wave release strategies

6.2.1 Two independent interception models

Using the previous research on one-time interception, we can find that although the improved interception algorithm is more efficient in most of the time, in the two-wave joint interception operation, after a large number of simulation experiments, we found that if All USV cluster algorithms are consistent, which will cause many USVs of the red side to be gathered into a group and forced to initiate interception from one direction. The interception efficiency is very poor. The simulated motion trajectory is shown in the following figure:

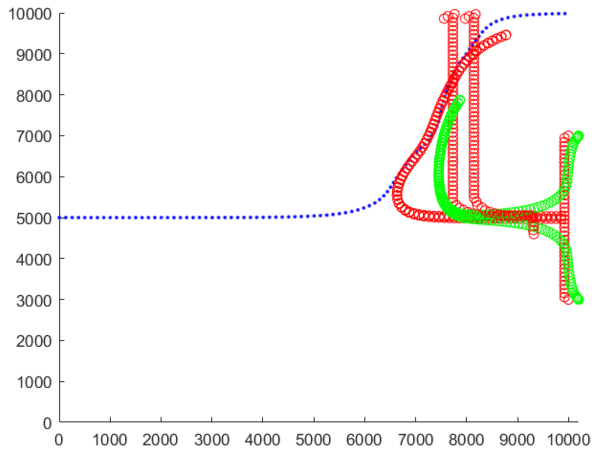


Figure6-4 Failed interception

Therefore, after studying the interception process, we hope to initiate interception from as many directions as possible, so the two intercepted USVs adopt different interception strategies. In this question, we intercepted the USV cluster that was released for the first time by direct pursuit, and always chased the blue USV. After the blue and red first wave of USV reached a certain distance, the red transport ship released the second wave of USV to intercept. This reserved distance is recorded as Dwave2. This wave of USV adopts an improved interception strategy, that is, prioritize adjustment of the vertical position. In this way, the interception process tends to be a situation where the previous wave and the next wave of USVs will pinch and attack the enemy USV, making the blue USV need to constantly avoid approaching from all directions, greatly enhancing the success rate of interception. The flow chart of the entire system is as follows:

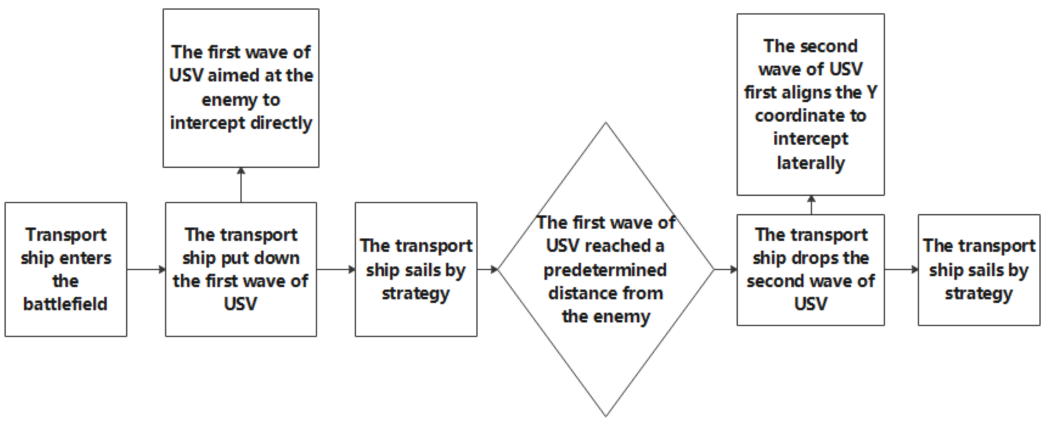


Figure6-5 Framework of independent interception process

In the design of the simulation system, we defined the variable wave2 to determine whether the transport ship has released the second wave of USV, so that the state space model we designed can continue to be used in such a situation without multiple releases. The second Issues such as the dislocation of the USV's position refresh and other issues have made the release of the second wave an important indicator of the interception phase. According to the description in the first part, here the first wave emits 3 USVs of regular triangles, and the second wave emits 7 USVs of regular heptagons.

6.2.2 Interception model of area defense

In ball games, in order to improve the efficiency of defensive interception as much as possible, we often adopt regional defense tactics [3]. In this question, we hope to use regional defense to allow the red USV to take priority in certain areas of the battlefield when the USV of the red side is not close to the USV of the blue side, so that once an interception is initiated at a certain moment, the USV occupying each area can Easily initiate interception of blue-side USVs from all directions. The USVs of each USV cluster give full play to their positional advantages, allowing their numerical advantages to be brought into play.

In the design of this question, I first spread the preset defense points evenly in line with the width of the battlefield, so that no matter what direction the blue side penetrates from, there is at least one very close red USV cluster waiting on defense. The actual situation proceeded normally according to the principle, and played a certain effect.

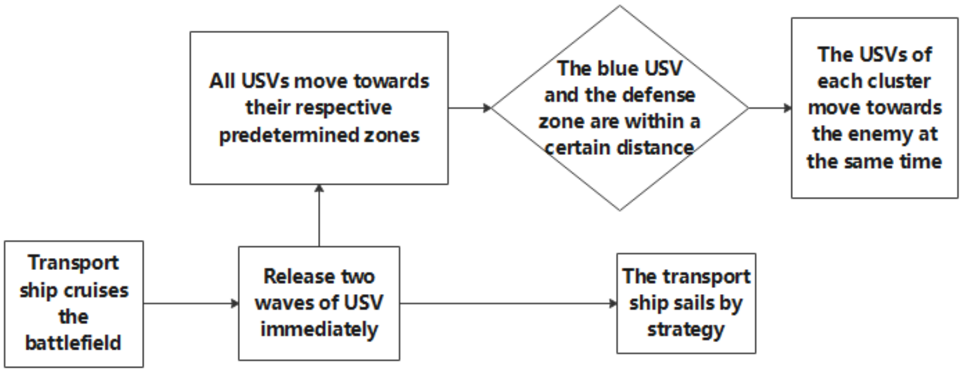


Figure6-6 Framework of area interception process

Here both waves release 5 USVs of a regular pentagon as a cluster.

7 Solution & Result of Problem 3

Because the release strategy has been described in the model establishment, it is not a fixed value, here we only need to focus on the solution of Mmax.

7.1 Independent interception

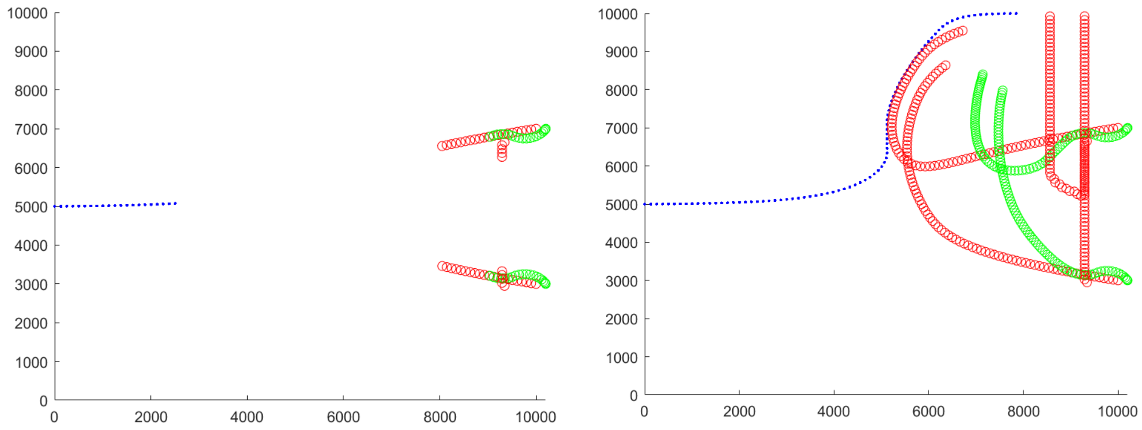


Figure 7-1 Trace of independent interception

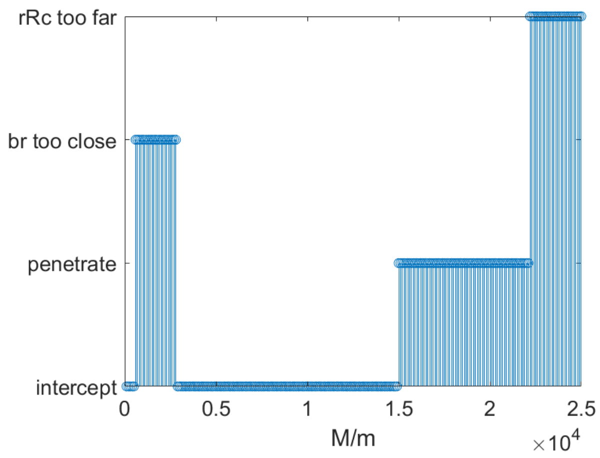


Figure 7-2 Finding of Mmax

When M<15000m, the Blue USV will be intercept， so **Mmax=15000m.**

7.2 Area interception

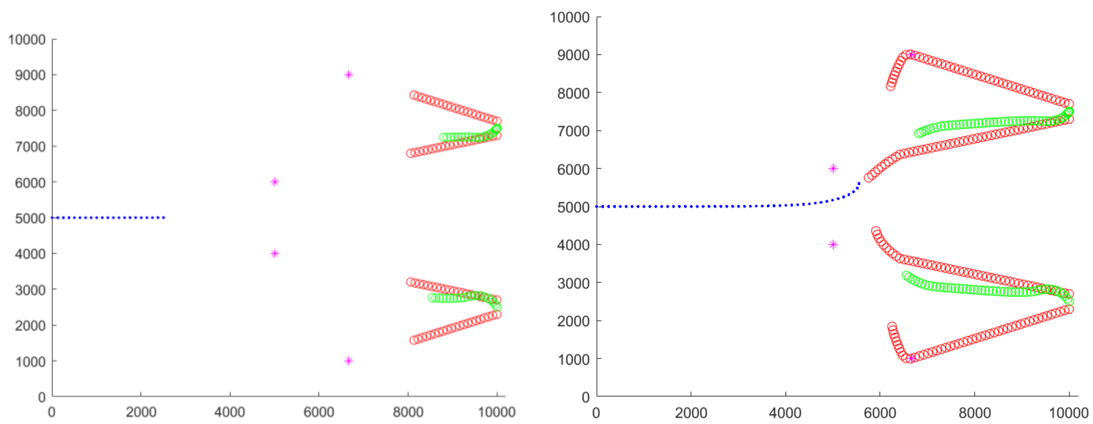


Figure 7-3 Trace of area interception

The figure above shows an example of successful interception

8 Strengths and Weaknesses

8.1 Strengths

Our model reflects the idea of decomposing the problem. Due to the different dimensions of the problem, it is possible to solve it separately, and such a process makes our thinking clear. In addition, we have also used a variety of modern algorithms to obtain the value of the solution through statistical techniques, making the results more reliable. In terms of mechanism analysis, we try to get inspiration from similar sports models, and get a solution suitable for this problem by optimizing mature sports strategies. The most important thing is that we used MATLAB simulation to fit the entire process, so that the variables involved in the interception and penetration confrontation can be set freely, which greatly facilitates our debugging process. During the simulation in the running process, we use graphics of different colors and shapes to make the visualization process easy to understand and clear at a glance.

8.2 Weakness

Due to the lack of time and related technologies, we failed to complete the construction and training of GAN, so we could not find the closest and most reasonable two-party strategy. In addition, our mechanism analysis process is not absolutely reasonable, it is a compromise between the feasibility of our simulation and closer to the actual situation. The selection of functions in our decision-making model may be too rough, and may not fully reflect the process of the two parties in the process of operation.

9 References and notes

[1] Ouhsaine Lahoucine, Ramenah Harry, El Ganaoui Mohammed, et al. Dynamic state-space model and performance analysis for solar active walls embedded phase change material. 2020, 24

[2] Gao Fugang, Zhang Gao. Effectiveness analysis of maritime blockade and interception operations[J]. Military Operations Research and Systems Engineering, 2014, 28(01): 30-32+60.

[3] Li Min, Li Bin, Huang Hao, Liu Lu, Li Ruijiang, Sun Chunxing. Based on 5VS5 simulation robot soccer area defense strategy design [J]. Digital Technology and Application, 2011 (10): 172-173.

Appendix

1. Calculate effective area

S1.m

```
function S1
p1=[50 0];
r72=pi*72/180;%弧度
r36=r72/2;
p2=[50+100*cos(r72),100*sin(r72)];
p3=[0 100*(sin(r36)+sin(r72))];
p4=[-p2(1) p2(2)];
p5=-p1;
S1=cal_S([p1;p2;p3;p4;p5],-160,160,-100,250)
r1=sqrt(S1/pi)
```

cal_S.m

```
function S=cal_S(p,xl,xu,yl,yu)
%p:n*2 存 n 个圆心坐标 x y 投点范围[xl,xu]
close all
r=100;
[n,~]=size(p);
f=cell(n,1);
for i=1:n
f{i}=@(x,y)(x-p(i,1)).^2+(y-p(i,2)).^2-r^2;
end
N=1e5;%投点个数
x=xl+(xu-xl)*rand(1,N);
y=yl+(yu-yl)*rand(1,N);
M=zeros(1,N);
for i=1:n
    M=M+[f{i}(x,y)<=0];%方程小于等于 0 代表在圆内 0 1 向量
end
id=[M>=2];%至少在两个圆内
m=sum(id);
plot(x(id(1:1e5)),y(id(1:1e5)),'r.')%展示 1e5 个点
axis equal
S=(xu-xl)*(yu-yl)*m/N;
```

2. Simulation process

angle.m

```
function a=angle(d,s)
%a: the direction of self to destination(rad)
if d(1)-s(1)<0
    a=atan((d(2)-s(2))/(d(1)-s(1)))+pi;
```

```
elseif d(1)-s(1)>=0%第 14 象限角，朝 x 正方向运动
        a=atan((d(2)-s(2))/(d(1)-s(1)));
        if a<0
                a=a+2*pi;%arctan 输出负的转为正的
        end
end

cal_direc.m
function dir_n=cal_direc(dir,p_dst,p_self,ddirmax)
%运动方向直接指向目标
%输入当前方向角，目的坐标，自己坐标，最大转向角度，输出下一方向角
% dis=dis_p2line(pR,tan(dR),pB);
% dd=asin(dis/norm(pR-pB));%方向角变化量
if p_dst(1)-p_self(1)<0%目的坐标小于自己的 x 坐标，朝 x 负方向运动，第 23 象限角
        dir_n=atan((p_dst(2)-p_self(2))/(p_dst(1)-p_self(1)))+pi;
elseif p_dst(1)-p_self(1)>0%第 14 象限角，朝 x 正方向运动
        dir_n=atan((p_dst(2)-p_self(2))/(p_dst(1)-p_self(1)));
        if dir_n<0
                dir_n=dir_n+2*pi;%arctan 输出负的转为正的
        end
elseif p_dst(1)-p_self(1)==0
        if p_dst(2)-p_self(2)<0 %y 坐标减小
                dir_n=3*pi/2;
        else
                dir_n=pi/2;
        end
end
if abs(dir_n-dir)>ddirmax %最大转向角度限制
        if dir_n>dir
                dir_n=dir+ddirmax;
        else
                dir_n=dir-ddirmax;
        end
end

q1.m
function r=q1()
close all
global ddBmax dt vB M vRc0 vRcmax
L=1e4;
M=7000;
pR1n=[L,5*M/7];%红中心点 1 初始位置坐标[x,y]
pR2n=[L,2*M/7];
pRc1n=[L+200,pR1n(2)];%红运输舰初始位置  在红集群之后 200m
```

```
pRc2n=[L+200,pR2n(2)];%以后 Rc 代表红运输舰，R 代表红集群中心
pBn=[0,M*1/2];
rB=100;%蓝最小转弯半径
rR=80;
rRc=500;
vB=25;%蓝速度 m/s
vR=20;%红中心点速度 m/s
vRc0=13;%红运输舰初始速度
vRcmax=16;%红运输舰最大速度
ddBmax=dt*vB/rB;%蓝方向变化最大角度
ddRmax=dt*vR/rR;
dt=1;%模拟间隔时间/s
dBn=0;%蓝初始方向角
dR1n=atan((pBn(2)-pR1n(2))/(pBn(1)-pR1n(1)))+pi;%红 1 初始化方向，指向蓝方
dR2n=atan((pBn(2)-pR2n(2))/(pBn(1)-pR2n(1)))+pi;%第 23 象限角，朝 x 负方向运动
dRc1n=pi; dRc2n=pi;
Dth=123;
t=0;
hold on
axis([0 L+200 0 M])%坐标轴范围
vRc1=vRc0; vRc2=vRc0;
while pBn(1)<L & t<700 & min(norm(pBn-pR1n),norm(pBn-pR2n))>Dth
    pB=pBn;%更新位置
    pR1=pR1n; pR2=pR2n; pRc1=pRc1n; pRc2=pRc2n;
    dB=dBn;%更新方向
    dR1=dR1n; dR2=dR2n; dRc1=dRc1n; dRc2=dRc2n;

    dBn=cal_direcB(dB,pB,pR1,pR2);%下一蓝速度方向角
    pBn=cal_position(pB,vB,dB);%下一蓝位置坐标

    %由蓝的位置坐标计算红的速度方向
    dR1n=cal_direc(dR1,pB,pR1,ddRmax);%下一红 1 速度方向角
    %输入上一个红的方向角，蓝的位置坐标为目标，红的位置坐标，输出下一个红的方向角
    dR2n=cal_direc(dR2,pB,pR2,ddRmax);

    ddRcmax1=dt*vRc1/rRc;
    ddRcmax2=dt*vRc2/rRc;
    dRc1n=cal_direcRc(dRc1,pB,pR1,pRc1,ddRcmax1);%下一红运输舰 1 速度方向角
    %红集群的位置坐标为目标
    dRc2n=cal_direcRc(dRc2,pB,pR2,pRc2,ddRcmax2);

    pR1n=cal_position(pR1,vR,dR1);%由速度方向确定下一红 1 位置坐标
    pR2n=cal_position(pR2,vR,dR2);
    vRc1=get_vRc(pR1,pB,pRc1);%获得运输舰速度大小  由相对位置决定
```

```
    vRc2=get_vRc(pR2,pB,pRc2);

    pRc1n=cal_position(pRc1,vRc1,dRc1);%下一红运载舰 1 位置坐标
    pRc2n=cal_position(pRc2,vRc2,dRc2);
    if min(norm(pRc1n-pB),norm(pRc2n-pB))<1000
        disp('蓝与红运载舰距离过近')
        r=2;
        return
    end
    if max(norm(pRc1n-pR1n),norm(pRc2n-pR2n))>2000+Dth
        disp('红运载舰与集群距离过远')
        r=2;
        return
    end
    if mod(t,5)<0.1 %5s 更新一次图
        scatter(pBn(1),pBn(2),'b.')
        scatter(pR1n(1),pR1n(2),'r')
        scatter(pR2n(1),pR2n(2),'r')
        scatter(pRc1n(1),pRc1n(2),'g')
        scatter(pRc2n(1),pRc2n(2),'g')
    end
    pause(0.0002)
    t=t+dt;
end
if min(norm(pBn-pR1n),norm(pBn-pR2n))<=Dth
    disp('successful interception')
    r=0;
elseif t>=700
    disp('time out')
    r=0;
elseif pBn(1)>=L
    disp('successful penetration')
    r=1;
end

function dBn=cal_direcB(dB,pB,pRc1,pRc2)
%输入上一个蓝的方向角，蓝的位置坐标，红的位置坐标，输出下一个蓝的方向角
global ddBmax vB M
Dbr=min(norm(pRc1-pB),norm(pRc2-pB));
D0=400;%临界距离  该距离下蓝方竖直速度最大
Deth=500;%蓝方与边界距离阈值，小于后竖直速度减小
a0=10/M;%0.0001-0.001
if pB(2)<M/2 %下半平面蓝向下运动
    k=-1;
```

```
else
    k=1;
end
if Dbr>D0
    vy=k*vB*exp(-a0*(Dbr-D0));
else
    vy=k*vB;
end
if k==1%向上运动
    De=M-pB(2);%到上边界的距离
else%向下
    De=pB(2);
end
if De<Deth
    vy=vy*(-3/Deth^2*De*(De-2*Deth)-2);
end
vx=sqrt(vB^2-vy^2);
dBn=atan(vy/vx);
if abs(dBn-dB)>ddBmax
    %disp(1)
    if dBn>dB
        dBn=dB+ddBmax;
    else
        dBn=dB-ddBmax;
    end
end

function v=get_vRc(pR,pB,pRc)
global vRc0 vRcmax
Dr=norm(pRc-pR);
v=(vRcmax-vRc0)./(1+exp(-Dr/100+5))+vRc0;
Db=norm(pRc-pB);
Dbth=3000;
if Db<Dbth
    k=-1/(Dbth-1000)^2*(Db-1000)*(Db+1000-2*Dbth);
else
    k=1;
end
v=v*k;

function pn=cal_position(p,v,d)
%由当前位置、速度大小、速度方向角计算下一个位置坐标 pn
%微元法  每小段近似为直线运动
global dt
```

```
vx=v*cos(d);
vy=v*sin(d);
xn=p(1)+vx*dt;
yn=p(2)+vy*dt;
pn=[xn yn];

q2_M.m
function r=q2_M(M)
%close all
global ddBmax dt vB vRc0 vRcmax
L=1e4;
pR1n=[L,5*M/7];%红中心点 1 初始位置坐标[x,y]
pR2n=[L,2*M/7];
pRc1n=[L+200,pR1n(2)];%红运输舰初始位置  在红集群之后 200m
pRc2n=[L+200,pR2n(2)];%以后 Rc 代表红运输舰，R 代表红集群中心
pBn=[0,M*1/2];
rB=100;%蓝最小转弯半径
rR=80;
rRc=500;
vB=25;%蓝速度 m/s
vR=20;%红中心点速度 m/s
vRc0=13;%红运输舰初始速度
vRcmax=16;%红运输舰最大速度
ddBmax=dt*vB/rB;%蓝方向变化最大角度
ddRmax=dt*vR/rR;
dt=0.1;%模拟间隔时间/s
dBn=0;%蓝初始方向角
dR1n=atan((pBn(2)-pR1n(2))/(pBn(1)-pR1n(1)))+pi;%红 1 初始化方向，指向蓝方
dR2n=atan((pBn(2)-pR2n(2))/(pBn(1)-pR2n(1)))+pi;%第 23 象限角，朝 x 负方向运动
dRc1n=pi; dRc2n=pi;
Dth=123;
t=0;
% hold on
% axis([0 L+200 0 M])%坐标轴范围
vRc1=vRc0; vRc2=vRc0;
while pBn(1)<L & t<700 & min(norm(pBn-pR1n),norm(pBn-pR2n))>Dth
    pB=pBn;%更新位置
    pR1=pR1n; pR2=pR2n; pRc1=pRc1n; pRc2=pRc2n;
    dB=dBn;%更新方向
    dR1=dR1n; dR2=dR2n; dRc1=dRc1n; dRc2=dRc2n;

    dBn=cal_direcB(dB,pB,pR1,pR2,M);%下一蓝速度方向角
    pBn=cal_position(pB,vB,dB);%下一蓝位置坐标
```

```
    %由蓝的位置坐标计算红的速度方向
    dR1n=cal_direc(dR1,pB,pR1,ddRmax);%下一红 1 速度方向角
    %输入上一个红的方向角，蓝的位置坐标为目标，红的位置坐标，输出下一个红的方向角
    dR2n=cal_direc(dR2,pB,pR2,ddRmax);

    ddRcmax1=dt*vRc1/rRc;
    ddRcmax2=dt*vRc2/rRc;
    dRc1n=cal_direcRc(dRc1,pB,pR1,pRc1,ddRcmax1);%下一红运输舰 1 速度方向角
    %红集群的位置坐标为目标
    dRc2n=cal_direcRc(dRc2,pB,pR2,pRc2,ddRcmax2);

    pR1n=cal_position(pR1,vR,dR1);%由速度方向确定下一红 1 位置坐标
    pR2n=cal_position(pR2,vR,dR2);
    vRc1=get_vRc(pR1,pB,pRc1);%获得运输舰速度大小  由相对位置决定
    vRc2=get_vRc(pR2,pB,pRc2);

    pRc1n=cal_position(pRc1,vRc1,dRc1);%下一红运载舰 1 位置坐标
    pRc2n=cal_position(pRc2,vRc2,dRc2);
    if min(norm(pRc1n-pB),norm(pRc2n-pB))<1000
        %disp('蓝与红运载舰距离过近')
        r=2;
        return
    end
    if max(norm(pRc1n-pR1n),norm(pRc2n-pR2n))>2000+Dth
        %disp('红运载舰与集群距离过远')
        r=3;
        return
    end
%     if mod(t,5)<0.1 %5s 更新一次图
%         scatter(pBn(1),pBn(2),'b.')
%         scatter(pR1n(1),pR1n(2),'r')
%         scatter(pR2n(1),pR2n(2),'r')
%         scatter(pRc1n(1),pRc1n(2),'g')
%         scatter(pRc2n(1),pRc2n(2),'g')
%     end
%     pause(0.0002)
    t=t+dt;
end
if min(norm(pBn-pR1n),norm(pBn-pR2n))<=Dth
    %disp('successful interception')
    r=0;
elseif t>=700
    disp('time out')
    r=0;
```

```
else
    %disp('successful penetration')
    r=1;
end

function dBn=cal_direcB(dB,pB,pRc1,pRc2,M)
%输入上一个蓝的方向角，蓝的位置坐标，红的位置坐标，输出下一个蓝的方向角
global ddBmax vB
Dbr=min(norm(pRc1-pB),norm(pRc2-pB));
D0=400;%临界距离  该距离下蓝方竖直速度最大
Deth=500;%蓝方与边界距离阈值，小于后竖直速度减小
a0=10/M;%0.0001-0.001
if pB(2)<M/2 %下半平面蓝向下运动
    k=-1;
else
    k=1;
end
if Dbr>D0
    vy=k*vB*exp(-a0*(Dbr-D0));
else
    vy=k*vB;
end
if k==1%向上运动
    De=M-pB(2);%到上边界的距离
else%向下
    De=pB(2);
end
if De<Deth
    vy=vy*(-3/Deth^2*De*(De-2*Deth)-2);
end
vx=sqrt(vB^2-vy^2);
dBn=atan(vy/vx);
if abs(dBn-dB)>ddBmax
    %disp(1)
    if dBn>dB
        dBn=dB+ddBmax;
    else
        dBn=dB-ddBmax;
    end
end

function v=get_vRc(pR,pB,pRc)
global vRc0 vRcmax
Dr=norm(pRc-pR);
```

```
v=(vRcmax-vRc0)./(1+exp(-Dr/100+5))+vRc0;
Db=norm(pRc-pB);
Dbth=3000;
if Db<Dbth
    k=-1/(Dbth-1000)^2*(Db-1000)*(Db+1000-2*Dbth);
else
    k=1;
end
v=v*k;

function pn=cal_position(p,v,d)
%由当前位置、速度大小、速度方向角计算下一个位置坐标 pn
%微元法  每小段近似为直线运动
global dt
vx=v*cos(d);
vy=v*sin(d);
xn=p(1)+vx*dt;
yn=p(2)+vy*dt;
pn=[xn yn];

q3_1.m
function r=q3_1(M)%区域防守
global ddBmax dt vB L vRc0 vRcmax
L=1e4;
%M=10000;
% close all
% hold on
% axis([0 L+200 0 M])%坐标轴范围
%初始位置，红方一次性释放
pRc1n = [L,7.5*M/10];
pRc2n = [L,2.5*M/10];
pR1n = [L,7.5*M/10 + 200];
pR2n = [L,7.5*M/10 - 200];%12 对应运输舰 1
pR3n = [L,2.5*M/10 + 200];
pR4n = [L,2.5*M/10 - 200];%34 对应运输舰 2
pBn = [0,M*1/2];
pD1n = [L/4 4*M/5];      %初始防守阵地坐标
pD2n = [L/4 3*M/5];
pD3n = [L/4 2*M/5];
pD4n = [L/4 1*M/5];
pD10 = [L*2/3 9*M/10];      %初始防守阵地坐标
pD20 = [L/2 3*M/5];
pD30 = [L/2 2*M/5];
pD40 = [L*2/3 1*M/10];
```

```
rB=100;%蓝最小转弯半径
rR=80;
rRc=500;
vB=25;%蓝速度 m/s
vR=20;%红中心点速度 m/s
vRc0=0;%红运输舰初始速度
vRcmax=16;%红运输舰最大速度
ddBmax=dt*vB/rB;%蓝方向变化最大角度
ddRmax=dt*vR/rR;
dt=1;%模拟间隔时间

%初始方向角
dBn=0;
% dR1n=atan((pBn(2)-pR1n(2))/(pBn(1)-pR1n(1)))+pi;%红 1 初始化方向，指向蓝方
% dR2n=atan((pBn(2)-pR2n(2))/(pBn(1)-pR2n(1)))+pi;%第 23 象限角，朝 x 负方向运动
dR1n=pi; dR2n=pi;dR3n=pi; dR4n=pi;%所有红方 USV 初始向左

dRc1n=cal_dRc_init(pRc1n,pBn);
dRc2n=cal_dRc_init(pRc2n,pBn);%红方运输舰初始方向

Dth = 123.5;%拦截半径
D_attack = 1000;   %蓝方距离其中一个防区距离很近时集体发起攻击

t=0;
vRc1=vRc0; vRc2=vRc0;
attack = 0;%初始化状态，0 为未发起总攻，1 为发起总攻后
while pBn(1)<L & t<700 & ...
    min([norm(pBn-pR1n),norm(pBn-pR2n),norm(pBn-pR3n),norm(pBn-pR4n)])>Dth

    %更新位置
    pB=pBn;
    pR1=pR1n; pR2=pR2n;   pR3=pR3n; pR4=pR4n;
    pRc1=pRc1n; pRc2=pRc2n;
%     pD1=pD1n;pD2=pD2n;pD3=pD3n;pD4=pD4n;

%     if min([norm(pBn-pD1n),norm(pBn-pD2n),norm(pBn-pD3n),norm(pBn-pD4n)])<D_attack
    if min([norm(pBn-pD10),norm(pBn-pD20),norm(pBn-pD30),norm(pBn-pD40)])<D_attack
        attack = 1;
    end
    %更新方向
    dB=dBn;
    dR1=dR1n; dR2=dR2n; dR3=dR3n; dR4=dR4n;
    dRc1=dRc1n; dRc2=dRc2n;
```

```
    dBn=cal_direcB(dB,pB,pR1,pR2,pR3,pR4,M);%下一蓝速度方向角
    pBn=cal_position(pB,vB,dB);%下一蓝位置坐标

    %区域防守算法计算红的速度方向
    %输入上一个红的方向角，目标为防守阵地，输出下一个红的方向角
    if attack == 0          %2 种阶段红方 USV 的目标
        dR1n=cal_direc(dR1,pD10,pR1,ddRmax);    %下一红 1 速度方向角
        dR2n=cal_direc(dR2,pD20,pR2,ddRmax);    %下一红 2 速度方向角
        dR3n=cal_direc(dR3,pD30,pR3,ddRmax);    %下一红 3 速度方向角
        dR4n=cal_direc(dR4,pD40,pR4,ddRmax);    %下一红 4 速度方向角
    else
        dR1n=cal_direc(dR1,pB,pR1,ddRmax);    %下一红 1 速度方向角
        dR2n=cal_direc(dR2,pB,pR2,ddRmax);    %下一红 2 速度方向角
        dR3n=cal_direc(dR3,pB,pR3,ddRmax);    %下一红 3 速度方向角
        dR4n=cal_direc(dR4,pB,pR4,ddRmax);    %下一红 4 速度方向角
    end

    pR1n=cal_position(pR1,vR,dR1);%由速度方向确定下一红 1 位置坐标
    pR2n=cal_position(pR2,vR,dR2);%由速度方向确定下一红 2 位置坐标
    pR3n=cal_position(pR3,vR,dR3);%由速度方向确定下一红 3 位置坐标
    pR4n=cal_position(pR4,vR,dR4);%由速度方向确定下一红 4 位置坐标
%     pD1n = cal_pD(pD1,pB);               %确定下一时刻目标防区位置坐标
%     pD2n = cal_pD(pD2,pB);
%     pD3n = cal_pD(pD3,pB);
%     pD4n = cal_pD(pD4,pB);

    ddRcmax1=dt*vRc1/rRc;
    ddRcmax2=dt*vRc2/rRc;
    dRc1n=cal_direc(dRc1,(pR1+3*pR2)/4,pRc1,ddRcmax1);%下一红运输舰 1 速度方向角
    dRc2n=cal_direc(dRc2,(pR4+3*pR3)/4,pRc2,ddRcmax2);%下一红运输舰 2 速度方向角


    vRc1=get_vRc(pR1,pB,pRc1);%获得运输舰速度大小  由相对位置决定
    vRc2=get_vRc(pR2,pB,pRc2);              %第二波近，暂不考虑

    pRc1n=cal_position(pRc1,vRc1,dRc1);%下一红运载舰 1 位置坐标
    pRc2n=cal_position(pRc2,vRc2,dRc2);%下一红运载舰 2 位置坐标

    if min(norm(pRc1n-pB),norm(pRc2n-pB))<1000
        %disp('蓝与红运载舰距离过近')
        r=2;
        return
    end
```

```
    if          max([norm(pRc1n-pR1n),norm(pRc1n-pR2n),norm(pRc2n-pR3n),norm(pRc2n-
pR4n)])>2000+Dth
        %disp('红运载舰与集群距离过远')
        r=3;
        return
    end
%     if mod(t,5)<0.1 %5s 更新一次图
%         scatter(pBn(1),pBn(2),'b.')
%         scatter(pR1n(1),pR1n(2),'r')
%         scatter(pR2n(1),pR2n(2),'r')
%         scatter(pR3n(1),pR3n(2),'r')
%         scatter(pR4n(1),pR4n(2),'r')
%         scatter(pRc1n(1),pRc1n(2),'g')
%         scatter(pRc2n(1),pRc2n(2),'g')
%         plot(pD10(1),pD10(2),'m*')
%         plot(pD20(1),pD20(2),'m*')
%         plot(pD30(1),pD30(2),'m*')
%         plot(pD40(1),pD40(2),'m*')
%     end
%     pause(0.002)
    t=t+dt;
end
if min([norm(pBn-pR1n),norm(pBn-pR2n),norm(pBn-pR3n),norm(pBn-pR4n)])<=Dth
    %disp('successful interception')
    r=0;
elseif t>=700
    %disp('time out')
    r=0;
elseif pBn(1)>=L
    %disp('successful penetration')
    r=1;
end
end

function dBn=cal_direcB(dB,pB,pR1,pR2,pR3,pR4,M)
%输入上一个蓝的方向角，蓝的位置坐标，红的位置坐标，输出下一个蓝的方向角
global ddBmax vB
Dbr=min([norm(pR1-pB),norm(pR2-pB),norm(pR3-pB),norm(pR4-pB)]);
D0=300;%临界距离  该距离下蓝方竖直速度最大
Deth=500;%蓝方与边界距离阈值，小于后竖直速度减小
a0=0.001;
if pB(2)<M/2 %下半平面蓝向下运动
    k=-1;
else
```

```
    k=1;
end
if Dbr>D0
    vy=k*vB*exp(-a0*(Dbr-D0));
else
    vy=k*vB;
end
if k==1%向上运动
    De=M-pB(2);%到上边界的距离
else%向下
    De=pB(2);
end
if De<Deth
    vy=vy*(-1/Deth^2*De*(De-2*Deth));
end
vx=sqrt(vB^2-vy^2);
dBn=atan(vy/vx);
if abs(dBn-dB)>ddBmax
    %disp(1)
    if dBn>dB
        dBn=dB+ddBmax;
    else
        dBn=dB-ddBmax;
    end
end
end
function v=get_vRc(pR,pB,pRc)
global vRc0 vRcmax
Dr=norm(pRc-pR);
v=(vRcmax-vRc0)./(1+exp(-Dr/100+5))+vRc0;
Db=norm(pRc-pB);
Dbth=3000;
%和 B 距离判定
if Db<Dbth
    k=-1/(Dbth-1000)^2*(Db-1000)*(Db+1000-2*Dbth);
else
    k=1;
end
v=v*k;
end
function pn=cal_position(p,v,d)
%由当前位置、速度大小、速度方向角计算下一个位置坐标 pn
%微元法  每小段近似为直线运动
global dt
```

```
vx=v*cos(d);
vy=v*sin(d);
xn=p(1)+vx*dt;
yn=p(2)+vy*dt;
pn=[xn yn];
end
function pDn = cal_pD(pD,pB)
global L
%由当前阵地位置和蓝 USV 位置变化出下一个阵地位置，移动量是距离的函数
d_DB = norm(pD-pB);
bP = L*1/4;             %防区位置平移函数为直线，下面表示截距和斜率
kP = -bP/7500;
Dx = kP*d_DB+bP;
pDn = [Dx+pD(1),pD(2)];
end
```

3. Solve the optimal solution

```
find_fastest_SA,m
function [S,Min,MI]=find_fastest_SA()
global n T0 Range%全局变量
T0=50; %初始化温度值
T_min=0.1; %设置温度下界
r=0.98; %温度的下降率
n=4;%解的维数
Range=[130,400;
    500,3000;
    -400,100;
    0.0001,0.002];%各维定义域
k=20; %同一温度下循环次数
S=zeros(1,n);
Min=0;
for i=1:n
S(i)=(Range(i,2)-Range(i,1))*rand()+Range(i,1); %随机得到初始解 range(i,1)~range(i,2)
end
c=0;
cp=0;
MI=[];%记录最小值下降
T=T0;
hold on
while(T>T_min)
    for i=1:k
    f=q2_t(S);
    S_new=getS(S,T); %根据当前解和当前温度产生新解
    f_new=q2_t(S_new);%新的函数值
```

```
        delta=f_new-f;
        if (delta<0) %新解更优
            S=S_new;
            Min=f_new;
        elseif exp(-delta/T)>rand %新解较差时以一定概率接受新解
            S=S_new;
            Min=f_new;
        end
        end
T=T*r;  %降温
if mod(c,10)<0.1
        cp=cp+1;
        fprintf('%f:%f\n',T,Min)
        MI=[MI Min];
        plot(cp,Min,'o');
end
c=c+1;%用于记录循环次数，显示程序进度
end
%fprintf('f(%.4f, %.4f)=%.4f\n',S(1),S(2),fun3(S))
% fprintf('fun3 最优解为:x=%f, y=%f\n',S(1),S(2))
% fprintf('最小值为：%f\n',fun3(S))
% disp('fun1 最优解为:')
% disp(S)
% fprintf('最小值为：%f\n',fun1(S))


function S_new=getS(S,T) %产生新解
global n T0 Range
S_new=zeros(1,n);
for i=1:n
        u=Range(i,2);
        l=Range(i,1);
        S_new(i)=S(i)+(u-l)/2*T/T0*(2*rand()-1);
        %在原解基础上加上范围随 T 变化的随机量，T 越小，新解越集中
        if S_new(i)>u
            S_new(i)=u;
        elseif S_new(i)<l
            S_new(i)=l;
        end
end
```