Team Control Number

202011211782

Problem Chosen

C

Summary

In this article, we used MATLAB, Excel and other software to establish and solve mathematical models based on graph theory knowledge and 0-1 planning ideas and solved the best strategy for snow removal under different backgrounds.

For question one, we perform cluster analysis based on the distance between two intersections on the surface of the earth to divide all the intersections into groups. Then we used Q-value method to allocate resources for cleaning snow (hereinafter referred to as "resources") to each group. We constructed the Urban transportation network matrix and gave the method of calculating the shortest path between any two points. To allocate task in one group, we introduced the task allocation matrix, abstracted the process of each resource allocation as a Traveling Salesman Problem (TSP), including its solution. Finally we abstracted the problem into a multi-objective 0-1 programming problem. (Goal 1: Making the end time of the last cleaning point earliest, Goal 2: Making the time of every clean-up point obtains resources earliest.) We solved this problem with depth first search by converting multiple targets into single targets and gave both the cleaning scheme and its effect in the condition that total number of different resources are given.

For question two, we adopted the strategy that when one or more sweeping points need to transport snow, the surrounding snow truck will be dispatched. For the scheduling of snow trucks, we abstract it as a complete matching of bipartite graphs problem, and use Hungarian algorithms to solve it. We modified the multi-objective 0-1 planning model established in question one because of the time used with snow truck, and the solution idea remains unchanged.

For question three, to take the impact of vehicles on the sweeping snow on the cleaning speed into consideration, we defined the resistance coefficient, whose value is related to the width of the road and the number of vehicles. According to the obstacle coefficient, we modified the cleaning speed and brought the new speed into the multi-objective 0-1 programming model. By solving the model, the snow cleaning scheme with vehicles on the roadside can be obtained.

For question four, to consider the importance of priority levels in the process of sweeping strategy, we defined the finish ratio and add new goals to the multi-objective 0-1 programming model: Making the average finish ratio smallest, and hierarchical solution is used to solve the new goal first. The two previous targets are still processed in the multi-objective conversion target way, and finally we give the processing effect of the model on the degree of importance.

Key word: **0-1 programming, TSP, The complete matching of bipartite graphs problem, Resistance Coefficient, Finish ratio**

Content

1. Introduction

1.1 Background

Recently, there was heavy snow in the north of China. Large-scale heavy snow has caused a lot of inconvenience to people, and road congestion caused by snow removal delays seriously affected the daily life of residents. For example, many parts of Changchun city were closed for work and school because of heavy snow.

Sanitation workers began trying to clear and transport snow as roads remained clear after the snow fell to ensure normal traffic for residents. A scientific and reasonable road snow removal scheme needs to be given, and the method of snow removal varies with the depth of snow. When the snow depth is small, people can simply sweep the snow in the middle of the road to the green belt or recreation area on the side of the road. However, when the depth of snow increases to the extent that it is difficult to temporarily store it in the green belt, snow sweepers are used to collect snow and dump it in large trucks for transport to large open spaces.

1.2 Work

Annex 1 provides road information for a city, including the latitude and longitude of the junction, the junction connection point, the width of the junction, and the location of potential snow. Try solving the following four problems with mathematical modeling:

(1) Combined with the data in Annex 1, a reasonable cleaning plan is provided on the premise that the snow can be cleared into a nearby green belt or leisure area and the number of cleaning vehicles is fixed.

(2) Provide the best distribution plan for the allocation of snow removal machines, transport vehicles and sanitation workers, and in the case of heavy snow and municipal sanitation facilities snow removal machines, transport vehicles and sanitation facilities relatively fixed, can also complete the task of snow removal as soon as possible.

(3) Temporary parking spaces on both sides of the road are often full of cars, which can hinder snow clearing the road. When the road clears snow, we need to remove vehicles on both sides of the road to facilitate snow clearance, we need to establish a snow clearing schedule, make reasonable arrangements for temporary parking spaces on both sides of the road, and restore the use of temporary parking spaces as soon as possible.

(4) in addition to the city has a flat road there are elevated, intersecting, these will add different difficulties to the snow clearing work, need to be given according to the

road priority (6 is the most important, need to clean first; 1 means the least important, can finally clean up) to deploy a best snow clearing program.

2. Fundamental assumptions

1. Assuming that the sweeper does not have car failures, oil-free and traffic accidents during the process of rushing to the scene.

2. All speeds (including driving speed, cleaning speed, etc.) are uniform, and the speed remains unchanged when the road section does not change.

3. Exposed space exits near the resource station of each group.

4. The open-air area is large enough not to accumulate snow.

5. Don't consider the start-up, turn around, and accelerate the deceleration process when the traffic is stopped, and do not consider traffic problems such as traffic lights.

3. Symbol Description

Symbol	Description
τ_i	When intersection i was cleaned
$allocation_{ij}$	Is resource i allocated to intersection j
$f(i,V)$	The shortest path from point i to the starting point once through the points in the set V
$t_n(i)$	Cleaning time per unit resource required for cleaning point i
$t_w(i)$	The time taken from the start to the completion of cleaning point i
$p_r(i)$	The average value of the priority of the road section with cleaning point i as the end point
$R_p(i)$	The finish ratio of cleaning point i

4. Problem analysis

The formulation of the best snow removal plan for urban roads is essentially a problem of the scheduling of human and material resources, which involves the arrangement of work tasks, the allocation of resources and the formulation of task implementation strategies, which involves the application of graph theory and planning in discrete mathematics and operational research.

4.1 Analysis of question one

Question one requires us to formulate the best urban snow cleaning strategy according to annex I, provided that the snow can be swept to the green belt or other free areas by the roadside and the number of clean cars is fixed.

The number and distribution of intersections in the city are large, and it is difficult to arrange work directly from the global perspective in real life. For this, a grouping strategy can be adopted. Each group is responsible for a certain number of intersections. The grouping is based on the results of clustering analysis based on the distance between two points on the earth's surface.

Because the amount of resources such as clean cars is limited and fixed, resources need to be allocated on the basis of the sum of all potential snow areas within the scope of each group's responsibility. In order to facilitate the establishment of the model and the description of resources, resources need to be normalized.

After completing the allocation of resources in each group, task allocation within the group is required. Since each sweep point requires the allocation of resources, each resource must be allocated to the sweep point, so the two are multi-to-many relationships. The assignment of tasks can be 0-1 programming by building a 0-1 matrix. The overall and partial efficiency of the cleaning should be considered in the setting of the planning goal. The quantitative criteria are the earliest time for the end time of the last clean-up point is the earliest, and the last sweep point that obtains resources has the earliest time. For the path of resource scheduling in each unit, you should go through all the cleaning points that require the resource as shortest and only once as possible, which can be abstracted as a Traveling Salesman Problem.

In summary, after proper processing of the data, a multi-objective 0-1 programming problem for the Traveling Salesman Problem opportunity can be established. Because the dimensional target range is the same, it can be solved by changing multi-objective to single objective.

4.2 Analysis of question two

Compared with the problem one, the second problem increases the condition of "large snow volume" and loses the "green belt that can sweep the snow to the roadside", which requires the snow truck to carry away the snow.

There are many ways to dispatch snow vehicles, including "the allocation of one or more snow carriers per unit, the separation of snow carriers from the cleaning resources, and the transportation of snow in corresponding places when necessary". In this article, we adopt the strategy of "separation of snow trucks from sweeping resources, and transportation snow at corresponding places when necessary". This strategy involves in-depth graph theory knowledge including complete matching of bipartite graphs, which needs to be solved by Hungarian algorithm.

For the multi-objective 0-1 programming model corresponding to task allocation in the group, because it takes time to load snow and wait for snow trucks, the relevant expressions in the model need to be modified, and the solution idea remains unchanged.

In summary, the second problem is to make simple modifications on the basis of problem one and introduce a complete matching model of bipartite graph.

4.3 Analysis of question three

Compared with the previous two questions, the third question increases the obstacle of vehicles to cleaning. For the wider section of the road, because the space is relatively empty, the vehicles on both sides of the road have little impact on the cleaning work, and for the narrow section, the vehicles on both sides of the road have a greater impact on the cleaning work. Based on this rule, it can be defined to describe the degree of impact of vehicles on the cleaning speed - the resistance coefficient, modify the speed of cleaning according to the resistance coefficient, and re-resolving the model according to the new speed.

To sum up, the sweeping speed in question three changes from constant to function, and the rest remains unchanged.

4.4 Analysis of question four

Question four requires an optimal snow removal plan when considering different priorities of different roads. For priority consideration, it can be achieved by introducing new planning goals. For this, it is necessary to define a new objective function and refer to the "response ratio" in operating system process scheduling. This car can define the "completion time ratio". Because of the different dimensions and it is difficult to dimensionless, the new target is solved by hierarchical method, and the original target is still solved by multi-objective to single objective.

In summary, question four adds a planning objective to the previous problem.

5. Model establishment and solution

5.1 Model Establishment and Solving of Problem one

5.1.1 Analysis of the model

First of all, the functional expression of the distance between two points on the earth's surface about the latitude and longitude of the two points is derived, and on the basis of this expression, MATLAB is used for clustering analysis. Based on the results of clustering analysis, all intersections in the city are grouped. Each group of points used to park clean cars and station human and material resources are the closest. Points with average latitude and longitude at all points in this group.

Then formulate a resource allocation strategy. Unitize resources, and each unit's resources will sweep up the same amount of snow at the same time. Based on the sum of snow areas in each group, the Q-value method is adopted to allocate resources.

Then build an urban transportation network and give a calculation method for the shortest path between any two points. Then the first allocation scheme and the secondary allocation scheme of task allocation in each group are given, and the detailed algorithm is given.

Finally, a multi-objective 0-1 programming model is established to solve the model under different parameter values, and the result analysis and sensitivity analysis are given.

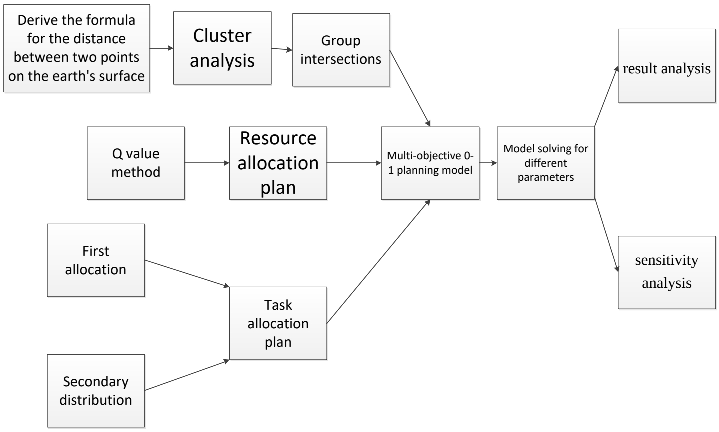

Figure 1. Flowchart for Problem One

5.1.2 Derive the distance between two points on the earth's surface

The earth is regarded as a regular sphere with radius , with the line where the earth axis is located as the axis, the north Arctic direction is the positive of axis, and the center of the earth establishes a spatial right angle coordinate system for the origin . There is an intersection between the prime meridian and axis, the longitude of the Eastern Hemisphere is positive, the western hemisphere longitude is negative, the northern hemisphere latitude is positive, and the southern latitude is negative. Let the latitudes of any two points $A(x_1, y_1, z_1)$ and $B(x_2, y_2, z_2)$ on earth be α_1 and α_2, and longitudes β_1 and β_2 ,respectively, then we can get:

$$\begin{cases} x_1 = R\cos\alpha_1\cos\beta_1 \\ y_1 = R\cos\alpha_1\sin\beta_1 \\ z_1 = R\sin\alpha_1 \end{cases} \qquad \begin{cases} x_1 = R\cos\alpha_2\cos\beta_2 \\ y_1 = R\cos\alpha_2\sin\beta_2 \\ z_1 = R\sin\alpha_2 \end{cases}$$

Because of the distance of point A and B is

$$AB = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Bring the spherical equation $x^2 + y^2 + z^2 = R^2$ into it we can get the equation: $AB = \sqrt{2R^2 - 2(x_1x_2 + y_1y_2 + z_1z_2)}$, Then there is

$$AB = \sqrt{2R^2 - 2R^2\left[\cos\alpha_1\cos\alpha_2\cos(\beta_1 - \beta_2) + \sin\alpha_1\sin\alpha_2\right]}$$

According to the law of cosines $\cos\angle AOB = \cos\alpha_1\cos\alpha_2\cos(\beta_1 - \beta_2) + \sin\alpha_1\sin\alpha_2$, So according to the arc length formula, we can get the distance between two points on the earth's surface is:

$$AB = R \cdot \angle AOB = R\arccos\left[\cos\alpha_1\cos\alpha_2\cos(\beta_1 - \beta_2) + \sin\alpha_1\sin\alpha_2\right]$$

5.1.3 Group intersections

① Perform cluster analysis on the distribution of intersections

Using the distance formula obtained in 5.1.2 as the distance function of the cluster analysis, use MATLAB to perform cluster analysis on the 141 intersections in Annex 1, and draw the cluster tree as shown in Figure 2

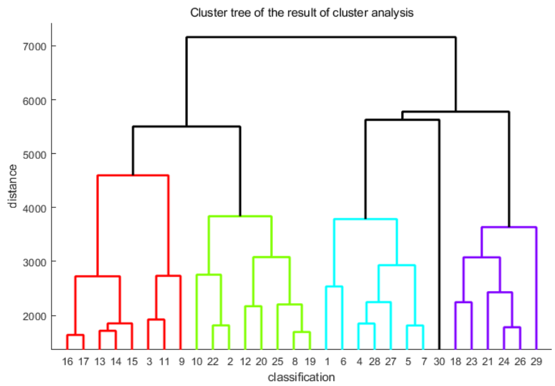

Figure 2. Cluster tree of the result of cluster analysis

It can be seen from Figure 2 that all intersections 2 are divided into 30 categories, and the categories to which each intersection belongs are shown in Table 1

Table 1. each intersection belongs

category	node id	category	node id	category	node id
1	1、2、3	11	45、46、48、49、50	21	95、96、99、100、101
2	34、35、36	12	12、16、21、107、108、109、126、127	22	22、23、24、27、28、29
3	37、38、39、40、41、42、43、44	13	56、63、64、66、67、68	23	98、114、115
4	4、131	14	57、58、59、60	24	110、111、112、113、116、117、118、119、120
5	5、130、132	15	61、62、69、70	25	25、26、30、31、32、33、77
6	6、8、9	16	71、72、82、83	26	121、122、123、124
7	7、125、128、129	17	73、74、80、81、85、86、87	27	133、136、137
8	40、5、76、79	18	84、97	28	134、135
9	41、47、51、52、53、54、55、65	19	88、89、90	29	138、140、141
10	10、11、13、14、15、17、18、19、20	20	91、92、93、94、102、103、104、105、106	30	139

② Determination of grouping

For 141 intersections, 30 classes are a bit too many, and the number of intersections of different classes varies greatly (ranging from 1 to 9), so further processing of the classification is required, and some classes are combined to determine the grouping. The grouping scheme is shown in Table 2.

Table 2. Each group's category

group id	category
1	3 9 11 13 14 15
2	2 8 10 12 19 20 22 25
3	1 4 5 6 7 27 28 30
4	16 17 18 21 23 24 26 29

The distribution of intersections of different groups is shown in Figure 3

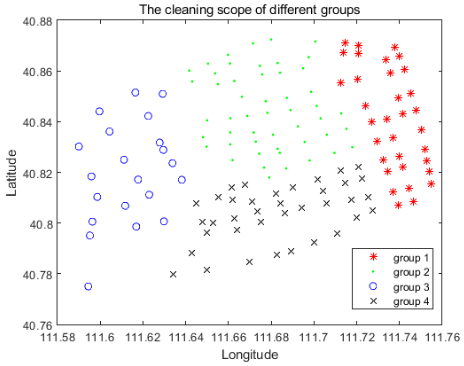

Figure 3. The distribution of intersections of different groups

5.1.4 Treatment of resource units

The human and material resources that can perform snow cleaning work are combined according to a certain standard and defined as a resource unit, and the time for a resource unit to clear the snow with an area of S_0 is defined as a time unit.

5.1.4 Q value method to allocate resources

Assuming that the number of resources is more than the number of groups, refer to the relevant information [1], assuming that the first group has been allocated to the unit of resources, the total snow area to be cleaned by this group is S_i, and the snow area that can be cleaned per unit resource per unit time For S_0, define the Q value:

$$Q = \frac{\left\lceil \frac{S_i}{S_0} \right\rceil^2}{n_i(n_i+1)}$$

Among them, $\lceil x \rceil$ means rounding up x, and assigning resources preferentially to groups with high Q value.

5.1.5 Urban Transportation Network

① Construct a weighted adjacency matrix of urban transportation network

There are 141 intersections in Annex 1. For these intersections, define the matrix

$$City = \{city_{i,j}\}_{141\times 141}$$

$$city_{i,j} = \begin{cases} \infty & \text{node } i \text{ and } j \text{ don't connect directly} \\ dist_{i,j} & \text{node } i \text{ and } j \text{ connect directly} \end{cases}$$

Where $dist_{i,j}$ represents the distance between the point i and point j on the surface of the earth. The city network diagram is shown in Figure4:

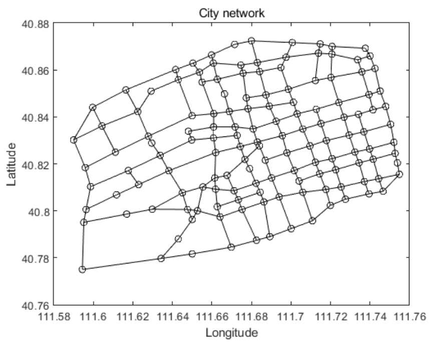

Figure 4. City Network

② The shortest path

Using the Floyd algorithm[1], the length of the shortest path and the corresponding path between any two points in the city network graph can be calculated.

Algorithm 1. Floyd algorithm

Input: City network adjacency matrix, starting point number, end point number

Output: Shortest distance, path

```
function Floyd (G, s, e)
    n=the order of G
    path= n-th order zero matrix
    for k=1:n
        for i=1:n
            for j=1:n
                if G(i,j)>G(I,k)+G(k,j)
                    G(i,j)=G(I,k)+G(k,j)
                    path(i,j)=k
                end if
            end for
        end for
    end for
    dist=G(s,e)
    p=path(s,:)
    for every point in p which value is 0
        this point=s
    end for
    mypath=e
    t=e
    while t not equals s
        q=p(t)
        mypath=[q,mypath]
        t=q
    end while
end function
```

5.1.6 Assignment and planning of tasks within the group

Because there are many uncertain factors in actual work, it is necessary to formulate the first allocation plan and the second allocation plan when formulating the task allocation plan in the group. The first allocation plan does not consider the cleaning work of one point to help the other after the completion of the cleaning work. In the case of points, this situation is handled by the secondary allocation plan. The first allocation is determined and there is a clearly planned route, while the second allocation plan is the reuse of resources and is only a plan.

5.1.7 First allocation plan

① **Construct task allocation matrix**

For intersections to be cleaned and each group resource, construct the following task allocation matrix:

$$Allocation = \{allocation_{ij}\}$$

Among it $allocation_{ij} = \begin{cases} 1 & \text{resource } i \text{ allocated to intersection } j \\ 0 & \text{resource } i \text{ isn't allocated to intersection } j \end{cases}$

② Determination of the best route

When a unit's resources need to be allocated to different intersections for cleaning, it is necessary to find a shortest route that can pass through all points at once. At this time, the problem can be abstracted as a TSP problem.

Let $V_i = \{v_0, v_1, ..., v_{n-1}\}$ is the set of all intersection nodes to be cleaned by the resources of the i -th unit, $e_{m,n}$ stands for the shortest distance between v_m.

A. Establish dynamic programming equation

Let $f(i, V')$ means starting from the i-th vertex, passing through each vertex in vertex set V' once, and finally returning to the shortest path of v_0.

a. when $V' = \varnothing$, $f(i, V') = e_{i,0}$

b. when $V' \neq \varnothing$, At this time, you need to determine a vertex v_k, and then calculate the shortest path starting from v_k, passing through all the points in the set $V' - \{v_k\}$, and finally returning to A. Traverse all the vertices v_k and find the one that makes A the smallest, so that $f(i, V') = \min(f(k, V' - \{v_k\}) + e_{i,k})$

Above all $f(i, V') = \begin{cases} \min(f(k, V' - \{v_k\}) + e_{i,k}), V' \neq \varnothing \\ e_{i,0}, V' = \varnothing \end{cases}$

5.1.8 Secondary distribution plan

When a unit resource has completed the current cleaning work, if there is cleaning work in the range less than the distance from it, consider going to support. The selection criterion is that the selected point is not selected by other unit resources and after reaching the point The point whose completion is less than 80% is the closest.

5.1.9 Establishment of planning model

① Determination of objective function

In the work of snow cleaning, all work should be completed in the shortest time. The quantitative standard is that the maximum use time of each resource in the group is the smallest. Therefore, the following objective function is available:

$$\min \max \tau_i$$

In addition, in order to take into account fairness, try to get all the points to be cleaned to obtain resources as soon as possible, so there is the following objective function

$$\min \max start_i$$

② Determination of constraint conditions

a. Calculation of τ_i

τ_i is determined by the cleaning time and the driving time. For the driving time, assuming a constant speed at speed v_{road}, the driving time is $\frac{f(0, V_i)}{v_{road}}$, and for the cleaning time, assuming the cleaning speed is v_{clean}, the cleaning time is $\frac{\sum_j S_{i,j}}{v_{clean}}$, so that:

$$\tau_i = \frac{f(0,V_i)}{v_{road}} + \frac{\sum_j S_{i,j}}{v_{clean}}$$

b. Constraints on task allocation

Since each cleaning point needs to be allocated resources, so

$$\forall_i, \sum_j allocation_{i,j} \geq m$$

c. Other constraints

Since the value of the elements of matrix $Allocation$ can only be 0 or 1, so

$$allocation_{i,j} \in \{0,1\}$$

③ The establishment of a complete model

Synthesize the objective function and constraints, establish the following multi-objective 0-1 programming model:

$$\min \max \tau_i$$

$$\min \max start_i$$

$$s.t. \begin{cases} \tau_i = \frac{f(0,V_i)}{v_{road}} + \frac{\sum_j S_{i,j}}{v_{clean}} \\ \forall_i, \sum_j allocation_{i,j} \geq m \\ allocation_{i,j} \in \{0,1\} \end{cases}$$

5.1.10 Solving the multi-objective 0-1 programming model

1) Convert multiple goals to single goals

$$\min[\alpha \cdot \max \tau_i + (1-\alpha) \max start_i]$$

Among it $0 \leq \alpha \leq 1$

2) Depth first method to solve the model

Step1: Set the total amount of resources to M unit and set the value of α .

Step2: Use the Q value method to allocate resources to each group

Step3: Use the depth-first method to search for different task assignment combinations in each group. For each determined combination, use the method in 5.1.7② to arrange the cleaning route.

Step4: Find the combination that minimizes the objective function in Step 3 as the final cleaning plan

3) The solution situation under different total resources:

According to the third table in Annex 1, the numbers and snow area of the cleaning points responsible for each group are as shown in Table 3.

Table 3. the cleaning points responsible for each group

group id	Number of potential snow area	Total snow area
1	5，9，12，15，22，24，25，30	4860
2	1，2，7，11，13，14，29	4680
3	4，17，18，28	3780
4	3，6，8，10，16，19，20，21，23，26，27	7450

When $N = 20$

Table 4. The result of Deployment

id	Get resources	route	end time	latest
1	47	Resources can be distributed to all locations at the same time, the amount of resources is as follows 5,7,4,10,4,8,4,5	370	374
2	45	Resources can be distributed to all locations at the same time, the amount of resources is as follows 7,6,4,9,5,6,8	374	
3	36	Resources can be distributed to all locations at the same time, the amount of resources is as follows 7,9,12,8	362	
4	72	Resources can be distributed to all locations at the same time, the amount of resources is as follows 9,5,6,6,7,6,8,6,7,6,6	365	

When $N = 200$

Table 5. The result of Deployment

id	Get resources	route	end time	latest
1	6	5，12，15，24，30 Allocate resources at the same time departure ->22->9->25	216	222
2	5	2，7，13，14 Allocate resources at the same time departure ->1->29->11	222	
3	4	Allocate resources at the same time	208	
4	10	3，6， 10，16，19，20，21，23，26 Allocate resources at the same time， departure ->10->27	195	

4) Result analysis

It can be seen from Table 5 that when the total amount of resources is 200 units, the total amount of resources is sufficient, and all cleaning points can exclusively share one or more units of resources. When the total amount of resources is 20 units, only One group can satisfy the situation that all cleaning points share one resource, and other groups need to reuse resources.

The objective function not only requires all cleaning work to be completed as soon as possible, but also requires all the points to be cleaned to be allocated to resources as early as possible. From the solution results, it can be seen that there is no cleaning point when other cleaning points obtain multiple units of resources. Circumstances where one has not obtained resources.

In order to better reflect the impact of total resources, we make the following image.

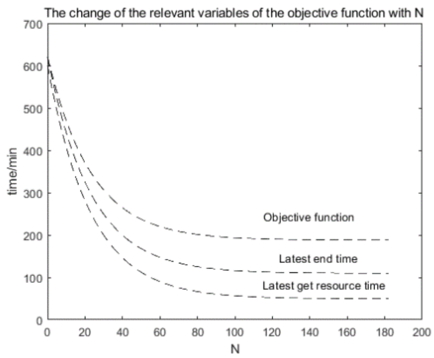

Figure 5. The changes of the relevant Variable value

It can be seen from the figure that at first, as the total amount of resources increased, all the three kind of value decreased significantly, but the rate of decrease became slower and slower, eventually tending to 0.

5) Sensitivity analysis of the model:

In actual work, the vehicle will be affected by various factors during the driving process, and it is difficult to guarantee the speed constant or small-range fluctuations. In order to test the robustness of the model to speed changes, a random increase in the driving speed on each road is adopted. Error resolving the model.

Add random error to the driving speed on each road ς, Suppose the random variable satisfies:

$$\varsigma \sim N(\mu, \sigma^2)$$

For different μ and σ^2, respectively calculate the value of the objective function at the current speed, and make a three-dimensional graph of the absolute value of the marginalization rate of the objective function value under different μ and σ^2

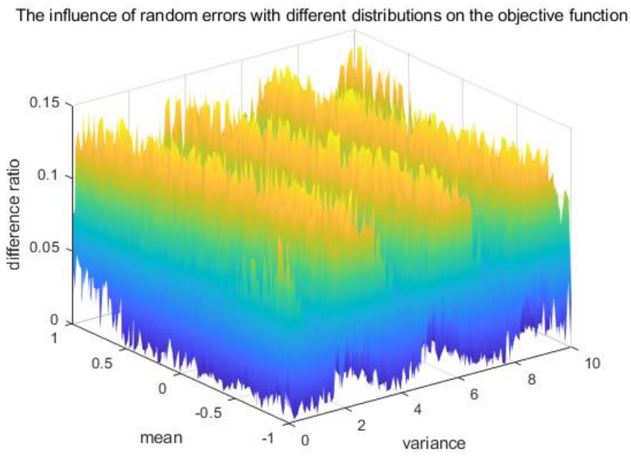

Figure 6. The result of testing

It can be seen from Figure 6 that when the speed has a small random error, the error of the objective function value is small, and the maximum does not exceed 15%. It can be seen that the small range error of the speed has little effect on the objective function.

5.2 Model establishment and solution of problem two

5.2.1 Analysis of the model

For the scheduling of snow trucks, we adopt the strategy of "separating snow trucks from cleaning resources, and go to the corresponding locations to transport snow when necessary". This strategy involves in-depth maps including the complete matching of the bipartite maps. On knowledge, it is necessary to use the Hungarian algorithm to solve.

For the multi-objective 0-1 programming model corresponding to the task distribution in the group, it takes time to wait for the snow truck and snow loading, so the relevant expressions in the model need to be modified, and the solution idea remains unchanged.

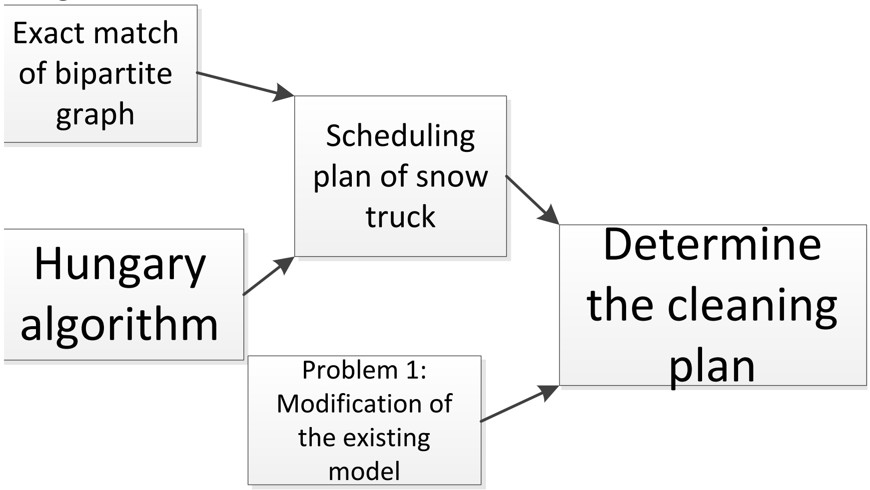

Figure 7. Flowchart for Problem Two

5.2.2 The establishment of the dynamic allocation model of snow trucks (bipartite graph matching model)

Due to the uncertainty of the time required to transport the snow at each cleaning point, the time required for the snow transport vehicle is uncertain, and because the time when the snow transport vehicle is full and the snow fall is uncertain, there are many uncertain factors in the entire snow removal process, which can be used The following scheme: Set up open spaces in the city and outside the city. The number of open areas in the city is 1 in the open area near the station point. There are four open spaces outside the city, which are located at the east, west, and south. The far north. When snow is required to be transported at a certain cleaning point, set the proportion of snow transport vehicle i to the maximum snow transport capacity as R_i, and all snow transport vehicles with R_i less than a certain threshold R can be arranged to transport the snow vehicles in the nearest vicinity. In order to facilitate dispatching and improve efficiency, it is stipulated that snow trucks shall be allocated uniformly only when snow is required to be transported at multiple cleaning points.

Suppose set $V_1(t)$ represents the collection of snow removal points that meet the snow removal conditions at time t after the snow removal work starts, and represents the collection of all cleaning points that can be reached by all snow transport vehicles at time A after the snow removal work starts. In time interval Δt, If a certain cleaning point v satisfies $v \in V_2(t-\Delta t)$, it can be approximately considered that a snow truck has arrived at the cleaning point at time t, and if the point satisfies $v \in V_1(t-\Delta t)$, it is considered that v can transport snow at time t.

In order to prevent the situation where there are free snow trucks but cannot be allocated the required cleaning points, the snow trucks currently assigned to the task will drive to the open field to clean their loaded snow, and then send to someone who is waiting When driving at snow-carrying locations, the selection criteria for locations is to select the cleaning point with the longest waiting time.

Take $A(t) \subseteq V_2(t)$, let the points in $A(t-\Delta t)$ and the edges formed by points in $V_1(t)$ (correspondence between cleaning points and snow trucks) constitute E, and assign snow trucks at time t_0, then the problem can be transformed into the shortest To find the complete matching problem of bipartite graph $G=(A(t-\Delta t)\cup V_1(t), E)$ in time, the mathematical model is as follows:

$$\min t$$
$$s.t.\begin{cases} t > t_0 \\ R_j < R \\ d_{i,j} \le (t-t_0)\cdot v \\ |A(t-\Delta t)| = |V_1(t)| \\ v_i \in V_1(t) \\ v_i \in A(t) \subseteq V_2(t) \end{cases}$$

5.2.3 Hungarian Algorithm[2]

The pseudo code of the Hungarian algorithm is as follows:

Algorithm 2. Hungarian Algorithm

Input: Bipartite graph

Output: Match result

```
function Hungarian (G)
    Initial maximum match is empty
    for Each point on the left half of the bipartite graphi
        do Starting from i, looking for an augmentation path
            if   find
                The current match is successful
            end if
    end for
end function
```

The algorithm for finding the augmentation path in the Hungary algorithm is as follows

Algorithm 3. Find an augmentation path

Input: Bipartite graph G, vertex number k

Output: Is there a Zengguang Road

```
function fun(G, k)
    while List the vertices j that k can be related to from the adjacency list of G
        if   j is not on the way
            Add j to Zengguang Road
            if j is an uncovered point or there is no way to expand from the
               corresponding item of j
                Modify the corresponding item of j to k
                return True
            end if
        end if
    end while
    return False
end function
```

5.2.4 Establishment of multi-objective 0-1 planning model

Compared with the first problem, the second problem has one more snow truck scheduling. During the cleaning process, waiting for the snow truck may take time. Therefore, the expression of τ_i needs to be modified:

$$\tau_i = \frac{f(0,V_i)}{v_{road}} + \frac{\sum_j S_{i,j}}{v_{clean}} + \sum_j T_w(i,j)$$

Where $T_w(i,j)$ represents the sum of the time that the i-th resource waits for the snow truck at the j-th cleaning point and loads the school on the snow truck.

The final multi-objective 0-1 planning model is as follows:

$$\min \max \tau_i$$

$$\min \max start_i$$

$$s.t.\begin{cases} \tau_i = \dfrac{f(0,V_i)}{v_{road}} + \dfrac{\sum_j S_{i,j}}{v_{clean}} + \sum_j T_w(i,j) \\ \forall_i, \sum_j allocation_{i,j} \ge m \\ allocation_{i,j} \in \{0,1\} \end{cases}$$

5.2.4 Model solution

1) Setting of parameters

Table 6. parameter settings

name	value	meaning	explanation
R	75%	Heavy utilization rate of snow transport vehicles	Used to judge whether the snow truck is going to load snow or pour snow
Δt	10	time interval	Used to buffer various time errors
v	40	Driving speed	
D	15	Distance threshold	Used to judge whether the distance meets the requirements

2) Solve

The solution is solved by MATLAB according to the previously determined solution steps. Since the specific resource scheduling route and effect are not easy to display and the relevant results have been analyzed in problem 1, only part of the data matching the snow truck is shown here.

The following table shows the matching of the courier and the delivery point when the number of couriers is 15, and t_0 is equal to a specific value. In order to prevent other factors from affecting, we give a feasible solution within 18 minutes:

Table 7. some information of snow truck use

$t-t_0$ /min	Cleaning point number	Snow truck number	Utilization rate of snow trucks
13	6	8	71%
	4	3	35%

	9	6	20%
15	6	8	71%
	4	3	35%
	9	6	20%
	2	7	48%
18	6	8	71%
	4	3	35%
	9	6	20%
	2	7	48%
	7	5	33%
	8	4	0%
	10	1	0%

It can be seen from Table 7 that as the matching process continues, the number of successfully matched cleaning points is increasing. This is because these cleaning points have begun to have snow transport needs and the number of snow transport vehicles available nearby is increasing. At 18 minutes, the two snow trucks that had just cleaned up the snow were also assigned.

5.3 Model Establishment and Solution of Question Three

5.3.1 Analysis of the model

First of all, the obstacle coefficient is positively correlated with the number of vehicles and negatively correlated with the width of the road. After defining the obstacle coefficient, the clearance speed is modified according to the specific situation of each section, and the modified cleaning speed can be brought into the previously established multi-objective 0-1 planning model to build a new model. The idea of solving the problem remains unchanged.

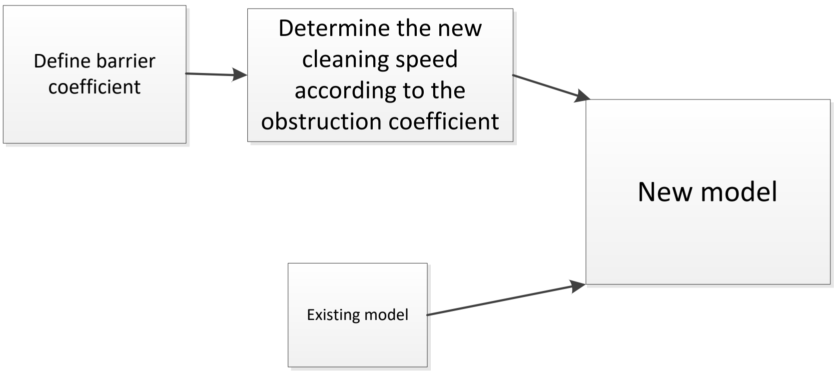

Figure 8. Flowchart for Problem Three

5.3.2 Modeling

①Define the obstacle factor

The obstruction factor should be between 0 and 1 and positively correlated with the number of vehicles and negatively correlated with the width of the road. If the width of roads in j-th clear area is w_j and the number of vehicles parked at $\varphi_j(t)$ at t time is set, the obstruction factor shall be defined as

$$\eta_j(t) = e^{-\frac{w_j}{\varphi_j(t)}}$$

②**modify the clean-up speed**

The modified cleaning speed should be positive and less than the original speed, and it is advisable to take the modified speed $v_{clean}^{'}$ to

$$v_{clean}^{'} = v_{clean}(1 - n_j(t)) .$$

③Example analysis

Suppose $\varphi_j(t) = \lfloor |20\sin(5j+10)| \rfloor$,where $\lfloor x \rfloor$ represents downward consolidation of x

Table 8. gives some relevant data at intersections

id	width	Number of vehicles	Barrier coefficient	Number of resources obtained	Area	time
1	2	13	0.8574	9	870	134
2	4	18	0.8007	8	420	126
3	4	2	0.1353	1	560	115
4	6	19	0.7292	8	540	142
5	6	8	0.4724	4	920	132

It can be seen from table 8 that vehicle obstruction will seriously affect the cleaning speed. In order to overcome the impact caused by vehicles, more resources will be tilted to places with many vehicles. For the five locations in the table, the time difference in cleaning is not large. In terms of resource allocation, the impact of the obstruction coefficient is greater, and the impact on snow cover area is small.

5.4 Model Establishment and Solution of Question Four

5.4.1 Analysis of the model

To consider the importance of different sections of the road, we introduced the completion time ratio and added a new goal based on the previously established 0-1 planning model: the average completion time ratio is the smallest. In the newly established multi-objective 0-1 programming model, the idea of hierarchical solution is used for new targets, and for the two previously existing targets, the multi-objective transfer single target method is used to solve the problem.

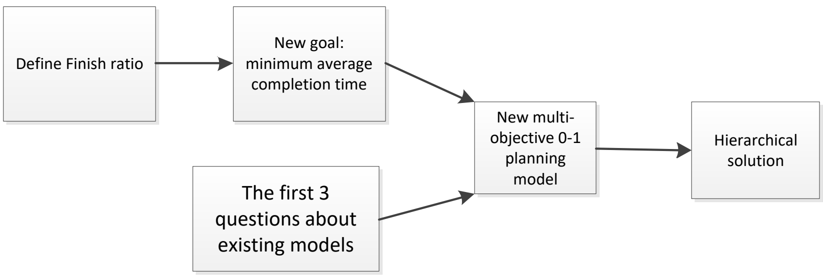

Figure 9. Flowchart for Problem Four

5.4.2 Definition of Finish ratio

For the i to be cleared intersection, the unit of time required to clear the snow in the area is $t_n(i)$, the priority $p_r(i)$ is the average priority of the section with it as the breakpoint, and the time it waits from the cleaning to the completion is $t_w(i)$, define finish ratio as:

$$R_p(i)=\frac{p_r(i)\cdot t_w(i)+t_n(i)}{t_n(i)}$$

5.4.3 Establishment of a multi-objective planning model

Add the goal based on the objectives of the previous problem: the average completion ratio is the smallest, that is:

$$\min\frac{1}{N}\sum_{i=1}^{N}R_p(i)$$

The multi-objective programming model can be organized into:

$$\min\max\tau_i$$
$$\min\max start_i$$
$$\min\frac{1}{N}\sum_{i=1}^{N}R_p(i)$$

$$s.t.\begin{cases}\tau_i=\dfrac{f(0,V_i)}{v_{road}}+\dfrac{\sum_j S_{i,j}}{v_{clean}}+\sum_j T_w(i,j)\\ \forall_i,\sum_j allocation_{i,j}\ge m\\ allocation_{i,j}\in\{0,1\}\end{cases}$$

5.4.4 Solution of Multi-objective Planning Model

1) solve ideas

Set a threshold λ, take $\min\frac{1}{N}\sum_{i=1}^{N}R_p(i)$ as the primary optimization goal, find out all solutions that meet $\frac{1}{N}\sum_{i=1}^{N}R_p(i)\le\lambda$, complete the goal, take the optimization results as the constraint condition, and then use $\min[\alpha\cdot\max\tau_i+(1-\alpha)\max start_i]$ as the optimization goal, and solve the results through MATLAB to obtain the results.

2) solution steps

Step1: ignore other goals, consider only the goal of $\min \frac{1}{N}\sum_{i=1}^{N} R_p(i)$, and use depth first to search to solve $\min \frac{1}{N}\sum_{i=1}^{N} R_p(i) = \lambda_0$.

Step2: set threshold $\lambda = 1.2\lambda_0$.

Step3: use depth-first search to untie all combinations that make $\frac{1}{N}\sum_{i=1}^{N} R_p(i) \le \lambda_0$, forming a set Ω.

Step4: traverse all the elements in Ω to find the ultimate solution that minimizes $\alpha \cdot \max \tau_i + (1-\alpha)\max start_i$.

3) Results presentation and analysis

Take $N = 200$ as an example, we calculate and consider the relevant index values before and after the new objective function, as shown in Table 9:

Table 9. effect

name	Before considering new goals	After considering the new goal	Rate of change
Latest end time	222	245	↑10.36%
Latest get resource time	185	201	↑8.65%
average finish ratio	17.67	8.4	↓52.46%

It can be seen from table 9 that after adding the objective function, although the late end time and the latest resource acquisition time have increased, the increase is not significant, and the average completion time ratio is greatly reduced, which shows the rationality of the model.

6. Model evaluation, improvement and promotion

6.1 Advantages and disadvantages of the model

6.1.1 Advantages of the Model

This paper abstracts the problem based on graph theory knowledge, and also establishes a mathematical model based on 0-1 programming. The connection and difference between each question are considered in modeling. The following questions are always extended from the basis of previous problem modeling. The methods adopted in this modeling are in line with the reality.

For problem one, in order to determine the allocation scheme of tasks in the group, we established a 0-1 matrix and abstracted a layer of planning model. In order to optimize the distribution of tasks within the group, we extracted the traveling Salesman Problem.

For problem two, we note that the difference between it and problem one is that increasing the snow cover requires transportation restrictions, so we built a model that perfectly matches the bipartite graph, and then modified the existing model of problem one to solve the problem.

For problem three, we note that a clear difference between it and the previous two questions is that the presence of roadside vehicles affects cleaning, and we equate the vehicle's impact on cleaning to the vehicle's impact on cleaning speed.

For problem four, in order to consider the priorities of each path in the formulation of the strategy, a response ratio has been introduced, and a new objective function has been added, that is, priority is considered, and the overall cleaning effect is taken into account.

In a word, our model will skillfully use the knowledge of graph theory and operations research to solve complex problems in a simple way, and we will flexibly adopt new strategies for new demands in the problem.

6.1.2 Disadvantages of the model

The model algorithm used in this paper is highly complex, and the solution results cannot be obtained in the short term when the data volume is too large.

This paper ignores the possibility of influencing various unexpected circumstances and environmental factors in modeling, and the model is a little ideal.

6.2 Model Improvement and Extension

6.2.1 Model Improvement

In order to reduce the complexity of the algorithm, you can take multiple data simulations to determine which steps have little impact on the objective function. You can consider canceling these steps to reduce the complexity of the algorithm.

For the occurrence of accidents, a probability model can be established, and the response strategy of the accident can be given according to the probability model.

6.2.2 Extension of the Model

The model of "Best Snow removal Plan for Urban Roads" established in this paper is universal and can be extended to scheduling problems in other fields, such as courier scheduling problems, bus route arrangement, etc. The model can make reasonable use of known relevant data and give a reasonable scheme.

References

[1] Shoukui Si，Zhaolaing Sun，Mathematical Modeling Algorithms and Applications，

Beijing：national defence industrial press,2019
[2] J.Munkres，Algorithms for the Assignment and Transportation Problems,Journal of the Society for Industrial and Applied Mathematics,5(1):32-38,1957

Appendix

Appendix 1 MATLAB code and itsexamination
The examination of MATLAB code：
The attachment in the C question needs to be placed in the current folder of MATLAB, otherwise the program cannot run。
!!!!!!
If an error is reported when a program is running, it is likely that the program needs to use the execution results of other programs or the attachments given by the problem are not placed in the current MATLAB folder, which is not that the program really cannot run.!!!
!!!!!!
funtion file:
dist.m
The function corresponding to the distance formula between two points on the surface of the earth
Some other functions are just some encapsulation freely, there is no need to introduce
script file:
Q1_1.m
Perform cluster analysis and write the data that will be used later in the file
See the generated file "T.csv" for the class to which each node belongs
Q1_2.m、picture.m、Q1_5.m、Q1.m、drawcity.m
draw a picture
Q1_3.m
a template tool
Q1_res.m
The data source of the last two tables of question one
Q2res.m
The data source of the last table of question 2
Q3_RES.m
The data source of the last table of question 3
Q4_RES.m
The data source of the last table of question 4

dist.m

```
function [dis] = dist(P1,P2)
R=6371000;
%R=1;
jd1=P1(1);
wd1=P1(2);
jd2=P2(1);
wd2=P2(2);
dis=R*acos(cosd(wd1)*cosd(wd2)*cosd(jd1-jd2)+sind(wd1)*sind(wd2));
```

```
end

distfun.m
function [ret_dist] = distfun(X,Y,varargin)
[xrow,xcolumn]=size(X);
[yrow,ycolumn]=size(Y);
nVarargs = length(varargin);
ret_dist=zeros(yrow,1);
if (xrow==1 && xcolumn==ycolumn)
    for m=1:yrow
        ret_dist(m,1)=dist(X,Y(m,:));
    end
end
end

Q1_1.m
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
Y=pdist(X,@distfun);
SF=squareform(Y);
Z=linkage(Y,'average');
[H,T] =dendrogram(Z,'colorthreshold','default');
xlswrite('T.csv',T);
numCluster=numel(H);
set(H,'LineWidth',2);

Q1_2.m
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
T=xlsread('T.csv',1);
group=[3 9 11 13 14 15 -1 -1;
    2 8 10 12 19 20 22 25;
    1 4 5 6 7 27 28 30;
    18 21 23 24 26 29 16 17
    ];
g1=[];
g2=[];
g3=[];
g4=[];
for i=1:141
```

```
    if isin(T(i),group(1,:))
        g1=[g1;X(i,:)];
        continue
    elseif isin(T(i),group(2,:))
        g2=[g2;X(i,:)];
        continue
    elseif isin(T(i),group(3,:))
        g3=[g3;X(i,:)];
        continue
    else
        g4=[g4;X(i,:)];
        continue
    end
end
plot(g1(:,1),g1(:,2),'r*');hold on;
plot(g2(:,1),g2(:,2),'g.');hold on;
plot(g3(:,1),g3(:,2),'bo');hold on;
plot(g4(:,1),g4(:,2),'kx');hold on;
```

drawcity.m

```
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
R=xlsread('Attachment 1.xlsx',2,'A2:B242');
for i =1:241
    sp=X(R(i,1),:);
    ep=X(R(i,2),:);
    plot([sp(1) ep(1)],[sp(2) ep(2)],'ko-');
    hold on
end
```

Q1_res.m

```
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
T=xlsread('T.csv',1);
group=[3 9 11 13 14 15 -1 -1;
    2 8 10 12 19 20 22 25;
    1 4 5 6 7 27 28 30;
    18 21 23 24 26 29 16 17
    ];
```

```
g1=[];
g2=[];
g3=[];
g4=[];
for i=1:141
    if isin(T(i),group(1,:))
        g1=[g1;X(i,:)];
        continue
    elseif isin(T(i),group(2,:))
        g2=[g2;X(i,:)];
        continue
    elseif isin(T(i),group(3,:))
        g3=[g3;X(i,:)];
        continue
    else
        g4=[g4;X(i,:)];
        continue
    end
end
dx=0.1/1000;
k=[0.082,0.37,0.045,0.028];
c=[1377,2100,1726,1005];
rou=[300,862,74.2,1.18];
thick=[0.6,6,3.6,5]/1000;
%lamda=k./(c.*rou)*dt/(dx*dx);
n=round(sum(thick/dx)+1);
MAIN=zeros(n,n);
b=zeros(1,n);
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5]
t2=[7,6,4,9,5,6,8]
t3=[7,9,12,8]
t4=[9,5,6,6,7,6,8,6,7,6,6]
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5]
t2=[7,6,4,9,5,6,8]
t3=[7,9,12,8]
t4=[9,5,6,6,7,6,8,6,7,6,6]
stt=zeros(1,1645);
mins=inf;
lke=0;
```

Q2res.m

```
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
T=xlsread('T.csv',1);
group=[3 9 11 13 14 15 -1 -1;
    2 8 10 12 19 20 22 25;
    1 4 5 6 7 27 28 30;
    18 21 23 24 26 29 16 17
    ];
g1=[];
g2=[];
g3=[];
g4=[];
for i=1:141
    if isin(T(i),group(1,:))
        g1=[g1;X(i,:)];
        continue
    elseif isin(T(i),group(2,:))
        g2=[g2;X(i,:)];
        continue
    elseif isin(T(i),group(3,:))
        g3=[g3;X(i,:)];
        continue
    else
        g4=[g4;X(i,:)];
        continue
    end
end
dx=0.1/1000;
k=[0.082,0.37,0.045,0.028];
c=[1377,2100,1726,1005];
rou=[300,862,74.2,1.18];
thick=[0.6,6,3.6,5]/1000;
%lamda=k./(c.*rou)*dt/(dx*dx);
n=round(sum(thick/dx)+1);
MAIN=zeros(n,n);
b=zeros(1,n);
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
```

```
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
A=[13   6    8    71
      13 4      3    35%
      13 9      6    20%
       15  6    8    71%
      15 4      3    35%
      15 9      6    20%
      15 2      7    48%
18   6    8    71%
      18 4      3    35%
      18 9      6    20%
      18 2      7    48%
      18 7      5    33%
      18 8      4    0%
      18 10     1    0%
]
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
```

Q3.RES.m

```
clear
clc
```

```
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
T=xlsread('T.csv',1);
group=[3 9 11 13 14 15 -1 -1;
    2 8 10 12 19 20 22 25;
    1 4 5 6 7 27 28 30;
    18 21 23 24 26 29 16 17
    ];
g1=[];
g2=[];
g3=[];
g4=[];
for i=1:141
    if isin(T(i),group(1,:))
        g1=[g1;X(i,:)];
        continue
    elseif isin(T(i),group(2,:))
        g2=[g2;X(i,:)];
        continue
    elseif isin(T(i),group(3,:))
        g3=[g3;X(i,:)];
        continue
    else
        g4=[g4;X(i,:)];
        continue
    end
end
dx=0.1/1000;
k=[0.082,0.37,0.045,0.028];
c=[1377,2100,1726,1005];
rou=[300,862,74.2,1.18];
thick=[0.6,6,3.6,5]/1000;
%lamda=k./(c.*rou)*dt/(dx*dx);
n=round(sum(thick/dx)+1);
MAIN=zeros(n,n);
b=zeros(1,n);
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
```

```
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
ANS=[1 2     13   0.8574   9     870 134
2     4     18   0.8007   8     420 126
3     4     2     0.1353   1     560 115
4     6     19   0.7292   8     540 142
5     6     8     0.4724   4     920 132
]
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
```

Q4_RES.m

```
clear
clc
close all
X=xlsread('Attachment 1.xlsx',1,'B2:C142');
T=xlsread('T.csv',1);
group=[3 9 11 13 14 15 -1 -1;
      2 8 10 12 19 20 22 25;
      1 4 5 6 7 27 28 30;
      18 21 23 24 26 29 16 17
      ];
g1=[];
g2=[];
g3=[];
g4=[];
```

```
for i=1:141
        if isin(T(i),group(1,:))
                g1=[g1;X(i,:)];
                continue
        elseif isin(T(i),group(2,:))
                g2=[g2;X(i,:)];
                continue
        elseif isin(T(i),group(3,:))
                g3=[g3;X(i,:)];
                continue
        else
                g4=[g4;X(i,:)];
                continue
        end
end
dx=0.1/1000;
k=[0.082,0.37,0.045,0.028];
c=[1377,2100,1726,1005];
rou=[300,862,74.2,1.18];
thick=[0.6,6,3.6,5]/1000;
%lamda=k./(c.*rou)*dt/(dx*dx);
n=round(sum(thick/dx)+1);
MAIN=zeros(n,n);
b=zeros(1,n);
last=ones(1,n)*37;
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
ANS=[222   245 10.36
185 201 8.65
17.67     8.4 -52.46
]
last=ones(1,n)*37;
```

```
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
ue=75;
ls1=[370 374 362 365];
ls2=[216 222 208 195];
t1=[5,7,4,10,4,8,4,5];
t2=[7,6,4,9,5,6,8];
t3=[7,9,12,8];
t4=[9,5,6,6,7,6,8,6,7,6,6];
stt=zeros(1,1645);
mins=inf;
lke=0;
```

Q1_3.m

```
A=[111.675804   40.858645
111.682991  40.829283
111.645046  40.807662
111.616444  40.851497
111.720216  40.85679
111.655395  40.810174
111.668761  40.842274
111.686296  40.803758
111.734158  40.841073
111.705987  40.807881
111.655035  40.854799
111.745081  40.851006
111.679972  40.872394
111.687302  40.849369
111.712311  40.855481
111.671205  40.808536
111.598622  40.810283
111.604371  40.836106
111.634267  40.779697
111.701675  40.817163
111.700094  40.792398
111.734446  40.820439
111.690177  40.814024
111.750399  40.836816
111.740051  40.865957
111.650077  40.796303
111.725391  40.810174
111.611701  40.806788
```

```
111.674367  40.827345
111.737176  40.812467
];
gdv=zeros(30,1);
for i=1:30
    gdv(i)=getgroup(A(i,:),X);
end
```

Q1_5.m

```
dx=0.1/1000;
k=[0.082,0.37,0.045,0.028];
c=[1377,2100,1726,1005];
rou=[300,862,74.2,1.18];
thick=[0.6,6,3.6,5]/1000;
lamda=k./(c.*rou)*dt/(dx*dx);
n=round(sum(thick/dx)+1);
MAIN=zeros(n,n);
b=zeros(1,n);
last=ones(1,n)*37;
ue=75;
stt=zeros(1,1645);
mins=inf;
lke=0;
ke=11262;
%t=zeros(2,10000);
```

chase.m

```
function x = chase(A,f)
n=length(f);
a=zeros(1,n);
b=zeros(1,n);
c=zeros(1,n);
a(1)=0;
a(n)=A(n,n-1);
b(n)=A(n,n);
b(1)=A(1,1);
c(1)=A(1,2);
c(n)=0;
for i=2:n-1
    b(i)=A(i,i);
    a(i)=A(i,i-1);
    c(i)=A(i,i+1);
end
x=zeros(1,n);
```

```
y=zeros(1,n);
d=zeros(1,n);
u=zeros(1,n);
d(1)=b(1);
for i=1:n-1
    u(i)=c(i)/d(i);
    d(i+1)=b(i+1)-a(i+1)*u(i);
end
y(1)=f(1)/d(1);
for i=2:n
    y(i)=(f(i)-a(i)*y(i-1))/d(i);
end
x(n)=y(n);
for i=n-1:-1:1
    x(i)=y(i)-u(i)*x(i+1);
end
```

isin.m

```
function [pd] = isin(sj,jh)
[m,n]=size(jh);
for i=1:m
    for j=1:n
        if sj==jh(i,j)
            pd=1;
            return ;
        end
    end
end
pd=0;
end
```