Problem Chosen	2022	Team Control Number
B	ShuWei Cup Summary Sheet	2022110613530

Summary

In modern warfare, both the attacking and defending sides need to introduce **effective warfare strategies** to achieve a relatively stable and balanced war posture, so the **rational layout of each combat position**, **the deployment of munitions** and **the distribution of munitions weapons** by both combatants is particularly important. In this paper, we analyze and study these issues based on the attack difficulty, marching distance, weapon range and air defense deployment of each node on the red and blue sides, respectively.

For problem one, the assigned positions and number sizes of infantry, tanks, self-propelled artillery and air defense artillery for both red and blue sides, as well as the optimal command positions and several alternative positions for both red and blue sides, need to be calculated. For this problem, **attack difficulty constraint**, march distance constraint, weapon range and air defense deployment constraint are set, and firstly, a **dynamic joint model** is established to obtain the heat map of the distribution of assigned positions and quantity scale of infantry, tanks, self-propelled artillery and air defense artillery of both Red and Blue sides; then a position command model with the maximum attack difficulty and minimum march distance as the **optimization objective** is established, and an immune algorithm for solving the problem is designed that obtained the coordinate distribution positions of the alternative positions of the optimal command positions.

For problem two, it is necessary to reasonably allocate medical supplies, military supplies and daily supplies to both red and blue sides, and then give the optimal supply plan for red and blue sides based on the total number of labor and total number of vehicles in non-supply mode considering the existence of potential attack strategies of the other side. For this problem firstly, medical supplies, military supplies and daily supplies are grouped into the category of munitions, and the optimal command positions and alternative positions derived from the joint problem one are **analyzed by comprehensive clustering**, and then the met acellular automata model is used to calculate the quantity of supplies to be placed and the total number of workers and vehicles required.

For problem three, it is necessary to propose a better offensive plan for the red side and a better retreat plan for the blue side in the case of red side attacking and blue side defending, as well as to analyze the different points of the overall retreat plan for the blue side in the case of good communication and communication disruption. For this problem, firstly, the optimal combat strategy provided by problem one and problem two is combined with an **ant colony algorithm - path optimization model** to derive a better offensive solution for red and a better retreat solution for blue in the case of red attacking and blue defending, as well as the overall retreat solution for blue in the case of good communication and communication disruption differing in terms of distance moved, order, path and time spent in retreat.

Key word: Dynamic Joint Model；Combat strategy optimization；Met acellular automata model；Ant colony optimization algorithm；Immuno-optimization algorithm

Content

1. Introduction

1.1 Background

Since ancient times, the degree of marching formation is often the key to determine whether a blade war will be won. Before the war, both sides should consider the number of soldiers, quantity and quality of weapons, topography, material transport and other factors to form the optimal battle formation, so as to give play to the advantages of various weapons. However, modern war is usually a local war under high-tech technology. The battlefield situation is changeable, and the various support structures are complex. The weapons used in modern war have strong lethality, wide range of effects and high cost.

Taking all these factors into account, both sides in modern warfare need to develop effective operational strategies that increase the threat of war while reducing losses on all fronts, and only if both sides form a more balanced, stable and mutually supportive operational posture can a consensus be reached as soon as possible. As the saying goes, war begins in the minds of men, so it is necessary to build a barrier of peace in the minds of men. The aim of modern warfare is not to win unilateral victories by force, but to promote the common good of long-term global security and development.

1.2 Work

We need to build a mathematical model based on the relationship between the coordinates of the base distribution nodes of the red and blue armies, as well as the difficulty of attack, marching distance, weapon range and air defense deployment of each node to calculate the size of the distribution positions and quantities of troops and weapons for both sides, and to find the optimal command positions and a number of alternative positions for both sides.

After arriving at the optimal site selection results, we need to build an optimized model for the distribution and supply of medical supplies, military supplies and daily supplies for both sides, and provide the number of workers and vehicles required in the non-supply mode, and finally present the optimal supply plan in the form of a graph.

Combined with the optimal supply plan given earlier, we need to propose the optimal attack plan for the red side and the optimal retreat plan for the blue side under the premise that the red side is attacking and the blue side is defending. And based on the given blue retreat node comparison analyze the difference between the overall retreat plan of the blue side in the two scenarios of good communication and communication breakdown.

2. Problem analysis

2.1 Data analysis

The data in this paper come from Annex 1 and Annex 2 of the title file package, including the coordinate positions of the red and blue assignable nodes, the distance between nodes of attack difficulty coefficient, and the parameter values of various weapons of the two armies.

2.2 Analysis of question one

For problem one, according to the data given in the annex, the dynamic joint model is established by combining the attack difficulty, marching distance, weapon range and air defense deployment of each node of red and blue sides, calculating the assigned positions and quantity scale of infantry and weapons of both sides, and then using immune optimization algorithm to calculate the optimal command positions and several alternative positions of both sides.

2.3 Analysis of question two

For problem two, medical supplies, military supplies, and daily supplies are grouped into the category of munitions, and the optimal command positions and alternative positions derived from problem one are combined to perform a comprehensive cluster analysis to derive the location of munitions drops. A meta-automata model is then used to calculate the quantity of materiel to be dropped and the total number of workers and vehicles required.

2.4 Analysis of question three

For Problem three, the optimal combat strategy provided by Problem one and Problem two is combined with an ant colony algorithm-path optimization model to plan the optimal attack plan for the red army and the optimal retreat plan for the blue army in the case of a red attack and a blue defense. A comprehensive comparison is made between the different retreat plans of the blue side in the case of good communication and communication disruption.

3. Symbol and Assumptions

3.1 Symbol Description

Symbol	Meaning
P	Number of troops
X_n	Difficulty of attack
α; β	The ratio of node position to number of troops; Node attack rate
S	Area of region
W	Combat motion distance
D_S; D_i	Weapon range span; Distance of march
G_i; G_s	The red and blue square indicates the distance between two nodes
C	Number of battlefield nodes
R_s	The distance between any two nodes on either side
B_i; C_i	Number of drones; Number of self-propelled guns
ax_v	The affinity of the antibody v to the antigen
opt_v	The binding strength of the antibody v to the antigen
f_v	The adaptation value of antibody v
$\mathrm{ay}_{\mathrm{v,w}}$	The affinity of antibody v and antibody w
e_v; c_v	The expected value of antibody v; The density of antibody v
q_k	An antibody with a greater affinity for antibody k
P_m	a soldier moving to the next meta-cell
L_m	the number of soldiers
Y_m	the number of all soldiers in a single direction
P_{ab}	a skilled soldier moving to the next cell
k_s	the static field parameter
ε_{ab}	tuple state parameters
$P_{ij}^k\ (t)$	The probability of soldier k moving from node i to node j
$\Delta\tau_{ij}^k$	Amount of pheromone
L_K	The length of the path taken by the kth soldier in this traversal

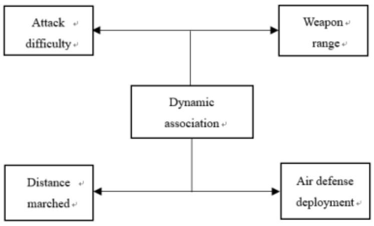
3.2 Fundamental assumptions

1. Assume that the acquired data has authenticity and reliability.
2. Assume that during the battle, the weapon attacks are directed to each distribution node.
3. Assume that medical supplies, military supplies and daily supplies are classified as munitions.
4. Assume that when the red army attacks in problem three, the blue army only defends but does not attack.
5. Assume that the soldiers in problem three will not return after attacking a battle node and continue to attack other battle nodes.

4. Model

4.1 Model of question one

The confrontation between the red and blue sides requires consideration of the intricate effects arising from various factors such as the difficulty of attack, distance travelled, range of weapons and air defense deployment of each node, so that comprehensive consideration is required in determining the allocated positions and quantitative scale of various weapons. Based on the references and surveys, a dynamic joint model consisting of a combination of four sub-models is developed and the specific procedures for the four sub-models are as follows[1]:



1.Attack difficulty based troop layout -- attack difficulty distribution model.

The attack difficulty of each node is the basis of rational layout, based on the determination of the attack difficulty of each node for macroscopic regulation of troops, the mathematical relationship between the number of troops P and the attack difficulty X_n (n=1、2、…、436):

$$X_n = \alpha\beta P \tag{1}$$

Where α is the ratio of node position to number of troops, based on the references, the ratio between node position and number of troops is kept between 2% and 3.5%, and β represents the rate of nodes being attacked. The rate of nodes being attacked is represented by the average rate of nodes being attacked, which is kept at 30%.

2. A strategic layout model for warfare based on the distance of attack - -a rational model of the distance travelled.

The positions of the nodes in the war zone are distributed so that the mathematical relationship between the number of troops P, the area of the zone S and the distance W to be moved during the battle can be deduced as:

$$P = \frac{S}{4W^2} \tag{2}$$

The area is considered to be a circle with a radius of 1km, with the node coordinates as the center. The "Rationalized distance" is defined as the single best distance that a soldier can travel from the point of attack to the enemy's point of attack, and the number of troops at that distance is the number of troops that can basically satisfy the enemy's base point. After reviewing the references, it can be seen that a reasonable distance for soldiers to travel is 1000-2000 meters, with a round trip time of approximately 10-20 minutes. The normal walking speed of a soldier is approximately

100 meters per minute, so when the number of nodes corresponding to this is an area of 1 kilometer, the red side arranges 100-127 troops to meet the minimum requirements for the attack. The Blue side has 78-89 troops to meet the minimum requirements to achieve the objective during the attack.

3. Layout based on weapon range - the weapon range span model.

When two sides are at war, the equipment they carry and the range of that equipment have an impact on the effectiveness and efficiency of the attack. In order to ensure that they can move easily and hit others from the longest distance, both sides carry the right number of weapons with the right range.

Based on the maximum effectiveness of the attack, the weapon range span is modeled; the weapon range span is the number of D_S soldiers as P, C is the number of battlefield nodes, G_i is the distance between two nodes on the red side, G_s is the distance between two nodes on the blue side, R_s is the distance between any two nodes on the red and blue sides:

$$D_s = \frac{T\ (1+G_i)\ (1+G_s)}{C(1+R_s)} \tag{3}$$

4. Rational layout based on air defense deployment - a model for the distribution of air defense deployment numbers.

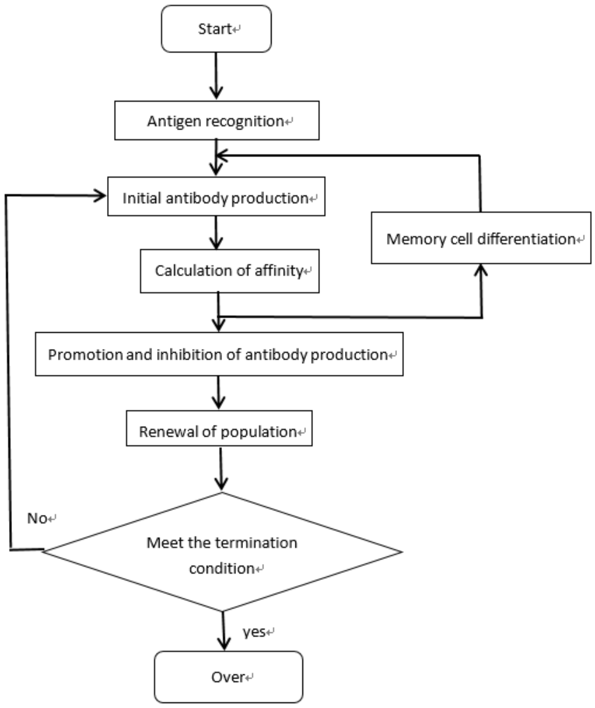
In order to avoid the destruction of ground bases by enemy air power, the Red and Blue sides deploy their air defense against each other's strengths. In this war, the number of drones, self-propelled artillery and anti-aircraft artillery are the main forces in the air attack. Based on the above conditions, a model is developed for the distribution of the number of air defense deployments:

The number of drones used by Red is B_1 $(0 \le B_1 \le 500)$ and the number of self-propelled guns used is C_1 $(0 \le C_1 \le 7000)$; the number of drones used by Blue is B_2 $(0 \le B_2 \le 300)$ and the number of self-propelled guns used is C_2 $(0 \le C_2 \le 14000)$; the attack difficulty of each node is X_n, the attack difficulty of both sides is X_{redn} and X_{bluen}, and the distance travelled is D_i.

Red air defense deployment allocation model: $Z_{red} = \frac{B_2 \times X_{redn} + C_2 \times X_{redn}}{14300}$ (in this equation X_{redn} belongs to the attack difficulty of each node on the red side)

Blue air defense deployment allocation model: $Z_{blue} = \frac{B_1 \times X_{redn} + C_1 \times X_{bluen}}{7500}$ (in this equation X_{bluen} belongs to the attack difficulty of each node on the blue side)

Basic flowchart of the immunization algorithm[2]:



The first step identifies the antigen and uses the target function and constraint as the antibody; The second step generates the initialized antibodies and randomly generates *N* antibodies of unique type string dimension *M*; The third step is calculate the affinity;

The affinity of antibody *v* and antigen is ax_v:

$$ax_v = \frac{1}{1+opt_v} \tag{4}$$

Where opt_v denotes the binding strength of antibody *v* and the antigen. For optimization problems, this can be expressed in terms of the similarity between the solution of the unique type of antibody *v* and the known optimal solution. If it is a maximization problem, then $opt_v = \frac{f_v - f_{max}}{f_{max}}$, where f_{max} is the fitness value of the optimal solution and f_v is the fitness value of the antibody *v*.

The affinity of antibody *v* and antibody *w* is:

$$ay_{v,w} = \frac{1}{1+E(2)} \tag{5}$$

And where *E(2)* is the average information entropy of *v* and *w*.

The fourth step is memory cell differentiation, where the newly generated antibody will replace the memory cell with which it has the greatest affinity, since the number of memory cells is limited. The fifth step antibody promotion and inhibition, by calculating the expected value of antibody *v* and eliminating antibodies with low expected value, the expected value of antibody *v*, e_v is calculated by the formula[3]:

$$e_v = \frac{ax_v}{c_v} \tag{6}$$

And where the density of antibody v is calculated as:

$$c_v = -\frac{q_k}{N} \tag{7}$$

where q_k denotes an antibody with greater affinity to antibody k.

The sixth step is to generate new antibodies based on the high or low affinity of different antibodies and antigens. The seventh step ends the algorithm if the derived optimal solution satisfies certain end conditions.

4.2 Model of question two

The difficulty of the attack for both Red and Blue, the distribution of node locations and the soldiers' familiarity with the location of the battlefield are combined with the infantry walking distance model from Problem one. A modified meta-automaton model is used in this problem.

The model is improved by adding the three main factors of attack difficulty, number of nodes and soldier movement rules to the traditional met-acellular automaton model. The cell size is 0.5m*0.5m, one cell intelligently holds one soldier, and the cell is divided into three states: vacant, occupied by obstacles and occupied by enemy soldiers. The tuple automaton model is divided into nodes with a larger number of soldiers and nodes with a smaller number of soldiers, depending on the number of nodes. In a single time step, soldiers can only move one tuple in eight directions around the central tuple, and each tuple can only hold one pedestrian. This is shown in the diagram. Also soldiers encounter an enemy or an obstacle as they travel as follows: R indicates the ability to see within that range for 0.5m, where black indicates an enemy soldier and grey indicates an obstacle encountered.

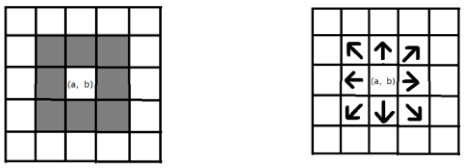


Figure 1 Soldier movement orientation map

In actual warfare, by observing the placement of enemies and obstacles within visibility, as well as the respective directions of movement of soldiers on both sides and the frequency of direction changes, it is possible to determine whether the infantry on both sides are familiar with the scenario environment[4]. It is also equipped with medical supplies and total number of vehicles. Considering that soldiers in the same nodal camp with less familiarity with the battlefield environment tend to follow the soldiers with more familiarity with the battlefield.

Define the probability of a soldier moving to the next meta-cell as P_m,

$$P_m = L_m \left(\sum_{m=1}^{8} Y_m\right)^{-1} \tag{8}$$

where m is the eight directions of meta-cell movement, taking values in the range [1,2,. .8], L_m is the number of soldiers that can be selected to follow in a single direction in the visibility range, Y_m is the number of all soldiers in a single direction in the visibility.

Define the probability of a skilled soldier moving to the next cell as P_{ab},

$$P_{ab} = N\exp\ (k_s S_{ab})\ (1 - n_{ab})\ \varepsilon_{ab} \tag{9}$$

where N is the normalization factor, k_s is the static field parameter, $k_s \in (0, \infty)$, and S_{ab} is the static field, indicating the arrival proportion of soldiers arriving at the planned material drop point; the closer the soldiers are to the material drop point, the greater the arrival proportion.ab is the coordinate of the current tuple position, n_{ab} and ε_{ab} are tuple state parameters, $n_{ab} = 1$ if the tuple is occupied by an obstacle, $n_{ab} = 2$ otherwise, and $\varepsilon_{ab} = 0$ if the tuple is occupied by an enemy army, $\varepsilon_{ab} = 1$ otherwise[5].

$$N = [\sum_{ab} \exp(K_s S_{ab})(1 - n_{ab})\varepsilon_{ab}]^{-1} \tag{10}$$

$$S_{ij} = \max_{(a,b)}\left\{\min_{i_{ek}} \sqrt{(i_{ek} - i)^2 + (j_{ek} - j)^2}\right\} - \min_{i_{ek}} \sqrt{(i_{ek} - i)^2 + (j_{ek} - j)^2} \tag{11}$$

where $(i_{ek},\ j_{ek})$ is the coordinates of the position of the arranged material camp, $k = 1,2,3$.

The allocated quantity R of material for each static field can be reallocated by the formula:

$$R = \frac{\max_{(a,b)}\left\{\min_{i_{ek}} \sqrt{(i_{ek} - i)^2 + (j_{ek} - j)^2}\right\} - \min_{i_{ek}} \sqrt{(i_{ek} - i)^2 + (j_{ek} - j)^2}}{\sqrt{\alpha_{red}{}^2 + \alpha_{blue}{}^2 \times N}} \times P \tag{12}$$

4.3 Model of question three

In question three, it is considered that each soldier will be in a different battle position (home and enemy) at different battle moments, and will also leave the appropriate information at the time of the battle (leaving clues to the home side or confusing the enemy's attack), while also being able to be in different battle node positions within the shortest distance. This problem adopts an ant colony optimization algorithm, which is performed as follows[7].

At the initial moment of the algorithm, *m* soldiers are randomly placed in *n* combat nodes, while the first element of each soldier's *tabu* of the taboo table is set to the position of the node where it is currently located.

At this point the amount of pheromones on each path is equal, and let $T_{ij}\ (0)\ = c$ (*c* is a smaller constant). Next, each soldier independently selects the next combat node based on the amount of pheromones remaining on the path and heuristic information (distance between two nodes), and at moment *t*, the probability that soldier *k* moves from node *i* to node *j*, $P_{ij}^k\ (t)$ is:

$$P_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta}}{\sum_{S \epsilon J_k(i)} [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta}} & \text{where } j \epsilon J_k(i) \\ 0 & \text{others} \end{cases} \tag{13}$$

where $j_k\ (i)\ = \{1,2,\dots,n\} - tabu_k$ denotes the set of combat nodes that soldier k is allowed to select next. The taboo table $tabu_k$ records the combat nodes that soldier *k* has currently traveled. When all n battle nodes are added to the taboo table $tabu_k$

(which is added to the taboo table after each battle node the soldier passes), soldier k has completed a traversal. The n_{ij} in Eq. (13) is a heuristic factor indicating the expected degree of soldier's transfer from node i to node j. In the algorithm, n_{ij} is usually taken as the derivative of the distance between node i and node j. The α and β in this equation denote the relative importance of the pheromone and the desired heuristic factor, respectively. When all soldiers have completed a traversal, the pheromones on each path are updated according to equation (14)[8].

$$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij} \tag{14}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij} \tag{15}$$

where $\rho(0 < \rho < 1)$ denotes the evaporation coefficient of the pheromone on the path (the information left by the soldier does not last forever) and 1-ρ denotes the persistence coefficient of the pheromone $\Delta_{\tau_{ij}}$.

The increment of pheromones on edge ij in this iteration, $\Delta\tau_{ij}^{k}$ denotes the amount of pheromones left on edge ij by the kth soldier in this iteration. If soldier k does not pass through edge ij, the value of $\Delta\tau_{ij}^{k}$ is zero. $\Delta\tau_{ij}^{k}$ is denoted as:

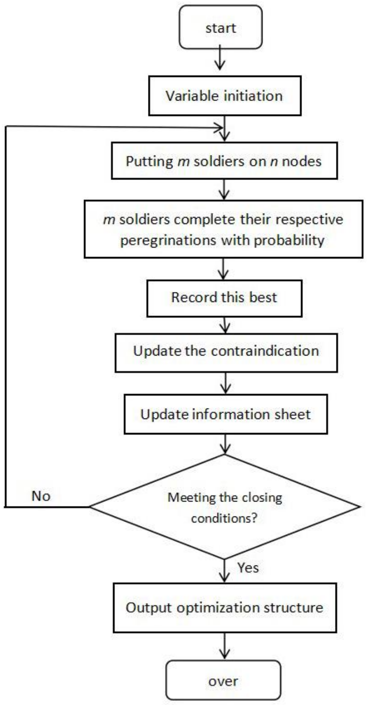
$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_K}, & \text{When soldier } k \text{ passes through edge } ij \text{ in this traversal} \\ 0, & \text{others} \end{cases} \tag{16}$$

where Q is the normal number and L_K denotes the length of the path taken by the *kth* soldier in this traversal[9].

The specific implementation steps of the basic ant colony algorithm are as follows.

1. Parameter initialization. Let the time $t=0$ and the number of cycles $N_c = 0$, set the maximum number of cycles G, place m soldiers on n elements (combat nodes), and let the initialization information $\tau_{ij}(t) = c, \Delta\tau_{ij}(t) = c$ for each edge (i, j) on the directed graph, where c denotes a constant and the initial moment $\Delta\tau_{ij}(0) = 0$.
2. The number of cycles $N_c = N_c + 1$.
3. Soldier's tabulation index number $k = 1$.
4. Number of soldiers $k = k + 1$.
5. The individual soldier selects element j and advances according to the probability calculated by the state transfer probability formula (1), $j \in J_k(i)$.
6. Modify the taboo table pointer, i.e. move the soldier to a new element after selection and move that element to the taboo table of that ant individual[10].
7. If the elements in the set C are not traversed, i.e. $k < m$, then skip to step 4; otherwise, perform step 8.
8. Record the best route this time.
9. Update the amount of information on each path according to Eq. (2) and (3).
10. If the end condition is satisfied, i.e. if the number of loops $N_c \geq G$, the loop ends and the program optimization result is output; otherwise, clear the taboo table and skip to step 2.

The flow chart of the algorithm is as follows：



5. Test the Models

5.1 Test the model of question one

The relevant graphs were derived from solving the data as follows:

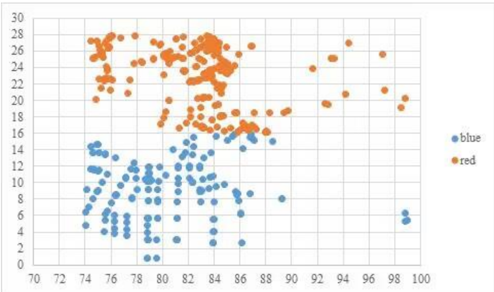


Figure 2 Distribution of Red and Blue assignable nodes

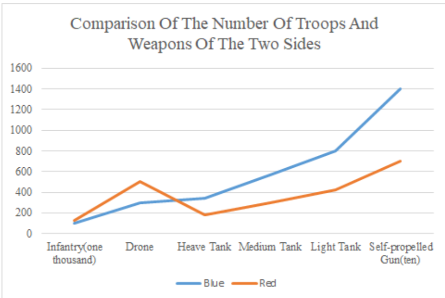


Figure 3 Comparison of the number of troops and weapons between red and blue

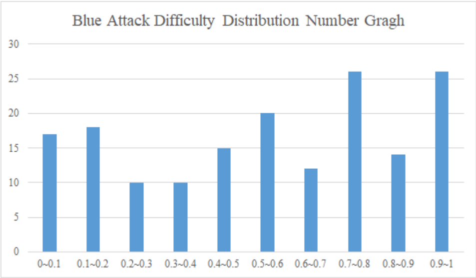


Figure 3 Distribution of Blue's attacks difficulty factors

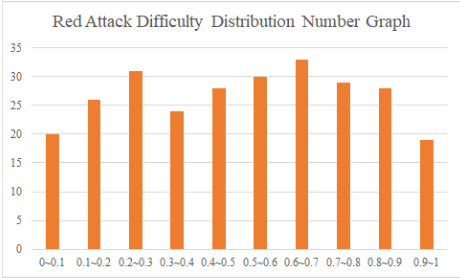


Figure 4 Distribution of Red's attacks difficulty factors

Although the Blue army has fewer infantry and drones than the Red army, it has more heavy tanks, medium tanks, light tanks and self-propelled artillery than the Red army. According to the distribution range of attack difficulty coefficients of both Red and Blue armies, the expected value of attack difficulty coefficient of Blue army is 0.5428 and the expected value of attack difficulty coefficient of Red army is 0.5044, comparing the expected values of both armies, we can see that the combined attack difficulty of all nodes of Blue army is higher than that of Red army.

The formula is used to calculate the size of the allocated positions and numbers of infantry, tanks, self-propelled artillery and anti-aircraft guns for each node on the red and blue sides (***see appendix table9-12 for details of the allocation results***):

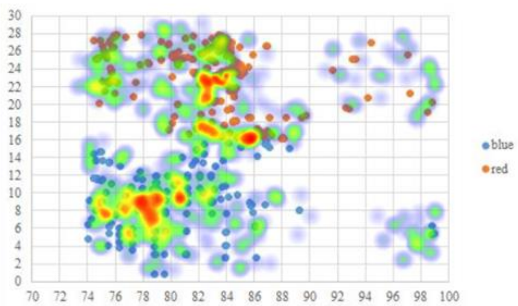


Figure 5 Heat map of the clustering of forces on both sides

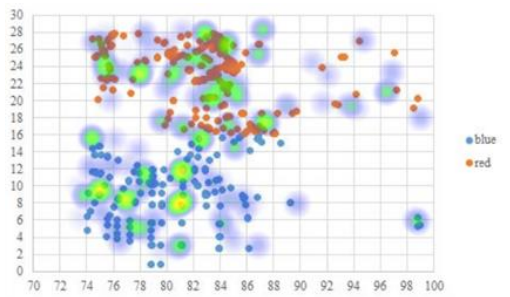


Figure 6 Heat map of tank clustering distribution for both sides

According to the heat map of the distribution of troops between the red and blue sides, it can be concluded that the forces of both sides are more concentrated in the following three areas:

Table 1 Red and blue forces distribution area location coordinates

	Red	Blue
Regional Location	$\{(x,y)\|82<x<84;\ 20<y<23\}$	$\{(x,y)\|77<x<80;\ 6<y<10\}$
	$\{(x,y)\|85<x<86;\ 15<y<17\}$	$\{(x,y)\|80<x<81;\ 7<y<10\}$
	$\{(x,y)\|81<x<83;\ 16<y<18\}$	$\{(x,y)\|75<x<85;\ 7<y<8\}$

According to the heat map of the distribution of the number of tanks between the red and blue sides, it can be concluded that the forces of both sides are more concentrated in these areas:

Table 2 Red and blue number of tanks distribution area location coordinates

	Red	Blue
Regional Location	$\{(x,y)\|75 < x < 76;\ 23 < y < 25\}$ $\{(x,y)\|77 < x < 79;\ 22 < y < 24\}$ $\{(x,y)\|83 < x < 85;\ 19 < y < 22\}$	$\{(x,y)\|74 < x < 76;\ 8 < y < 10\}$ $\{(x,y)\|76 < x < 78;\ 7 < y < 9\}$ $\{(x,y)\|80 < x < 82;\ 7 < y < 9\}$ $\{(x,y)\|80 < x < 82;\ 9 < y < 13\}$

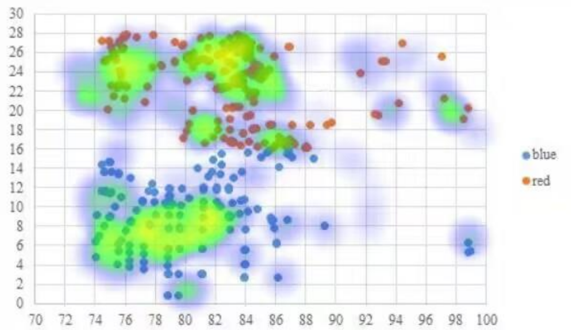


Figure 7 Heat map of the clustered distribution of the number of artillery pieces on each side

According to the heat map of the distribution of the number of guns between the red and blue sides, it can be concluded that the forces of both sides are more concentrated in these areas：

Table 3 Red and blue number of guns distribution area location coordinates

	Red	Blue
Regional Location	$\{(x,y)\|74 < x < 78;\ 22 < y < 26\}$ $\{(x,y)\|80 < x < 86;\ 20 < y < 28\}$	$\{(x,y)\|74 < x < 83;\ 4 < y < 10\}$

Combining the attack difficulty, marching distance, weapon range, and air defense deployment of each node on the red and blue sides, the assigned positions and quantitative scales of infantry, tanks, self-propelled artillery, and air defense artillery as well as the optimal command positions and alternative positions for the red and blue sides are derived as follows：

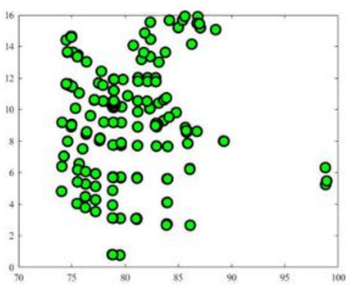


Figure 8 Distribution of the nodes of the blue army

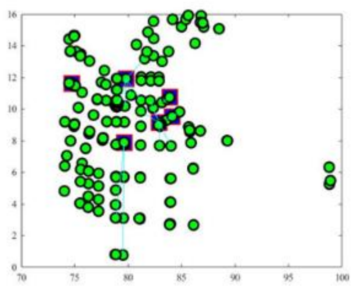


Figure 9 Blue command positions and alternative positions distribution map

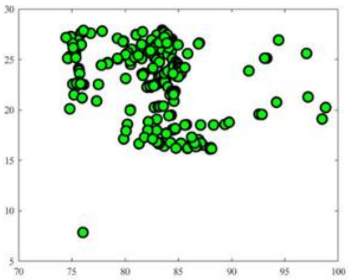


Figure 10 Distribution of the nodes of the red army

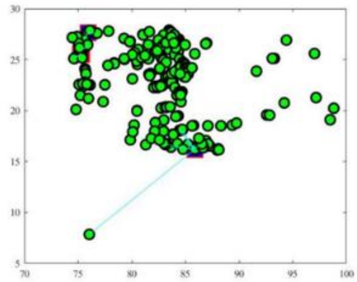


Figure 11 Red command positions and alternative positions distribution map

Table4 Coordinates of the optimal command positions and alternative positions for both red and blue

Coordinates	Red	Blue
Optimal command position	M（75.2444，25.5122） N（85.8568，16.1892）	A_1（74.7032，11.6198）;A_2（79.6279，7.8881）;A_3（79.8001，11.9088）
Alternative positions	Q（76.0419，27.8537）	D（83.6385，10.6162）；E（83.0695，9.0743）；F（83.6089，9.3024）

5.2 Test the model of question two

Considering that munitions are not carried by the soldiers, they need to be concentrated at the drop points. By using cluster analysis, the optimal command positions and alternative positions are combined with the above-mentioned factors to obtain a map of the location of the drop points (after optimization):

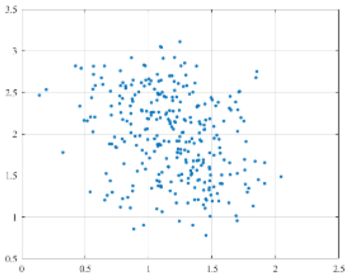


Figure 12 Red side military supplies distribution map

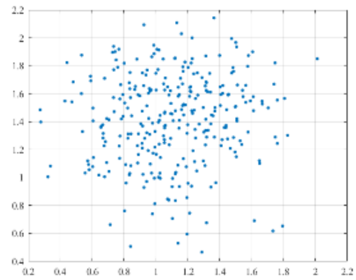
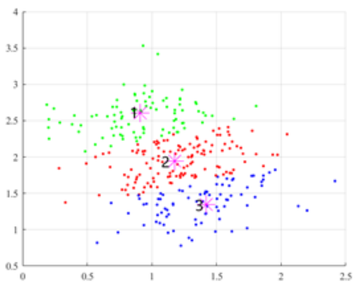


Figure 13 Blue side military supplies distribution map



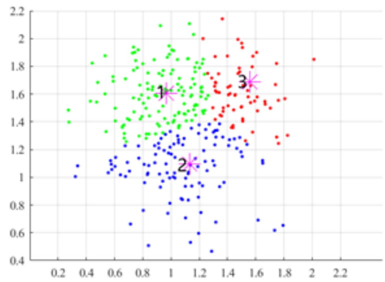


Figure 14 Red side location distribution map of military supply drop-off points

Figure 15 Blue side Location distribution map of military supply drop-off points

By reviewing the information, materiel deployment stations were established at the optimized location nodes on the red and blue sides and the corresponding materiel was placed.

Table5 The recommended number of materiel to be deployed at each location is (in units/units)

red			blue		
Location 1	Location 2	Location 3	Location 1	Location 2	Location 3
400000	500000	350000	330000	350000	320000

By setting the calculation, the simulation scenario is defined as a simulated 3D battlefield of 60km*60km*40km, the white area indicates the area where soldiers from both sides can walk, the black area indicates the battle boundary (the battle area cannot be expanded indefinitely), and both sides march with a certain amount of troop loss.

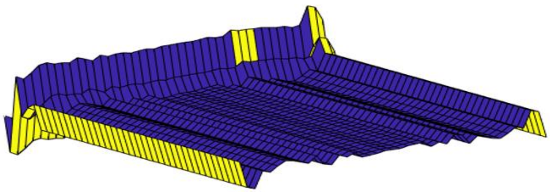


Figure 16Three-dimensional simulated battlefield map

The simulation time is measured in time steps and now simulates the combat process of both sides, using a countdown timer to make a preliminary judgement on the combat situation of both sides. As shown in the diagram below, the individual white dots represent the strength of your side, the change in the small white dots represents the direction of movement of the cases and the number of white dots represents the number of troops.

Figure 17 Start of battle (red side)

Figure 18 In battle (red side)

Figure 19 After the battle (red side)

Figure 20 Start of battle (blue side)

Figure 21 In battle (blue side)

Figure 22 After the battle (blue side)

By reviewing the information, material deployment stations were established at optimized location nodes on the red and blue sides and the corresponding material was dropped.

Table6 The recommended quantities of supplies to be placed at each location in the supply mode (in units/units)

Red side			Blue side		
Location 1	Location 2	Location 3	Location 1	Location 2	Location 3
801235	987600	1248765	651273	689412	726500

Table 7 Table of changes in the number of infantry on the red and blue sides

Teams	Start of battle	In battle	After the battle	Percentage of lost troops
Red	1250000	952371	656891	47.2815%
Blue	1000000	757412	682147	31.7853%

By comparing the troop losses of both sides in different phases of the battle, it is found that the blue side loses less troops than the red side. Therefore, based on the potential attack strategy of the opponent, the red side needs to draw more workers for internal support and more vehicles for the corresponding material transportation. The total number of workers and the total number of vehicles required for the red and blue sides in the non-supply mode are reprogrammed by the formula, the results are as follows:

Table8 Optimized quantity distribution table for each material

	Original number of troops	Total number of workers required in the non-supply model	Total number of vehicles	Total quantity of military supplies
Red（Node I）	412865	82573	3213	829279
Red（Node 2）	537263	107452	3957	805894
Red（Node 3）	299872	59974	2469	408953
Blue（Node I）	286549	57309	2368	458478
Blue（Node 2）	337213	67442	3372	539540
Blue（Node 3）	376238	75248	2841	526733

5.3 Test the model of question three

Through the above algorithm steps, the coordinates of each node position on the red and blue sides are input, an initial point is randomly generated at the existing nodes on the red side, an initial route is randomly generated by calculating the distance matrix of the nodes and the number of combined nodes, the path diagram of the random solution is drawn and the path and total distance of the random solution are output.

By gradually increasing the number of iterations to initialize the objective value matrix to generate new solutions, the Metropolis law determines whether to accept new solutions and selectively incorporates the current new solutions into the library until the optimal solution and the sum of information are finally output. (*During the red side attack, soldiers start moving from each node on the red side to the blue side, that is, all the location coordinates are taken into account to make a better attack plan*)

The nodes of both sides are all included in the path to be chosen, and the four sub factors *(difficulty of attack, distance travelled, weapon range and air defense deployment)* considered in problem one are combined to generate a new solution and optimize it with the goal of finding the shortest distance and maximum offensive benefit: the total distance of the original red scheme attack is 4817.6407km, and the total distance of the optimized attack is 1477.7774km . ***(Red's original attack node roadmap and optimized attack node roadmap and blue side original retreat node roadmap and optimized retreat node roadmap see appendix for details)***

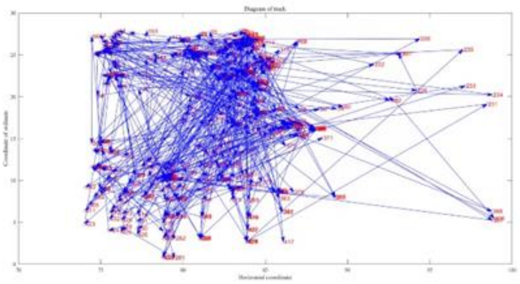


Figure23 Red's original attack roadmap

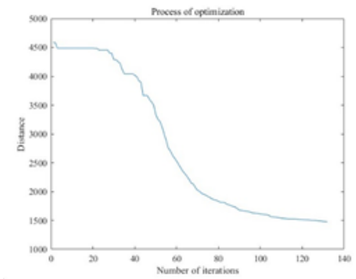


Figure24 Red side attack optimization iteration diagram

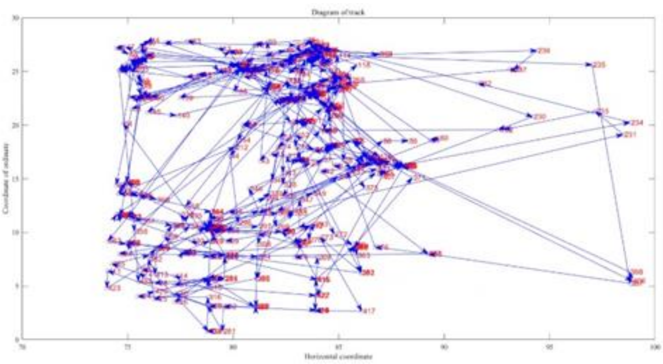


Figure25 Red's optimized attack roadmap

When Blue's retreat node is [37,140,378], the coordinates of the three retreat nodes are: B_1 (74.0826,9.1888); B_2 (79.5065,0.7677); B_3 (86.1134、2.6653).

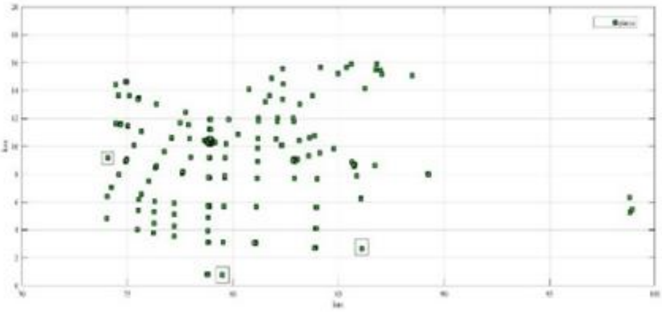


Figure 26 Blue Square Communication good time retreat node distribution map
(Mark the blue retreat node [37,140,378])

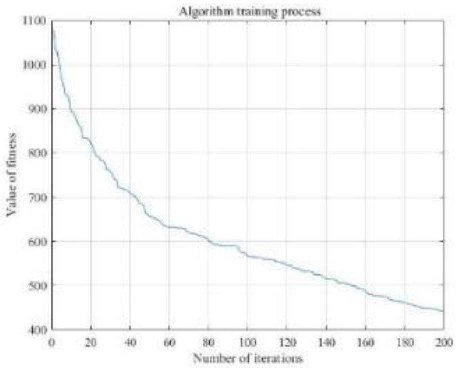


Figure 27 Number of fixed retreat node training on the blue side

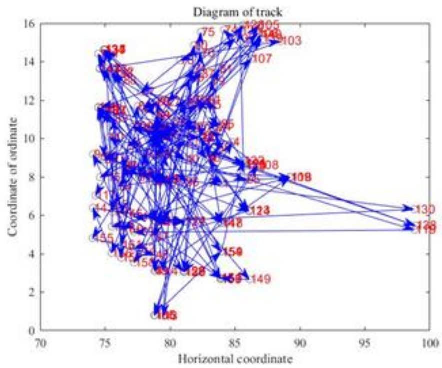


Figure 28 Blue side original retreat path planning

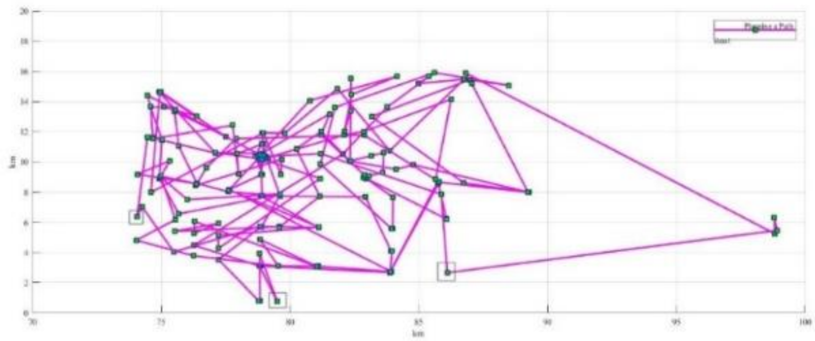
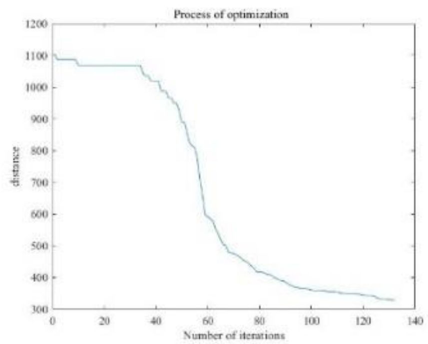


Figure 29 Path planning for retreating nodes when blue party communication is good



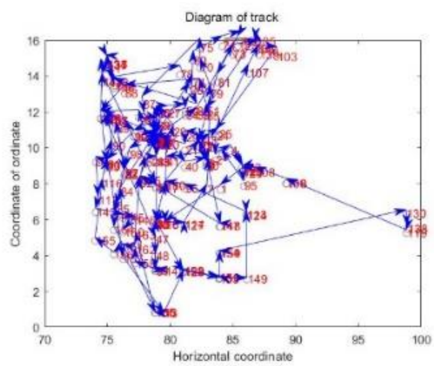


Figure 30 Number of iterations of the retreat path optimization process on the blue side

Figure 31 Blue's optimized retreat path

Combine the three optimal command positions of the blue army selected in Problem 1 to carry out a comprehensive scheduling retreat, avoiding to the maximum extent possible traveling to the node positions already occupied by the red side in the retreat process. Denote the coordinates of the three optimal command locations on the blue side as: A_1 (74.7032, 11.6198); A_2 (79.6279, 7.8881); A_3 (79.8001, 11.9088).

In the case of good communication, the blue command position will retreat information to the nodes after the army, the overall army to evacuate, taking into account the shortest retreat distance, retreat more efficient, command troops to retreat to the nearest retreat node location, the blue retreat road map is as follows *(The yellow line represents the original retreat route and the blue line represents the optimized retreat route)*:

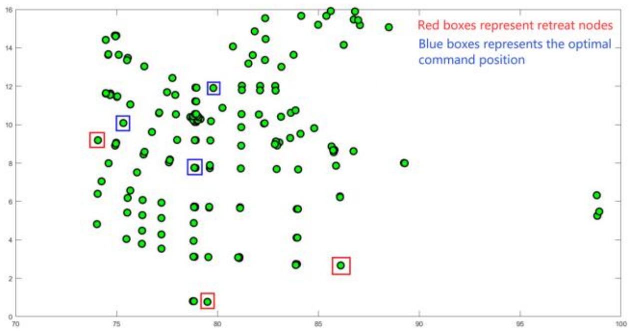


Figure 32 Blue side optimal command positions and retreat node location distribution map

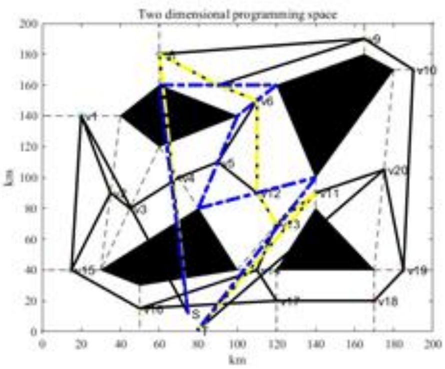


Figure 33 Retreat to node B_1 trajectory map

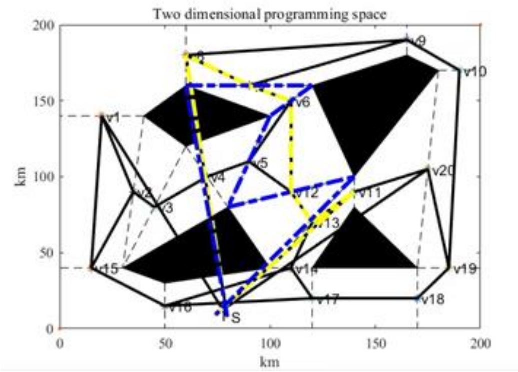


Figure 34 Retreat to node B_2 trajectory map

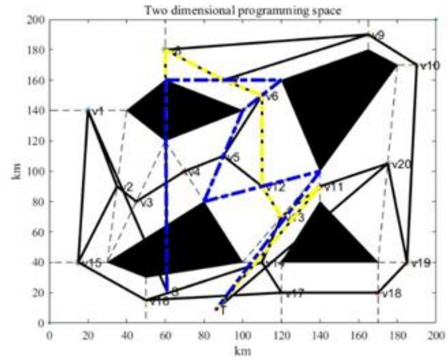


Figure 35 Retreat to node B_3 trajectory map

In the communication interruption, it will lead to the blue side of the command center can not be timely retreat information to each node of the deployment of the army, so the blue army retreat program for a re-regulation of planning to reach the retreat node distance nearest to the principle, a comprehensive problem in the four sub-factors, combined with the ant colony optimization algorithm model, resulting in the following retreat program:

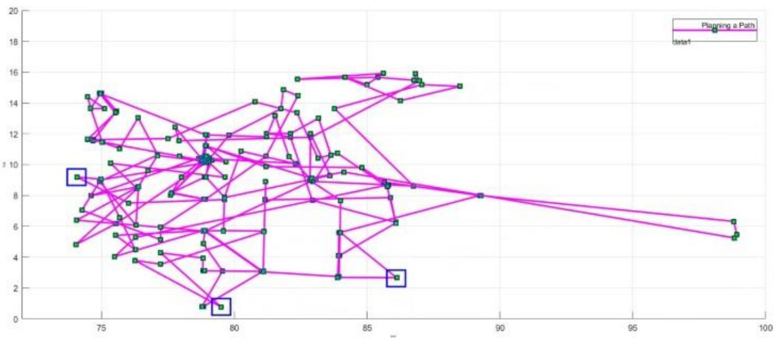


Figure 36 Path planning for retreating nodes when blue party communication interruptions

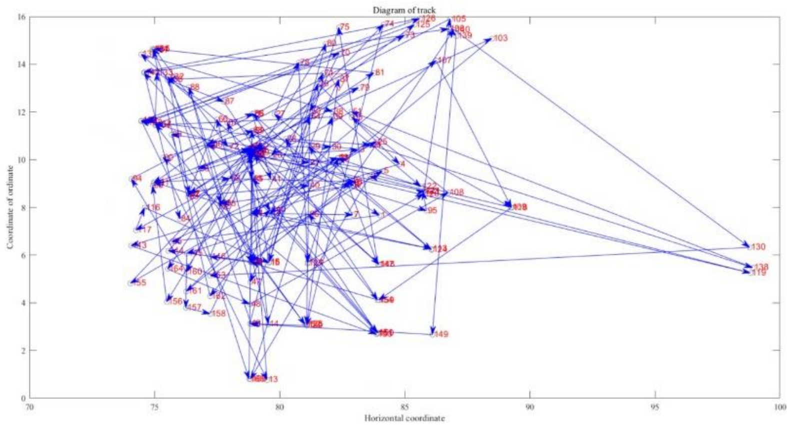


Figure 37 Blue side original retreat path planning

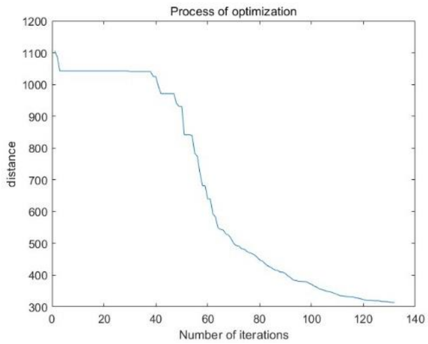


Figure 38 Number of iterations of the retreat path optimization process on the blue side

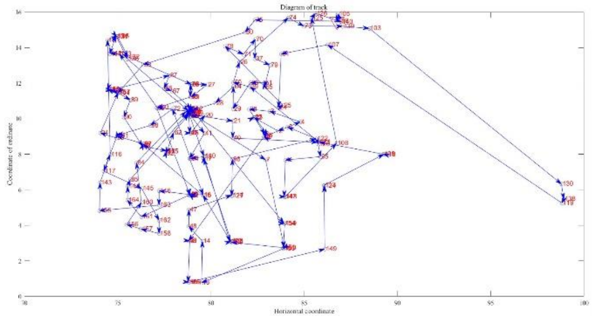


Figure 39 Blue's optimized retreat path

In the case of good communication, the blue side can retreat in a more orderly manner and the overall movement distance can be significantly reduced, i.e., the soldiers of each node can be made to reach the retreat node in a short time. In the case of signal disruption, it may make some nodes' soldiers unable to receive the retreat notification smoothly, more by observing the behavior of soldiers in nearby nodes to make the corresponding response. Moreover, it is not possible to reach the retreat node in a straight line *(considering the obstacles and the location captured by the enemy)*, so it is necessary to rendezvous with soldiers from other nodes for overall movement first, which is less effective than the case of good communication in terms of order and time during retreat.

6. Strengths and Weakness

6.1 Strengths

For problem one, the immune optimization algorithm finds the global optimal solution by optimal immune advantage, which has the advantages of global convergence, adaptivity and parallelism.

For problem 2, the meta-cellular automata model has simple rules, low resource consumption, and fast running speed.

For problem three, the search process of the ant colony path optimization algorithm adopts a distributed computation method, in which multiple individuals perform and compute simultaneously, which improves the computational capacity and operational efficiency of the algorithm, and adopts a positive feedback mechanism, which makes the search process converge continuously and finally approaches the optimal route.

6.2 Weakness

For problem one, the immune optimization algorithm runs fast but requires a global learning rate.

For problem two, the application of the metacell state transition rule is limited.

For problem three, the ant colony path optimization algorithm, if there is too much diversity, it will lead to too much random motion and fall into chaos; if there is not enough diversity and too much positive feedback, it will lead to rigidity and the ant colony cannot adjust accordingly when the environment changes.

7. Conclusion

Recommendation

Both combatants in modern warfare need to develop effective combat strategies to check and balance the combat power of both sides, so as to achieve the maximum common interests of both sides.

Firstly, based on the initial deployment status of both red and blue armies, a dynamic joint model is established based on the arithmetic results of node attack difficulty, short marching distance, optimal weapon range and air defense deployment, and the immune optimization algorithm is used to calculate three optimal command position area coordinates and three alternative position area coordinates for the blue side, and two optimal command position locations and one alternative position location coordinates for the red side.

Secondly, after getting the optimal position location coordinates, the optimization model of distribution and supply of munitions on both sides is established to get the distribution of the number and location of munitions drop points, and the number of workers and vehicles required in the non-supply mode is provided to reach the optimal supply plan based on the enemy's potential attack strategy.

Combined with the optimal supply plan given earlier, the optimal attack plan for the red side and the optimal retreat plan for the blue side are derived using the ant colony optimization algorithm under the premise that the red side attacks and the blue side defends. And according to the given blue side retreat node comparison analysis the difference of the overall retreat plan of blue side in two cases of good communication and communication interruption.

References

[1] Ge Xianlong,Wang Xu,Deng Lei. Research on open dynamic vehicle path problem and algorithm based on joint distribution[J]. Journal of Management Engineering, 2013, 27(03):60-68.DOI:10.13-587/j.cnki.jieem.2013.03.007.
[2] Liu Yahuan. Research on base station siting problem based on immune algorithm[D]. Dalian University of Technology,2016.
[3] Li Weibin,Wu Fang,Zhang Yan. Research on site selection of shared electric vehicle outlets based on immune algorithm[J]. Journal of Lanzhou Jiaotong University,2022, 41(01):34-39+51.
[4] Cao Binru,Wang Xia. Research on the optimization of multi-model vehicle distribution path based on dynamic demand[J]. Manufacturing Automation, 2021,43(06):79-87.
[5] Liu Wusheng, LI Wang, Zhou Qing, Dian Jian. Optimization model and algorithm for "AV-vehicle" delivery path[J]. Transportation Systems Engineering and Information,2021,21(06):176-186.DOI: 10.16097/j.cnki.1009-6744.2021.06.020.
[6] Huo Feizhou, Li Ying, Ma Yaping, Tu, Wenhao, Liu Chang, Yan Yan. A metacellular automata model considering environment familiarity and evacuation strategy[J]. Chinese Journal of Safety Science,2022,32(09):94-99.
[7] Xie Jiming, Peng Bo, Qin Yaqin. A meta-cellular automate model based on lane change probability distribution for multi-lane intertwined area[J]. Transportation Systems Engineering and Information,2022,22(03):276-285.DOI:10.16097/j.cnki.100--9-6744.2022.03.031.
[8]Gong Mingfan,Xu Haixiang,Feng Hui,Xue Xuehua. Intelligent ship path planning based on improved ant colony algorithm[J]. Journal of Wuhan University of Technology (Transportation Science and Engineering Edition),2020,44(06):1072-1076.
[9] Xu Kun. Research on the path planning of community express delivery based on improved ant colony algorithm [D]. Xinjiang University,2019.
[10] Fabio Lamantia. Emergence of competition and cooperation in an evolutionary resource war model[J]. Communications in Nonlinear Science and Numerical Simulation,2018,58.
[11] Paul Schuurman. Models of war 1770–1830: the birth of wargames and the trade-off between realism and simplicity[J]. History of European Ideas,2017,43(5).

Appendix

Table 9 Distribution of the number of infantry in each node on the blue side

Blue id	Infantry	Blue id	Infantry	Blue id	Infantry	Blue id	Infantry	Blue id	Infantry	Blue id	Infantry
25	1010	53	912	81	1021	125	195	300	717	368	804
26	358	54	1032	82	565	130	11	301	119	369	1053
27	586	55	597	83	413	131	836	304	304	372	554
28	619	56	977	84	641	132	174	305	109	373	706
29	380	57	880	85	87	133	641	306	250	374	847
30	1053	58	163	86	467	134	1075	307	152	375	999
31	543	59	445	87	825	135	11	308	206	376	771
32	967	60	195	88	608	138	87	311	738	377	684
33	543	61	565	89	163	139	532	312	119	378	1053
34	782	62	33	90	217	140	98	313	543	379	87
35	413	63	858	91	1032	143	500	314	76	380	217
36	967	64	695	92	847	144	521	329	500	381	825
37	815	65	586	93	847	160	988	330	1043	382	586
38	945	66	152	94	597	161	1021	331	652	383	988
39	272	67	304	95	760	162	315	332	282	384	532
40	1021	68	1053	96	456	163	804	333	847	385	1064
41	33	69	565	97	771	165	858	334	999	386	250
42	858	70	543	98	728	236	217	335	445	387	782
43	174	71	315	111	728	276	489	336	1021	388	673
44	87	72	358	112	858	288	771	337	738	428	1064
45	934	73	1010	113	999	292	630	338	1021	429	402
46	597	74	891	117	771	293	1043	339	793	430	999
47	565	75	912	118	804	294	901	340	652	431	706
48	163	76	1010	120	945	295	434	341	130	432	98
49	76	77	836	121	847	296	891	342	858	433	576
50	228	78	782	122	217	297	999	348	706	434	141
51	380	79	22	123	467	298	891	349	967	435	54
52	630	80	413	124	87	299	358	350	87	436	250

Table 10 Distribution of the number of infantry in each node on the red side

Red id	Infantry	Red id	Infantry	Red id	Infantry	Red id	Infantry	Red id	Infantry	Red id	Infantry
1	1733	145	4379	192	6477	238	9122	284	2189	367	8027
2	2007	146	3466	193	5291	239	5200	285	1277	370	7663
3	1642	147	9122	194	2645	240	3466	286	547	371	3740
4	9031	148	5929	195	365	241	9122	287	7115	389	4743

5	4743	149	547	196	2281	242	5838	289	5656	390	6203
6	9122	150	1733	197	6021	243	4196	290	5017	391	8484
7	2098	151	3558	198	2372	244	6842	291	365	392	1551
8	9122	152	4652	199	4287	245	2281	302	7571	393	2189
9	7480	153	8210	200	7480	246	7480	303	5291	394	7206
10	1551	154	2645	201	2919	247	4287	309	1460	395	2554
11	7298	155	5291	202	1733	248	6750	310	2554	396	6568
12	3466	156	9122	203	5108	249	5017	315	4926	397	5382
13	4652	157	8575	204	2098	250	5200	317	7663	398	8210
14	1733	158	5017	205	1824	251	8119	318	4196	399	5656
15	6203	159	6750	206	2919	252	2007	319	6021	400	7663
16	7115	164	3649	207	6021	253	5656	320	8940	401	7024
17	1368	166	274	208	3466	254	2007	321	6659	402	2645
18	365	167	821	209	7571	255	7024	322	5929	403	7024
19	5200	168	2281	210	4470	256	6750	323	456	404	1916
20	6385	169	2645	211	7936	257	6203	324	5473	405	7936
21	6933	170	2645	212	5564	258	4287	325	4561	406	5564
22	6294	171	3010	213	7754	259	2372	326	1733	407	4014
23	4561	172	6385	214	365	260	1916	327	8210	408	3193
24	4105	137	4105	215	4014	261	5108	328	1460	409	547
99	547	141	6659	216	8210	262	7115	343	5656	410	2919
100	7206	142	3102	217	2828	263	5200	344	3284	411	2828
101	2737	173	2919	218	3102	264	4652	345	5747	412	7115
102	1095	174	8210	219	4561	265	8575	346	4379	413	8848
103	8119	175	5656	220	2828	266	4014	347	5747	414	1003
104	5382	176	1916	221	1733	267	6933	351	5929	415	8757
105	5747	177	8666	222	5838	268	3831	352	5656	416	8027
106	1095	178	5291	223	5929	269	7115	353	4014	417	5200
107	2189	179	1368	224	365	270	3740	354	1916	418	1368
108	9122	180	3010	225	5382	271	3284	355	5929	419	6659
109	7845	181	3284	226	6021	272	8119	356	547	420	912
110	8392	182	5291	227	730	273	547	357	6568	421	4835
114	1824	183	6385	228	8119	274	4470	358	6385	422	182
115	6568	184	4196	229	6477	275	7389	359	639	423	1186
116	2007	185	2737	230	4652	277	5017	360	5747	424	912
119	4105	186	6933	231	7480	278	4287	361	7206	425	1368
126	5382	187	9031	232	547	279	1186	362	4196	426	3010
127	3010	188	4196	233	2463	280	2372	363	1460	427	7389
128	7571	189	2007	234	2189	281	4105	364	8484		
129	6842	190	6021	235	7298	282	1733	365	8757		
136	4014	191	1551	237	7845	283	5929	366	3558		

Table 11 Distribution of the number of self-propelled guns, anti-aircraft guns and tanks on the blue side

Blue id	Guns	Tanks	Blue id	Guns	Tanks	Blue id	Guns	Tanks	Blue id	Guns	Tanks	Blue id	Guns	Tanks
25	141	17	59	62	8	93	119	14	276	68	8	342	120	15
26	50	6	60	27	3	94	84	10	288	108	13	348	99	12
27	82	10	61	79	10	95	106	13	292	88	11	349	135	17
28	87	11	62	5	1	96	64	8	293	146	18	350	12	1
29	53	6	63	120	15	97	108	13	294	126	15	368	113	14
30	147	18	64	97	12	98	102	12	295	61	7	369	147	18
31	76	9	65	82	10	111	102	12	296	125	15	372	78	9
32	135	17	66	21	3	112	120	15	297	140	17	373	99	12
33	76	9	67	43	5	113	140	17	298	125	15	374	119	14
34	109	13	68	147	18	117	108	13	299	50	6	375	140	17
35	58	7	69	79	10	118	113	14	300	100	12	376	108	13
36	135	17	70	76	9	120	132	16	301	17	2	377	96	12
37	114	14	71	44	5	121	119	14	304	43	5	378	147	18
38	132	16	72	50	6	122	30	4	305	15	2	379	12	1
39	38	5	73	141	17	123	65	8	306	35	4	380	30	4
40	143	17	74	125	15	124	12	1	307	21	3	381	116	14
41	5	1	75	128	16	125	27	3	308	29	4	382	82	10
42	120	15	76	141	17	130	2	0	311	103	13	383	138	17
43	24	3	77	117	14	131	117	14	312	17	2	384	75	9
44	12	1	78	109	13	132	24	3	313	76	9	385	149	18
45	131	16	79	3	0	133	90	11	314	11	1	386	35	4
46	84	10	80	58	7	134	151	18	329	70	9	387	109	13
47	79	10	81	143	17	135	2	0	330	146	18	388	94	12
48	23	3	82	79	10	138	12	1	331	91	11	428	149	18
49	11	1	83	58	7	139	75	9	332	40	5	429	56	7
50	32	4	84	90	11	140	14	2	333	119	14	430	140	17
51	53	6	85	12	1	143	70	9	334	140	17	431	99	12
52	88	11	86	65	8	144	73	9	335	62	8	432	14	2
53	128	16	87	116	14	160	138	17	336	143	17	433	81	10
54	144	18	88	85	10	161	143	17	337	103	13	434	20	2
55	84	10	89	23	3	162	44	5	338	143	17	435	8	1
56	137	17	90	30	4	163	113	14	339	111	14	436	35	4
57	123	15	91	144	18	165	120	15	340	91	11			
58	23	3	92	119	14	236	30	4	341	18	2			

Table 12 Distribution of the number of self-propelled guns, anti-aircraft guns and tanks on the red side

Red id	Guns	Tanks	Red id	Guns	Tanks	Red id	Guns	Tanks	Red id	Guns	Tanks	Red id	Guns	Tanks
1	10	1	151	20	3	210	25	3	265	48	6	357	37	5

2	11	1	152	26	3	211	44	6	266	22	3	358	36	5
3	9	1	153	46	6	212	31	4	267	39	5	359	4	0
4	51	7	154	15	2	213	43	6	268	21	3	360	32	4
5	27	3	155	30	4	214	2	0	269	40	5	361	40	5
6	51	7	156	51	7	215	22	3	270	21	3	362	23	3
7	12	2	157	48	6	216	46	6	271	18	2	363	8	1
8	51	7	158	28	4	217	16	2	272	45	6	364	48	6
9	42	5	159	38	5	218	17	2	273	3	0	365	49	6
10	9	1	164	20	3	219	26	3	274	25	3	366	20	3
11	41	5	166	2	0	220	16	2	275	41	5	367	45	6
12	19	2	167	5	1	221	10	1	277	28	4	370	43	6
13	26	3	168	13	2	222	33	4	278	24	3	371	21	3
14	10	1	169	15	2	223	33	4	279	7	1	389	27	3
15	35	4	170	15	2	224	2	0	280	13	2	390	35	4
16	40	5	171	17	2	225	30	4	281	23	3	391	48	6
17	8	1	172	36	5	226	34	4	282	10	1	392	9	1
18	2	0	173	16	2	227	4	1	283	33	4	393	12	2
19	29	4	174	46	6	228	45	6	284	12	2	394	40	5
20	36	5	175	32	4	229	36	5	285	7	1	395	14	2
21	39	5	176	11	1	230	26	3	286	3	0	396	37	5
22	35	5	177	49	6	231	42	5	287	40	5	397	30	4
23	26	3	178	30	4	232	3	0	289	32	4	398	46	6
24	23	3	179	8	1	233	14	2	290	28	4	399	32	4
99	3	0	180	17	2	234	12	2	291	2	0	400	43	6
100	40	5	181	18	2	235	41	5	302	42	5	401	39	5
101	15	2	182	30	4	237	44	6	303	30	4	402	15	2
102	6	1	183	36	5	238	51	7	309	8	1	403	39	5
103	45	6	184	23	3	239	29	4	310	14	2	404	11	1
104	30	4	185	15	2	240	19	2	315	28	4	405	44	6
105	32	4	186	39	5	241	51	7	316	28	4	406	31	4
106	6	1	187	51	7	242	33	4	317	43	6	407	22	3
107	12	2	188	23	3	243	23	3	318	23	3	408	18	2
108	51	7	189	11	1	244	38	5	319	34	4	409	3	0
109	44	6	190	34	4	245	13	2	320	50	6	410	16	2
110	47	6	191	9	1	246	42	5	321	37	5	411	16	2
114	10	1	192	36	5	247	24	3	322	33	4	412	40	5
115	37	5	193	30	4	248	38	5	323	3	0	413	50	6
116	11	1	194	15	2	249	28	4	324	31	4	414	6	1
119	23	3	195	2	0	250	29	4	325	26	3	415	49	6
126	30	4	196	13	2	251	45	6	326	10	1	416	45	6
127	17	2	197	34	4	252	11	1	327	46	6	417	29	4
128	42	5	198	13	2	253	32	4	328	8	1	418	8	1
129	38	5	199	24	3	254	11	1	343	32	4	419	37	5
136	22	3	200	42	5	255	39	5	344	18	2	420	5	1

137	23	3	201	16	2	256	38	5	345	32	4	421	27	3
141	37	5	202	10	1	257	35	4	346	25	3	422	1	0
142	17	2	203	29	4	258	24	3	347	32	4	423	7	1
145	25	3	204	12	2	259	13	2	351	33	4	424	5	1
146	19	2	205	10	1	260	11	1	352	32	4	425	8	1
147	51	7	206	16	2	261	29	4	353	22	3	426	17	2
148	33	4	207	34	4	262	40	5	354	11	1	427	41	5
149	3	0	208	19	2	263	29	4	355	33	4			
150	10	1	209	42	5	264	26	3	356	3	0			

Red's attack roadmap before optimization:

158—>392—>207—>118—>273—>165—>169—>420—>153—>212—>11—>240—>86—>108—>76—>211—>300—>349—>248—>33—>80—>58—>144—>208—>362—>181—>301—>190—>49—>27—>217—>117—>120—>115—>85—>203—>266—>224—>160—>367—>10—>29—>257—>377—>9—>43—>267—>125—>421—>231—>64—>297—>81—>171—>431—>402—>342—>216—>124—>358—>184—>36—>427—>276—>70—>45—>398—>136—>121—>13—>95—>40—>351—>195—>147—>366—>21—>106—>287—>372—>31—>220—>12—>167—>344—>435—>282—>140—>310—>302—>318—>222—>416—>102—>330—>146—>223—>334—>261—>343—>132—>292—>312—>409—>159—>88—>62—>406—>156—>315—>379—>200—>188—>111—>7—>400—>193—>142—>177—>103—>191—>131—>225—>42—>299—>69—>128—>290—>119—>44—>294—>277—>173—>415—>101—>317—>319—>284—>265—>14—>239—>145—>234—>202—>176—>254—>149—>59—>214—>8—>37—>260—>432—>256—>148—>46—>281—>429—>424—>55—>399—>77—>104—>48—>143—>365—>127—>280—>197—>71—>383—>74—>283—>436—>316—>187—>253—>341—>328—>378—>371—>285—>201—>356—>215—>395—>359—>263—>60—>199—>382—>244—>41—>56—>346—>57—>172—>161—>237—>249—>293—>339—>374—>329—>182—>134—>348—>39—>309—>323—>123—>353—>304—>396—>189—>403—>192—>434—>241—>6—>96—>258—>75—>164—>286—>23—>139—>418—>235—>252—>347—>15—>368—>425—>100—>419—>305—>183—>364—>326—>291—>66—>411—>185—>112—>314—>78—>407—>388—>151—>295—>262—>227—>268—>137—>331—>152—>93—>229—>2—>90—>433—>213—>19—>296—>322—>236—>133—>157—>363—>206—>375—>126—>426—>141—>113—>92—>422—>306—>408—>413—>361—>251—>404—>175—>350—>308—>210—>394—>303—>28—>17—>54—>226—>272—>288—>274—>84—>196—>209—>428—>401—>51—>271—>135—>340—>166—>73—>4—>369—>380—>168—>25—>354—>218—>327—>34—>269—>259—>91—>204—>109—>16—>332—>390—>410—>232—>107—>386—>278—>122—>30—>1—>186—>47—>87—>18—>228—>313—>170—>245—>155—>24—>345—>412—>270—>430—>311—>105—>154—>97—>63—>174—>5—>50—>194—>246—>336—>32—>357—>384—>233—>405—>20—>61—>373—>242—>385—>72—>162—>307—>180—>298—>130—>414—>67—>179—>98—>221—>230—>238—>68—>219—>393—>355—>289—>89—>53—>22—>337—>82—>79—>397—>338—>138—>116—>83—>94—>279—>114—>417—>150—>129—>333—>178—>275—>38—>264—>370—>255—>423—>52—>35—>324—>352—>320—>243—>387—>321—>3—>163—>381—>247—>250—>99—>325—>391—>198—>110—>360—>205—>65—>335—>26—>376—>389

Red's optimized attack path diagram:

80—>99—>101—>106—>107—>117—>196—>133—>163—>185—>221—>154—>355—>328—>310—>349—>341—>288—>303—>315—>313—>420—>414—>353—>369—>130—>93—>78—>144—>135—>116—>96—>252—>181—>182—>65—>32—>177—>40—>394—>342—>377—>376—>393—>201—>58—>56—>74—>68—>73—>88—>111—>82—>102—>126—>204—>138—>11—>5—>21—>12—>45—>46—>205—>156—>86—>121—>230—>61—>408—>170—>179—>217—>305—>373—>172—>234—>232—>125—>105—>119—>236—>237—>238—>259—>266—>200—>3—>332—>295—>336—>294—>197—>124—>127—>268—>92—>149—>2—>27—>64—>183—>278—>314—>428—>412—>385—>6—>351—>335—>219—>254—>118—>123—>79—>97—>95—>229—>308—>395—>397—>396—>299—>430—>300—>284—>426—>319—>231—>233—>62—>150—>224—>19—>202—>63—>358—>432—>422—>419—>418—>282—>281—>317—>411—>423—>320—>60—>59—>104—>76—>75—>77—>372—>362—>361—>364—>292—>270—>277—>290—>325—>365—>413—>434—>433—>316—>424—>429—>301—>431—>283—>285—>416—>274—>276—>289—>324—>350—>384—>378—>330—>379—>436—>250—>257—>225—>343—>338—>37—>33—>207—>210—>187—>34—>152—>214—>348—>322—>380—>327—>339—>298—>333—>216—>31—>148—>180—>36—>26—>215—>211—>198—>260—>194—>159—>246—>186—>329—>435—>357—>191—>91—>89—>15—>47—>55—>48—>10—>14—>9—>85—>94—>71—>108—>248—>261—>265—>243—>242—>241—>143—>227—>52—>51—>188—>23—>50—>158—>83—>81—>137—>139—>239—>245—>256—>258—>28—>375—>271—>347—>190—>267—>253—>115—>72—>195—>142—>145—>140—>251—>247—>240—>226—>136—>206—>35—>272—>390—>360—>366—>367—>307—>178—>167—>406—>398—>169—>262—>114—>67—>66—>98—>90—>113—>193—>7—>146—>53—>399—>356—>403—>22—>8—>155—>164—>244—>168—>184—>306—>382—>425—>352—>340—>334—>400—>162—>161—>157—>175—>407—>427—>415—>391—>371—>176—>38—>208—>209—>153—>43—>134—>192—>141—>166—>1—>49—>405—>370—>280—>368—>321—>345—>337—>323—>297—>255—>103—>128—>122—>70—>112—>263—>174—>171—>57—>30—>199—>249—>173—>388—>389—>363—>392—>359—>286—>304—>279—>374—>39—>29—>220—>69—>84—>213—>381—>273—>417—>421—>269—>275—>293—>291—>309—>326—>344—>354—>401—>404—>409—>402—>410—>212—>222—>228—>218—>131—>132—>44—>296—>287—>318—>346—>25—>42—>41—>151—>264—>223—>165—>147—>160—>189—>110—>109—>235—>387—>386—>312—>302—>383—>331—>311—>54—>17—>203—>13—>16—>100—>120—>129—>87—>20—>18—>4—>24

Diagram of the retreat path before optimization on the blue side:

97—>66—>152—>122—>63—>13—>165—>5—>30—>1—>94—>164—>140—>147—>148—>2—>119—>125—>10—>42—>133—>123—>62—>118—>38—>37—>40—>71—>124—>93—>103—>108—>160—>56—>50—>84—>21—>149—>146—>115—>29—>145—>60—>53—>155—>33—>34—>83—>65—>68—>79—>15—>7—>49—>76—>22—>25—>27—>104—>127—>48—>153—>85—>129—>113—>61—>9—>46—>69—>116—>90—>109—>86—>39—>23—>6—>134—>120—>82—>18—>59—>74—>121—>51—>26—>131—>92—>162—>112—>77—>89—>43—>111—>28—>4—>130—>41—>11—>132—>96—>32—>47—>31—>64—>45—>137—>102—>88—>35—>87—>141—>144—>78—>114—>8—>156—>139—>159—>143—>136—>17—>107—>70—>57—>36—>138—>128—>73—>135—>20—>52—>75—>72—>166—>19—>99—>150—>163—>12—>154—>167—>44—>106—>158—>95—>58—>168—>100—>151—>142—>157—>117—>14—>98—>161—>110—>67—>81—>80—>3—>24—>55—>91—>54—>105—>126—>16—>101

Blue's optimized retreat path map:

61—>54—>59—>112—>76—>86—>27—>28—>29—>123—>95—>8—>9—>2—>23—>10—>154—>153—>128—>152—>129—>49—>31—>48—>96—>40—>69—>50—>26—>6—>5—>1—>149—>151—>150—>159—>114—>34—>33—>46—>45—>47—>97—>82—>110—>148—>124—>118—>108—>120—>24—>62—>91—>94—>93—>163—>162—>165—>166—>13—>14—>161—>160—>145—>85—>144—>116—>101—>87—>78—>74—>125—>105—>139—>103—>113—>147—>127—>164—>155—>143—>156—>157—>158—>43—>36—>56—>20—>41—>18—>17—>146—>117—>84—>83—>68—>80—>70—>81—>3—>65—>38—>64—>39—>51—>25—>4—>122—>121—>109—>119—>138—>130—>7—>19—>77—>89—>111—>168—>142—>141—>88—>67—>22—>11—>30—>52—>55—>42—>53—>100—>99—>98—>90—>102—>167—>104—>66—>136—>133—>132—>72—>63—>92—>115—>12—>131—>137—>134—>135—>75—>126—>106—>140—>107—>73—>79—>37—>71—>21—>16—>15—>32—>44—>35—>57—>58—>60

（1）

```
%% 清空环境
clc
clear

%% 算法基本参数
sizepop=100;          % 种群规模
overbest=100;         % 记忆库容量
MAXGEN=100;           % 迭代次数
pcross=0.5;           % 交叉概率
pmutation=0.4;        % 变异概率
ps=0.95;              % 多样性评价参数
length=3;             % 选址数
M=sizepop+overbest;

%% step1 识别抗原,将种群信息定义为一个结构体
individuals = struct(&apos;fitness&apos;,zeros(1,M), &apos;concentration&apos;,zeros(1,M),&apos;excellence&apos;,zeros(1,M),&apos;chrom&apos;,[]);
%% step2 产生初始抗体群
individuals.chrom = popinit(M,length);
trace=[]; %记录每代最个体优适应度和平均适应度

%% 迭代寻优
for iii=1:MAXGEN

   %% step3 抗体群多样性评价
   for i=1:M
      individuals.fitness(i) = fitness(individuals.chrom(i,:));      % 抗体与抗原亲和度(适应度值）计算
      individuals.concentration(i) = concentration(i,M,individuals); % 抗体浓度计算
   end
   % 综合亲和度和浓度评价抗体优秀程度，得出繁殖概率
   individuals.excellence = excellence(individuals,M,ps);

   % 记录当代最佳个体和种群平均适应度
   [best,index] = min(individuals.fitness);   % 找出最优适应度
   bestchrom = individuals.chrom(index,:);    % 找出最优个体
   average = mean(individuals.fitness);       % 计算平均适应度
   trace = [trace;best,average];              % 记录

   %% step4 根据 excellence，形成父代群，更新记忆库（加入精英保留策略，可由 s 控制）
   bestindividuals = bestselect(individuals,M,overbest);   % 更新记忆库
   individuals = bestselect(individuals,M,sizepop);        % 形成父代群
```

```
    %% step5 选择，交叉，变异操作，再加入记忆库中抗体，产生新种群
    individuals = Select(individuals,sizepop);                               % 选择
    individuals.chrom = Cross(pcross,individuals.chrom,sizepop,length);                    %
交叉
    individuals.chrom = Mutation(pmutation,individuals.chrom,sizepop,length);   % 变异
    individuals = incorporate(individuals,sizepop,bestindividuals,overbest);                  %
加入记忆库中抗体

end

%% 画出免疫算法收敛曲线
figure(1)
plot(trace(:,1));
hold on
plot(trace(:,2),&apos;--&apos;);
legend(&apos;最优适应度值&apos;,&apos;平均适应度值&apos;)
title(&apos;免疫算法收敛曲线&apos;,&apos;fontsize&apos;,12)
xlabel(&apos;迭代次数&apos;,&apos;fontsize&apos;,12)
ylabel(&apos;适应度值&apos;,&apos;fontsize&apos;,12)

%% 画出配送中心选址图
%城市坐标
city_coordinate=[84.001925,7.6674;
83.608775,9.302266667;
83.152275,10.4116;
84.798225,9.813633333;
84.119425,9.5218;
83.0695,9.070433333;
82.9367,7.685733333;
82.9601,8.910266667;
82.897875,9.1235;
82.84975,8.980533333;
82.326175,10.06613333;
74.703175,11.6198;
79.506525,0.7677;
79.5526,3.1024;
79.59225,5.6782;
79.593175,5.719533333;
79.62555,7.734833333;
79.627375,7.858366667;
79.163675,10.28883333;
79.681,10.17963333;
81.184325,9.865066667;
82.311575,10.05843333;
```

82.341275,10.07376667;
83.6385,10.6162;
83.875475,10.74283333;
81.539325,13.1841;
79.8001,11.90876667;
80.25245,10.87423333;
81.19695,10.5484;
82.05685,10.52913333;
78.8688,3.1138;
78.8941,5.690033333;
78.895,5.729266667;
78.916575,7.748866667;
78.9291,9.1881;
78.94735,10.13486667;
82.3602,13.36623333;
82.1199,12.01776667;
82.11305,11.7868;
81.182025,8.905;
79.653075,9.1751;
78.886575,10.13566667;
78.884075,9.187733333;
78.862925,7.757066667;
78.84545,5.734266667;
78.84635,5.695966667;
78.83165,4.873233333;
78.8208,3.9436;
78.813975,3.1215;
81.215175,12.02383333;
82.86895,12.011;
78.66115,10.39493333;
78.768325,10.29313333;
78.784175,10.41963333;
78.81575,10.22473333;
78.996725,10.20863333;
79.034,10.2374;
79.0688,10.38363333;
79.009225,10.46346667;
78.836375,10.46946667;
78.80105,10.4413;
78.0067,9.203466667;
77.650425,8.1824;
81.2145,11.8018;
82.885,11.77506667;
77.5086,11.6898;

77.915,11.5492;
78.90755,11.21816667;
78.953475,11.2071;
82.39055,14.45923333;
81.7528,13.61943333;
77.9453,10.5389;
84.99415,15.20203333;
84.156675,15.66553333;
82.3712,15.54373333;
78.915475,11.92993333;
78.891825,10.5552;
80.77115,14.06956667;
83.1716,13.0112;
81.848125,14.85973333;
83.77715,13.63506667;
76.383975,8.560433333;
76.3435,8.445833333;
76.0188,7.512;
75.6848,6.566633333;
78.9641,11.9222;
77.77315,12.4289;
76.38085,13.0342;
75.6854,11.05113333;
75.340375,10.0837;
74.9877,9.0587;
77.601125,8.035833333;
74.934825,8.908533333;
74.082625,9.188766667;
85.88005,7.8604;
81.16255,7.7161;
76.39245,8.5843;
76.756725,9.615333333;
77.106875,10.59543333;
77.1167,10.62643333;
74.972625,9.015533333;
74.68465,11.5419;
88.496875,15.0735;
75.0268,11.443;
86.8214,15.89463333;
86.75735,15.4741;
86.260675,14.15186667;
86.745925,8.617033333;
89.24665,8.002466667;
79.627875,7.888;

74.4701,11.61706667;
78.949925,10.55386667;
86.076675,6.2595;
81.126225,5.693666667;
77.639075,8.150066667;
74.608775,7.9878;
74.26155,7.046066667;
89.2823,7.992133333;
98.82555,5.2493;
85.76505,8.563333333;
85.80445,8.681033333;
85.658975,8.863733333;
85.73835,8.668933333;
86.0796,6.219533333;
85.399925,15.66363333;
85.61915,15.91586667;
81.125775,5.6522;
81.090475,3.100566667;
81.089725,3.0489;
98.8006,6.320566667;
74.470875,14.4144;
75.56255,13.4773;
75.11325,13.63716667;
74.92035,14.64306667;
74.9929,14.63606667;
75.52235,13.3556;
74.95505,14.59876667;
98.922125,5.4714;
87.0632,15.18726667;
86.9773,15.44476667;
74.587725,13.63546667;
74.59625,13.66843333;
74.067625,6.3984;
75.55455,6.179833333;
76.296225,6.070833333;
77.22945,5.933633333;
83.935525,5.604566667;
83.99155,5.603633333;
86.1134,2.665;
83.9439,2.730066667;
83.8943,2.739566667;
81.02785,3.0808;
83.8867,2.684333333;
83.925025,4.111466667;

```
74.0346,4.814966667;
75.49475,4.042833333;
76.261075,3.790233333;
77.230325,3.540033333;
83.96635,4.111366667;
76.284125,5.285033333;
76.27165,4.476233333;
77.228675,4.280233333;
77.22675,5.134366667;
75.5336,5.4157;
78.7876,0.8042;
78.838875,0.800533333;
75.0444,11.47406667;
74.47675,11.63876667;];
carge=[20,90,90,60,70,70,40,90,90,70,60,40,40,40,20,80,90,70,100,50,50,50,80,70,80,40,40,60,70
,50,30];
%找出最近配送点
for i=1:31
    distance(i,:)=dist(city_coordinate(i,:),city_coordinate(bestchrom,:)&apos;);
end
[a,b]=min(distance&apos;);

index=cell(1,length);

for i=1:length
%计算各个派送点的地址
index{i}=find(b==i);
end
figure(2)
title(&apos;最优规划派送路线&apos;)
cargox=city_coordinate(bestchrom,1);
cargoy=city_coordinate(bestchrom,2);
plot(cargox,cargoy,&apos;rs&apos;,&apos;LineWidth&apos;,2,...
    &apos;MarkerEdgeColor&apos;,&apos;r&apos;,...
    &apos;MarkerFaceColor&apos;,&apos;b&apos;,...
    &apos;MarkerSize&apos;,20)
hold on

plot(city_coordinate(:,1),city_coordinate(:,2),&apos;o&apos;,&apos;LineWidth&apos;,2,...
    &apos;MarkerEdgeColor&apos;,&apos;k&apos;,...
    &apos;MarkerFaceColor&apos;,&apos;g&apos;,...
    &apos;MarkerSize&apos;,10)

for i=1:31
```

```
    x=[city_coordinate(i,1),city_coordinate(bestchrom(b(i)),1)];
    y=[city_coordinate(i,2),city_coordinate(bestchrom(b(i)),2)];
    plot(x,y,&apos;c&apos;);hold on
end
```

（2）

```
            if N(1)~=0 & N(2)~=0 & i>=r+2 & i<=r+5 & j>=r+2 & j<=r+5
                V(1,8,kk,1)=V(1,1,kk,1)+X{i,j}(3);
                V(2,8,kk,1)=X{i,j}(3);
            end
        end
    end
    V(3,:,kk,1)=(((V(1,:,kk,1)+V(2,:,kk,1))./k)./t)./e.*0.5;
    V(4,:,kk,1)=(ones(1,8)-exp(-V(3,:,kk,1))).*0.9+0.01*((t-Tf)/(T0-Tf));
    V(6,:,kk,1)=V(4,:,kk,1);
    for ii=2:8
        V(4,ii,kk,1)=V(4,ii,kk,1)+V(4,ii-1,kk,1);
    end
    V(5,:,kk,1)=(V(4,:,kk,1)./max(V(4,:,kk,1))).*((t-Tf)/(T0-Tf)*0.99+0.01*((t-Tf)/(T0-
Tf)));
    f=rand;
    %左上判定
    if f<=V(5,1,kk,1) && f>0
        V(7,1,kk,1)=1;
    end
    %上判定
    if f>=V(5,1,kk,1) && f<=V(5,2,kk,1)
        V(7,2,kk,1)=1;
    end
    %右上判定
    if f>=V(5,2,kk,1) && f<=V(5,3,kk,1)
        V(7,3,kk,1)=1;
    end
    %左判定
    if f>=V(5,3,kk,1) && f<=V(5,4,kk,1)
        V(7,4,kk,1)=1;
    end
    %右判定
    if f>=V(5,4,kk,1) && f<=V(5,5,kk,1)
        V(7,5,kk,1)=1;
    end
    %左下判定
    if f>=V(5,5,kk,1) && f<=V(5,6,kk,1)
```

```
        V(7,6,kk,1)=1;
    end
    %下判定
    if f>=V(5,6,kk,1) && f<=V(5,7,kk,1)
        V(7,7,kk,1)=1;
    end
    %右下判定
    if f>=V(5,7,kk,1) && f<=V(5,8,kk,1)
        V(7,8,kk,1)=1;
    end
end
H=[];
H=HH;
for p=1:num
    h=find(V(7,:,p,1)==1);
    %左上移动
    if h==1
        H(p,:)=H(p,:)-[1 1];
        if H(p,1)==0
            H(p,1)=a;
        end
        if H(p,2)==0
            H(p,2)=b;
        end
        HHH=HH(:,1).*100+HH(:,2);
        zzz=find(HHH==(H(p,1).*100+H(p,2)));
        if  isempty(zzz)
            HH(p,:)=H(p,:);
        end
    end
    %上移动
    if h==2
        H(p,1)=H(p,1)-1;
        if H(p,1)==0
            H(p,1)=a;
        end
        HHH=HH(:,1).*100+HH(:,2);
        zzz=find(HHH==(H(p,1).*100+H(p,2)));
        if  isempty(zzz)
            HH(p,:)=H(p,:);
        end
    end
    %右上移动
    if h==3
```

```
    H(p,:)=H(p,:)+[-1 1];
    if H(p,1)==0
        H(p,1)=a;
    end
    if H(p,2)==b+1
        H(p,2)=1;
    end
    HHH=HH(:,1).*100+HH(:,2);
    zzz=find(HHH==(H(p,1).*100+H(p,2)));
    if  isempty(zzz)
        HH(p,:)=H(p,:);
    end
end
%左移动
if h==4
    H(p,2)=H(p,2)-1;
    if H(p,2)==0
        H(p,2)=b;
    end
    HHH=HH(:,1).*100+HH(:,2);
    zzz=find(HHH==(H(p,1).*100+H(p,2)));
    if  isempty(zzz)
        HH(p,:)=H(p,:);
    end
end
%右移动
if h==5
    H(p,2)=H(p,2)+1;
    if H(p,2)==b+1
        H(p,2)=1;
    end
    HHH=HH(:,1).*100+HH(:,2);
    zzz=find(HHH==(H(p,1).*100+H(p,2)));
    if  isempty(zzz)
        HH(p,:)=H(p,:);
    end
end
%左下移动
if h==6
    H(p,:)=H(p,:)+[1 -1];
    if H(p,1)==a+1
        H(p,1)=1;
    end
    if H(p,2)==0
```

```
            H(p,2)=b;
        end
        HHH=HH(:,1).*100+HH(:,2);
        zzz=find(HHH==(H(p,1).*100+H(p,2)));
        if  isempty(zzz)
            HH(p,:)=H(p,:);
        end
    end
    %下移动
    if h==7
        H(p,1)=H(p,1)+1;
        if H(p,1)==a+1
            H(p,1)=1;
        end
        HHH=HH(:,1).*100+HH(:,2);
        zzz=find(HHH==(H(p,1).*100+H(p,2)));
        if  isempty(zzz)
            HH(p,:)=H(p,:);
        end
    end
    %右下移动
    if h==8
        H(p,:)=H(p,:)+[1 1];
        if H(p,1)==a+1
            H(p,1)=1;
        end
        if H(p,2)==b+1
            H(p,2)=1;
        end
        HHH=HH(:,1).*100+HH(:,2);
        zzz=find(HHH==(H(p,1).*100+H(p,2)));
        if  isempty(zzz)
            HH(p,:)=H(p,:);
        end
    end
end
A=zeros(51,51);
for i=1:num
    A(HH(i,1),HH(i,2))=1;
end
t=t*qqq;
pause(0.01);
set(imh, &apos;cdata&apos;, cat(3,A,A,A) )
set(number,&apos;string&apos;,num2str(t))
```

```
    end
    if (freeze==1)
        run = 0;
        freeze = 0;
    end
    drawnow
end
```

（3）

```
clear
clc
rand(&apos;state&apos;,sum(clock));%初始化随机数发生器
%定义 用户界面
plotbutton=uicontrol(&apos;style&apos;,&apos;pushbutton&apos;,...
&apos;string&apos;,&apos;Run&apos;,...
&apos;fontsize&apos;,12,...
&apos;position&apos;,[200,400,50,20],...
&apos;callback&apos;,&apos;run=1;&apos;);
erasebutton=uicontrol(&apos;style&apos;,&apos;pushbutton&apos;,...
&apos;string&apos;,&apos;Stop&apos;,...
&apos;fontsize&apos;,12,...
&apos;position&apos;,[300,400,50,20],...
&apos;callback&apos;,&apos;freeze=1;&apos;);
number=uicontrol(&apos;style&apos;,&apos;text&apos;,...
&apos;string&apos;,&apos;373.15&apos;,...
&apos;fontsize&apos;,12,...
&apos;position&apos;,[20,400,50,20]);
r=5;%颗粒点作用力范围
zx=26;%中心
md=0.03;%密度
A=zeros(51,51);
[a,b]=size(A);
num=0;%记录颗粒数
T0=100+273.15;%初始温度
t=T0;%初始温度
Tf=23+273.15;%终止温度
qqq=0.996;%温度下降系数
k=1.3806505*10^(-23);
e=6.25*10^18;
for i=1:a
    for j=1:b
        if rand<=md
            A(i,j)=1;
            num=num+1;
```

```
        end
        C{i,j}=[0 0];
    end
end
imh = imshow(cat(3,A,A,A));%在用户界面中显示初始位置画面
axis equal
axis tight
stop= 0;
run = 0;
freeze = 0;
GG=[];
%%
while (stop==0)
    if(run==1)
    HH=[];
    B=zeros(a,b,num,1);%初始化，用于记录每个颗粒为中心的周期性矩阵
    BB=zeros(2*r+1,2*r+1,num,1);%同上，记录 2r-1 近邻
    for i=1:a
        for j=1:b
            if A(i,j)==1
                C{i,j}=[i j];%记录初始颗粒位置
                HH=[HH;i j];
            end
        end
    end
    D=repmat(A,3,3);
    m=1;
    CC=C;
    G=0;
    %初始能量计算，g 为每个颗粒的能量
    for i=1:a
        for j=1:b
            g=0;
            if C{i,j}~=[0 0]
                c=i+51;
                d=j+51;
                %E=D(c-zx+1:c+zx-1,d-zx+1:d+zx-1);
                %B(:,:,m,1)=E;
                EE=D(c-r:c+r,d-r:d+r);
                BB(:,:,m,1)=EE;
                for ii=1:2*r+1
                    for jj=1:2*r+1
                        if BB(ii,jj,m,1)==1 && ii~=r+1 && jj~=r+1
                            g=g+1/((r+1-ii)^6+(r+1-jj)^6);
```

```
            end
          end
        end

        CC{i,j}=[CC{i,j} g];
        m=m+1;
      else
        CC{i,j}=[CC{i,j} 100];
      end
      G=G+g;
    end
  end
  GG=[GG G];
  %提取每个颗粒点为中心区域及提取近邻区域、每个颗粒能量计算
  F=repmat(CC,3,3);
  m=1;
  %提取颗粒点中心 5 近邻范围
  for i=1:a
    for j=1:b
      if CC{i,j}~=[0 0 100]
        c=i+51;
        d=j+51;
        Z=F(c-r:c+r,d-r:d+r);
        FF(:,:,m,1)=Z;
        m=m+1;
      end
    end
  end
  V=zeros(7,8,num,1);%计算过程中的数据冀鲁豫 V 矩阵中，便于调用和查看
  for kk=1:num
    X=FF(:,:,kk,1);
    for i=1:2*r+1.
      for j=1:2*r+1
        N=X{i,j}(1:3);
        %左上
        if N(1)~=0 & N(2)~=0 & i>=1 & i<=r && j>=1 & j<=r
          V(1,1,kk,1)=V(1,1,kk,1)+X{i,j}(3);
          V(2,1,kk,1)=X{i,j}(3);
        end
        %上
        if N(1)~=0 & N(2)~=0 & i>=1 & i<=r & j==r+1
          V(1,2,kk,1)=V(1,1,kk,1)+X{i,j}(3);
          V(2,2,kk,1)=X{i,j}(3);
        end
```

```
        %右上
        if N(1)~=0 & N(2)~=0 & i>=1 & i<=r & j>=r+2 & j<=r+5
           V(1,3,kk,1)=V(1,1,kk,1)+X{i,j}(3);
           V(2,3,kk,1)=X{i,j}(3);
        end
        %左
        if N(1)~=0 & N(2)~=0 & i==r+1 & i>=1 & i<=r
           V(1,4,kk,1)=V(1,1,kk,1)+X{i,j}(3);
           V(2,4,kk,1)=X{i,j}(3);
        end
        %右
        if N(1)~=0 & N(2)~=0 & i==r+1 & i>=r+2 & i<=r+5
           V(1,5,kk,1)=V(1,1,kk,1)+X{i,j}(3);
           V(2,5,kk,1)=X{i,j}(3);
        end
        %左下
        if N(1)~=0 & N(2)~=0 & i>=r+2 & i<=r+5 & j>=1 & j<=r
           V(1,6,kk,1)=V(1,1,kk,1)+X{i,j}(3);
           V(2,6,kk,1)=X{i,j}(3);
        end
        %下
        if N(1)~=0 & N(2)~=0 & i>=r+2 & i<=r+5 & j==r+1
           V(1,7,kk,1)=V(1,1,kk,1)+X{i,j}(3);
           V(2,7,kk,1)=X{i,j}(3);
        end
        %右下
clc;
clear;
close all;
%%
tic
T0=1000;
Tend=1e-3;
L=200;
q=0.9;
X=[84.001925,7.6674;
83.608775,9.302266667;
83.152275,10.4116;
84.798225,9.813633333;
84.119425,9.5218;
83.0695,9.070433333;
82.9367,7.685733333;
82.9601,8.910266667;
82.897875,9.1235;
```

82.84975,8.980533333;
82.326175,10.06613333;
74.703175,11.6198;
79.506525,0.7677;
79.5526,3.1024;
79.59225,5.6782;
79.593175,5.719533333;
79.62555,7.734833333;
79.627375,7.858366667;
79.163675,10.28883333;
79.681,10.17963333;
81.184325,9.865066667;
82.311575,10.05843333;
82.341275,10.07376667;
83.6385,10.6162;
83.875475,10.74283333;
81.539325,13.1841;
79.8001,11.90876667;
80.25245,10.87423333;
81.19695,10.5484;
82.05685,10.52913333;
78.8688,3.1138;
78.8941,5.690033333;
78.895,5.729266667;
78.916575,7.748866667;
78.9291,9.1881;
78.94735,10.13486667;
82.3602,13.36623333;
82.1199,12.01776667;
82.11305,11.7868;
81.182025,8.905;
79.653075,9.1751;
78.886575,10.13566667;
78.884075,9.187733333;
78.862925,7.757066667;
78.84545,5.734266667;
78.84635,5.695966667;
78.83165,4.873233333;
78.8208,3.9436;
78.813975,3.1215;
81.215175,12.02383333;
82.86895,12.011;
78.66115,10.39493333;
78.768325,10.29313333;

78.784175,10.41963333;
78.81575,10.22473333;
78.996725,10.20863333;
79.034,10.2374;
79.0688,10.38363333;
79.009225,10.46346667;
78.836375,10.46946667;
78.80105,10.4413;
78.0067,9.203466667;
77.650425,8.1824;
81.2145,11.8018;
82.885,11.77506667;
77.5086,11.6898;
77.915,11.5492;
78.90755,11.21816667;
78.953475,11.2071;
82.39055,14.45923333;
81.7528,13.61943333;
77.9453,10.5389;
84.99415,15.20203333;
84.156675,15.66553333;
82.3712,15.54373333;
78.915475,11.92993333;
78.891825,10.5552;
80.77115,14.06956667;
83.1716,13.0112;
81.848125,14.85973333;
83.77715,13.63506667;
76.383975,8.560433333;
76.3435,8.445833333;
76.0188,7.512;
75.6848,6.566633333;
78.9641,11.9222;
77.77315,12.4289;
76.38085,13.0342;
75.6854,11.05113333;
75.340375,10.0837;
74.9877,9.0587;
77.601125,8.035833333;
74.934825,8.908533333;
74.082625,9.188766667;
85.88005,7.8604;
81.16255,7.7161;
76.39245,8.5843;

76.756725,9.615333333;
77.106875,10.59543333;
77.1167,10.62643333;
74.972625,9.015533333;
74.68465,11.5419;
88.496875,15.0735;
75.0268,11.443;
86.8214,15.89463333;
86.75735,15.4741;
86.260675,14.15186667;
86.745925,8.617033333;
89.24665,8.002466667;
79.627875,7.888;
74.4701,11.61706667;
78.949925,10.55386667;
86.076675,6.2595;
81.126225,5.693666667;
77.639075,8.150066667;
74.608775,7.9878;
74.26155,7.046066667;
89.2823,7.992133333;
98.82555,5.2493;
85.76505,8.563333333;
85.80445,8.681033333;
85.658975,8.863733333;
85.73835,8.668933333;
86.0796,6.219533333;
85.399925,15.66363333;
85.61915,15.91586667;
81.125775,5.6522;
81.090475,3.100566667;
81.089725,3.0489;
98.8006,6.320566667;
74.470875,14.4144;
75.56255,13.4773;
75.11325,13.63716667;
74.92035,14.64306667;
74.9929,14.63606667;
75.52235,13.3556;
74.95505,14.59876667;
98.922125,5.4714;
87.0632,15.18726667;
86.9773,15.44476667;
74.587725,13.63546667;

```
74.59625,13.66843333;
74.067625,6.3984;
75.55455,6.179833333;
76.296225,6.070833333;
77.22945,5.933633333;
83.935525,5.604566667;
83.99155,5.603633333;
86.1134,2.665;
83.9439,2.730066667;
83.8943,2.739566667;
81.02785,3.0808;
83.8867,2.684333333;
83.925025,4.111466667;
74.0346,4.814966667;
75.49475,4.042833333;
76.261075,3.790233333;
77.230325,3.540033333;
83.96635,4.111366667;
76.284125,5.285033333;
76.27165,4.476233333;
77.228675,4.280233333;
77.22675,5.134366667;
75.5336,5.4157;
78.7876,0.8042;
78.838875,0.800533333;
75.0444,11.47406667;
74.47675,11.63876667;];
%%
D=Distanse(X);
N=size(D,1);
%% 初始解
S1=randperm(N); %随机产生一个初始路线

%% 画出随机解的路径图
DrawPath(S1,X)
pause(0.0001)
%% 输出随机解的路径和总距离
disp(&apos;初始种群中的一个随机值:&apos;)
OutputPath(S1);
Rlength=PathLength(D,S1);
disp([&apos;总距离：&apos;,num2str(Rlength)]);

%% 计算迭代的次数 Time
Time=ceil(double(solve([&apos;1000*(0.9)^x=&apos;,num2str(Tend)])));
```

```
count=0;       %迭代计数
Obj=zeros(Time,1);       %目标值矩阵初始化
track=zeros(Time,N);      %每代的最优路线矩阵初始化
%% 迭代
while T0>Tend
   count=count+1;    %更新迭代次数
   temp=zeros(L,N+1);
   for k=1:L
      %% 产生新解
      S2=NewAnswer(S1);
      %% Metropolis 法则判断是否接受新解
      [S1,R]=Metropolis(S1,S2,D,T0);  %Metropolis 抽样算法
      temp(k,:)=[S1 R];         %记录下一路线的及其路程
   end
   %% 记录每次迭代过程的最优路线
   [d0,index]=min(temp(:,end));
   if count==1 || d0<Obj(count-1)
      Obj(count)=d0;
   else
      Obj(count)=Obj(count-1);
   end
   track(count,:)=temp(index,1:end-1);
   T0=q*T0;
   fprintf(1,&apos;%d\n&apos;,count)  %输出当前迭代次数
end
%% 优化过程迭代图
figure
plot(1:count,Obj)
xlabel(&apos;迭代次数&apos;)
ylabel(&apos;距离&apos;)
title(&apos;优化过程&apos;)

%% 最优解的路径图
DrawPath(track(end,:),X)

%% 输出最优解的路线和总距离
disp(&apos;最优解:&apos;)
S=track(end,:);
p=OutputPath(S);
disp([&apos;总距离：&apos;,num2str(PathLength(D,S))]);
disp(&apos;-------------------------------------------------------------&apos;)
toc
```