PROJECT ONE

Police Station Assignment Problem

何钰佳 1530005008
彭博雅 1530005027
熊　峰 1530005039
唐一菡 1530012043
王思祎 1530013026

Date: May 20, 2018

Content

Abstract

Police has a strong power to solve the traffic services and protect the safety of city. So it is necessary to utilize their capacity which is about setting some police service stations to benefit people. However, the resource of police is limit, we need to set some station points and arrange the corresponding area to maximize their power according to the real situation of this city and the demand nearby. In this project, we are aim to solve this kind of project.

We want to solve the problem to assign the 92 areas to 20 police stations. In question 1, we only considered the distance between every points. With use of Floyd Algorithm, we find all the shortest paths between every points. As well as considering the time cost for each police station to get to the criminal positions should be within 3 minutes. After dividing speed, we can get the time cost. Therefore, we consider minimum distance for all points to 20 police stations. According to the principle of assignment we know that each zone only in the charge of one policy station and each station is in charge of at least one zones, as a result we build location set covering model to simplify the question and use the 0,1 variable to reach the minimum time cost for all 92 points by using MATLAB.

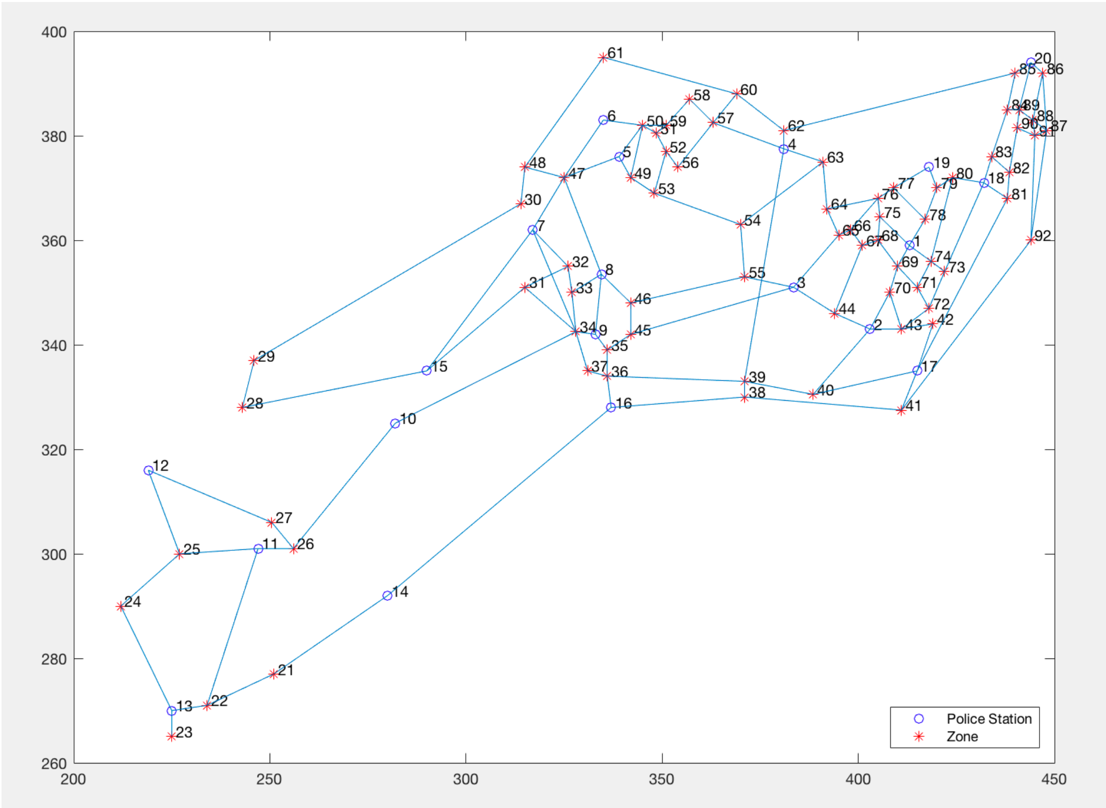
Since there are six zones that are not satisfy the requirement of reaching its police station within three minutes, the additional stations that we need build should consider those six zones first to guarantee the 'quick reach' condition. Then, we find that there still exist three blind points which means that two new stations are not enough.

Last but not the least; we consider 'duty balance' as another criterion which is measured as workload variance of each blind point. The less workload variance it gets when blind point transfer into police station, the more likely the blind point position should be considered as new police station location. And finally, we find that when four new stations built, it will reach the approximately optimal solution.

- ## Notation statement:

i	Represent the number of policy station.
j	Represent the number of zones that are in the charged of policy station.
Xij	When Xij=1, station I takes charge of intersection J; When Xij=0, when station I doesn't take charge of intersection J.
Lij	The shortest distance between station I and zone J.
rj	Represent the crime accident frequency: times per day.

Modelling



1.Restate problem

In order to implement the functions of police, it is necessary to set up patrol service platforms in some important areas. Assume the functions and capacities of each patrol service platform are equal. Since the policing resources are limited, how to properly set up the police service stations according to the actual situation and needs of the city, assign the charged area for each platform, and dispatch policing resources are a real hot issue faced to the policing department. This topic is connected with this problem, and aim to find some better solutions.

Question 1:

According to the excel data and graph, we should assign 92 jurisdiction areas to 20 police stations to make sure if there is crime accident happened, the police can arrive incident area within 3 minutes with the speed of 60 kilometers per hour. If cannot arrived in 3 minutes, find the better assignment can let the rescue time as small as possible.

Question 2:

Due to the unbalance of work for each police station and some palaces cannot be support in a short time, we want to build another two station to make sure each area can be quick reach' and 'duty balance' for police platform.

2.Modeling Building

Question 1

According to given appendix excel of path and traffic network, we can know which two points have the path information. If there is a direct path, then we calculate the distance between them by using the location to calculate the distance. Therefore, we get a distance matrix. They are recorded as initial adjacency matrix and distance matrix. Then we can use Floyd Algorithm to find the nearest station point for each intersection in MATLAB which is can be arrived within 3 minutes (all of MATLAB code are in appendix).

Floyd Algorithm:

Step 1: Use the position for each points and the path relation between them.

Therefore we get initial A(0)=[a_ij]92*92 { if there no path a_ij is infinite denote INF} and P(0) for path matrix.

A(0) matrix:{92*92} :initial path distance

0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	1.15974135	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	1.15974135	0	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

for the dimension is very large, we only show the first 12 lines

P(0) matrix: initial path matrix

0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	9	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	8	0	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

for the dimension is very large, we only show the first 12 lines

Step 2: if a_uv(j-1)<=a_uj(j-1)+ajv(j-1) then we assign a_uv(j)= a_uv(j-1) and P_uv(j)=P_uv(j-1) otherwise we assign a_uv(j)= a_uj(j-1)+ajv(j-1). Therefore we find all the shortest path matrix A(n) and path matrix P(n) for each points.

Step 3: we use the first 20 points as the tree solution for station, then other assign 72 areas into 20 police stations

A(n) matrix: shortest path distance between each vertex

0	1.89874901	3.88388424	4.53521672	9.37428888	9.53751781	11.5003493	9.02262469	9.22543783	14.6495675	19.0879287	22.2361527
1.89874901	0	2.11165363	5.68506755	7.83371133	9.84207712	9.72811872	7.25039408	7.45320722	12.8773369	17.3156981	20.4639221
3.88388424	2.11165363	0	4.04338518	5.7220577	7.73042349	7.61646509	5.13874045	5.34155358	10.7656833	15.2040445	18.3522685
4.53521672	5.68506755	4.04338518	0	4.92004352	5.00230109	7.65669035	8.32728285	8.98666812	14.4107978	18.849159	21.997383
9.37428888	7.83371133	5.7220577	4.92004352	0	2.94262885	2.73664683	3.53568541	4.69542676	10.0416059	14.4799671	17.6281911
9.53751781	9.84207712	7.73042349	5.00230109	2.94262885	0	2.76723172	3.56627031	4.72601166	10.0721908	14.510552	17.658776
11.5003493	9.72811872	7.61646509	7.65669035	2.73664683	2.76723172	0	2.47772464	2.90920846	7.32835058	11.7667118	14.9149358
9.02262469	7.25039408	5.13874045	8.32728285	3.53568541	3.56627031	2.47772464	0	1.15974135	6.50592047	10.9442817	14.0925057
9.22543783	7.45320722	5.34155358	8.98666812	4.69542676	4.72601166	2.90920846	1.15974135	0	5.42412969	9.86249089	13.0107149
14.6495675	12.8773369	10.7656833	14.4107978	10.0416059	10.0721908	7.32835058	6.50592047	5.42412969	0	4.4383612	7.58658521
19.0879287	17.3156981	15.2040445	18.849159	14.4799671	14.510552	11.7667118	10.9442817	9.86249089	4.4383612	0	3.79135282
22.2361527	20.4639221	18.3522685	21.997383	17.6281911	17.658776	14.9149358	14.0925057	13.0107149	7.58658521	3.79135282	0

for the dimension is very large, we only show the first 12 lines

P(n) matrix: shortest path from every vertex to other vertex

1	69	69	75	75	75	69	69	69	69	69	69
70	2	44	44	44	44	44	44	44	44	44	44
65	44	3	65	55	55	55	55	45	45	45	45
63	63	63	4	57	57	57	63	39	39	39	39
49	49	49	50	5	47	47	47	47	47	47	47
59	59	59	59	47	6	47	47	47	47	47	47
32	32	32	47	47	47	7	32	32	32	32	32
46	46	46	46	47	47	33	8	9	33	33	33
35	35	35	35	8	8	34	8	9	34	34	34
34	34	34	34	34	34	34	34	34	10	26	26
26	26	26	26	26	26	26	26	26	26	11	25
27	27	27	27	27	27	27	27	27	27	25	12

for the dimension is very large, we only show the first 12 lines

0-1 Location Set Covering Method:

We want to assign these zone to the closest policy station as a result the total distance is shortest and we transform this shortest path problem into Location Set Covering problem which is 0-1 covering problem, the function of 0-1Location Set Covering is:

$$X_{ij}=\begin{cases} 1 & \text{when i station is in charge of j zone} \\ 0 & \text{when i station is not in charge of j zone} \end{cases}$$

$$\min\ S=\sum_{i=1}^{20}\sum_{j=1}^{92}x_{ij}\cdot l_{ij}$$

$s.t.$

$$\sum_{i=1}^{20}x_{ij}=1,\ j=1,2,\ldots,92$$

$$x_{ij}=0,1$$

Constrains:

1) As each zone should in the charge of only one policy station;
2) According to the reality, each policy station could in charge of at least one zones;
3) when Xij=1 station i takes charge of j zone; when Xij=0 this station does not take charge of.

Here is the result:

Police Station Number	Intersection Area Number
1	1 67 68 69 71 73 74 75 76 78
2	2 40 43 44 70 72
3	3 54 55 65 66
4	4 57 60 62 63 64
5	5 49 50 51 52 53 56 58 59
6	6
7	7 30 32 47 48
8	8 33 46
9	9 31 34 35 45
10	10
11	11 26 27
12	12 25
13	13 21 22 23 24
14	14
15	15

16	16 36 37
17	17 41 42
18	18 80 81 82 83
19	19 77 79
20	20 84 85 86 87 88 89 90 91

In this question, we also need to assign the rest intersection areas which are arrived all police stations more than 3 minutes. We aim to the nearest station for those intersection. Here is the result for it:

Police Station Number	Intersection Area Number
2	39
7	61
15	28 29
16	38
20	92

In the conclusion, our assignment plan is as following:

Police Station Number	Intersection Area Number
1	1 67 68 69 71 73 74 75 76 78
2	2 39 40 43 44 70 72
3	3 54 55 65 66
4	4 57 60 62 63 64

5	5 49 50 51 52 53 56 58 59
6	6
7	7 30 32 47 48 61
8	8 33 46
9	9 31 34 35 45
10	10
11	11 26 27
12	12 25
13	13 21 22 23 24
14	14
15	15 28 29
16	16 36 37 38
17	17 41 42
18	18 80 81 82 83
19	19 77 79
20	20 84 85 86 87 88 89 90 91 92

Question 2

In last question, we already find there are 6 zones which all police stations cannot arrive within 3 minutes marked as failed points. Now we need to add two police stations considering from ‘quick reach’ to ‘duty balance’.

Step1.

In order to guarantee all zones can be arrived in time when events occur, we find out the zones that can arrive these points within 3 minutes as candidate points of police station. Here is result:

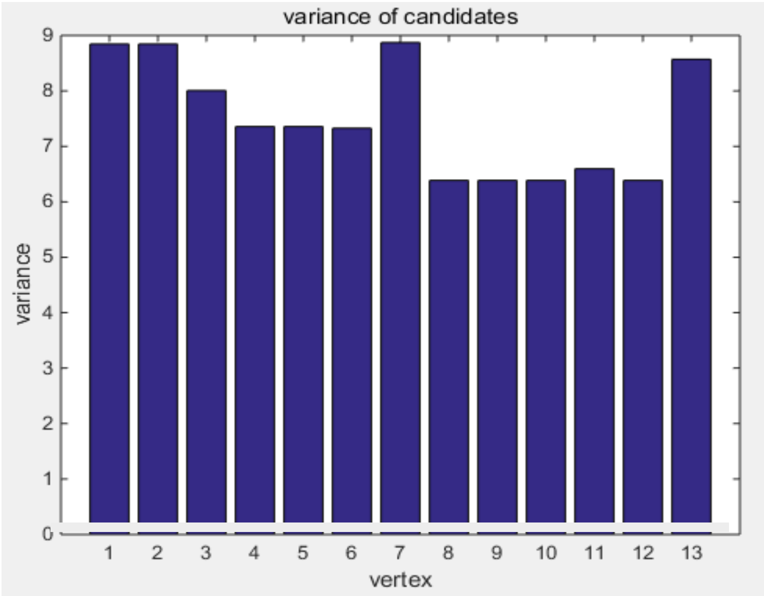
failed points	3 minute points
28	28, 29
29	28,29
38	38, 39, 40
39	38, 39, 40
61	48, 61
92	87, 88, 89, 90, 91, 92

All for all, these points as following: 28, 29, 38, 39, 40, 48, 61, 87, 88, 89, 90, 91, 92 are candidate points of police station.

Step2

According to the appendix crime frequency, we are supposed to consider their duty balance (variance of the crime times) if these points are set as police station.

$$\min\ \sigma^2 = \frac{1}{21}\sum_{i=1}^{21}(w_i - \overline{w})^2 \quad \text{Where} \quad \overline{w} = \frac{1}{21}\sum_{j=1}^{92} r_j$$



This figure illustrates that when we add one point from [v87, v88, v89, v91] and one from [v39, v40], we can mitigate overtime and duty imbalance problem in maximum extent. We randomly choose one pair of points in the project v39 and v87 as new station points. Then it becomes a problem of assigning 92 areas to 22 police stations, and it is similar to question1. We can use the same method to get the assignment:

police station	Intersection area										
1	0	0	0	0	0	0	0	0	0	0	0
2	1	67	68	69	71	73	74	75	76	78	0
3	2	40	43	44	70	72	0	0	0	0	0
4	3	54	55	65	66	0	0	0	0	0	0
5	4	57	60	62	63	64	0	0	0	0	0
6	5	49	50	51	52	53	56	58	59	0	0
7	6	0	0	0	0	0	0	0	0	0	0
8	7	30	32	47	48	61	0	0	0	0	0
9	8	33	46	0	0	0	0	0	0	0	0
10	9	31	34	35	45	0	0	0	0	0	0

11	10	0	0	0	0	0	0	0	0	0	0
12	11	26	27	0	0	0	0	0	0	0	0
13	12	25	0	0	0	0	0	0	0	0	0
14	13	21	22	23	24	0	0	0	0	0	0
15	14	0	0	0	0	0	0	0	0	0	0
16	15	28	29	0	0	0	0	0	0	0	0
17	16	36	37	0	0	0	0	0	0	0	0
18	17	41	42	0	0	0	0	0	0	0	0
19	18	80	81	82	83	84	0	0	0	0	0
20	19	77	79	0	0	0	0	0	0	0	0
39	38	39	0	0	0	0	0	0	0	0	0
87	20	85	86	87	88	89	90	91	92	0	0

After assignment, we find that there are also three failed points: v61, v28, v29 that cannot be arrived in three minutes. Therefore, it is not enough to add just 2 police stations, and we need to find an improved way to solve this problem.

Discussion

In last part, we can see that if we only add two points, we cannot perfectly solve this problem, so our group decides to add more points as police stations. As a whole, we will more focus on these three aspects:

a. Reaching time

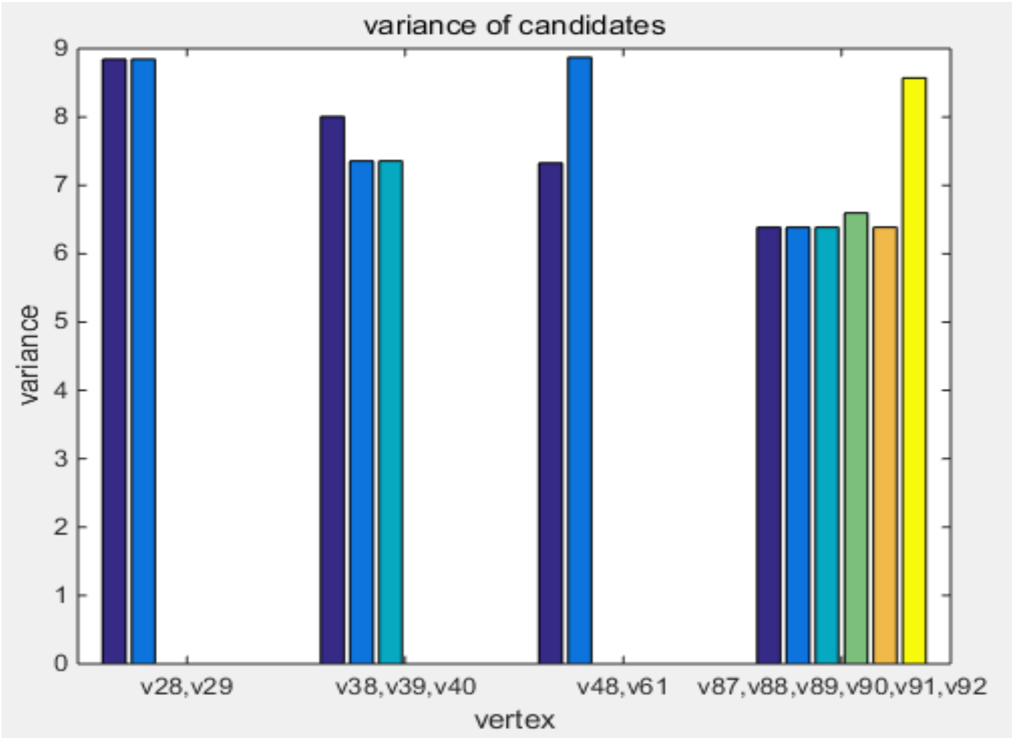
We are supposed to control the reaching time to every zone within 3 minutes and minimize the number of stations to minimize cost. Firstly, we have found all zones which near the 6 overtime zones, then we group them according to the distance.

failed points	3 minute points
28	28, 29
29	28,29
38	38, 39, 40
39	38, 39, 40
61	48, 61
92	87, 88, 89, 90, 91, 92

Because point 28 and 29 are capable to manage each other, so we can group them and only need choose one point from them. It is similar to [38, 39, 40]. So the possible new stations are four points from [28, 29], [38, 39, 40], [48, 61], [87, 88, 89, 90, 91, 92] respectively.

b. Duty balance

To make each police station duty equal and according to the grouped graph of variance of the crime times. We delete the option of v38, v61, v90, v92, which have the bigger variance in their groups.



So the possible new stations groups update to [28, 29], [39, 40], [48], [87, 88, 89, 91].

c. Shortest distance

Apart from the two factors before, we want to find which area set can make the total distance become shortest, which means the policeman can take the shortest time to reach the destination. So we set up the model

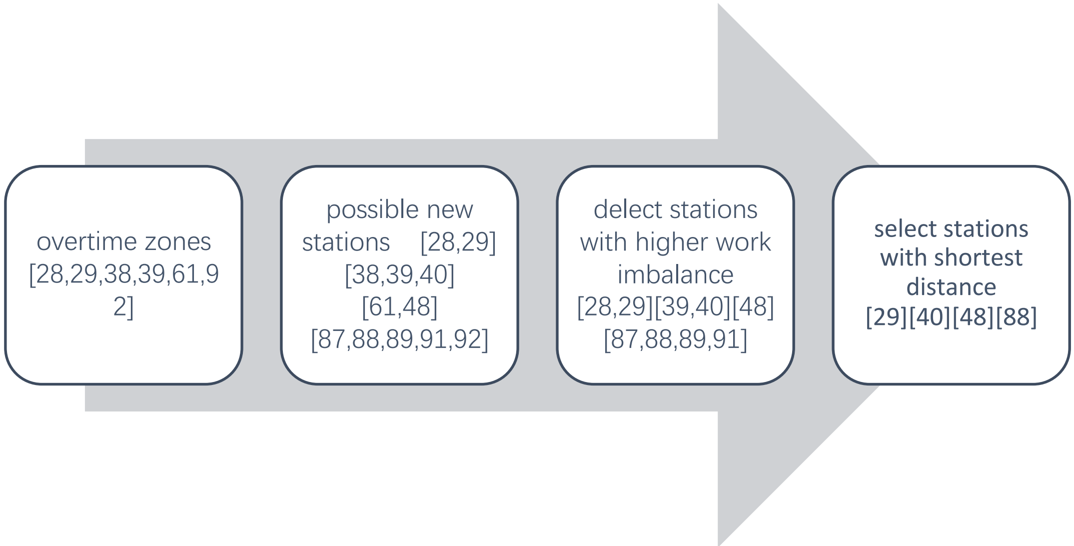
$$S=\sum_{j=1}^{92} X_{ij} r_j l_{ij}.$$

According to the model, we used MATLAB to calculate the distance between candidate station points and zones they take charge of it.

Option points	Distance
28	1.32815662
29	1.23328829
39	3.365203815
40	4.95619407

48	1.74
87	4.882049511
88	3.807815435
89	4.372173478
91	4.070758084

After select the shortest distance in each group, we get the improved answer 29, 39, 48, 88.

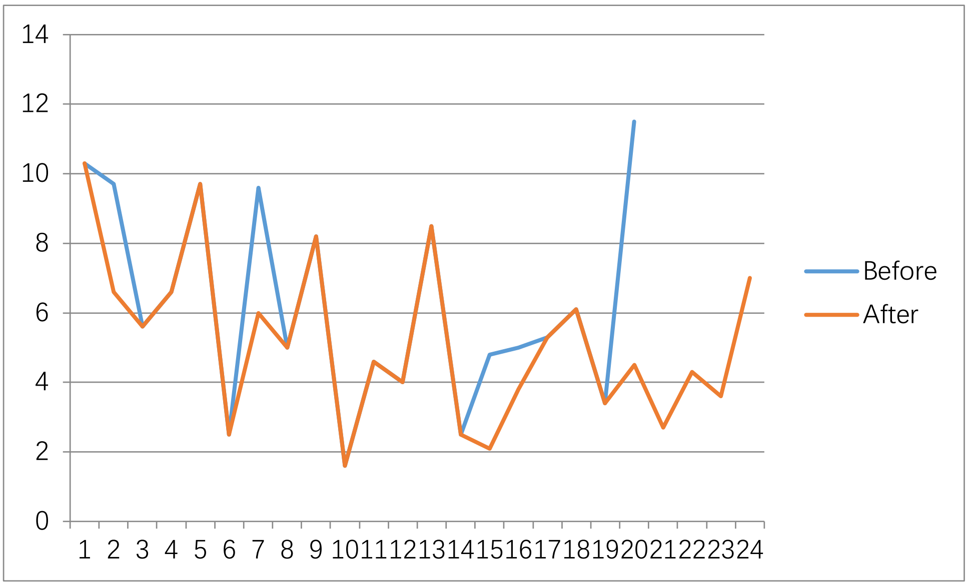


So the assignment of station and intersection is as following:

station point	intersection point
1	1 67 68 69 71 73 74 75 76 78
2	2 43 44 70 72
3	3 54 55 65 66
4	4 57 60 62 63 64
5	5 49 50 51 52 53 56 58 59
6	6
7	7 30 32
8	8 33 46
9	9 31 34 35 45
10	10
11	11 26 27

12	12	25					
13	13	21	22	23	24		
14	14						
15	15						
16	16	36	37				
17	17	41	42				
18	18	80	81	82	83		
19	19	77	79				
20	20	85	86				
28	28	29					
38	38	39	40				
47	47	48	61				
84	84	87	88	89	90	91	92

Then we compared the work of each police station after adding four points with the workforce before using EXCEL:



It is obvious that the overall workforce is reduced sharply due to the increasing number of police station; meanwhile, after adding [v29, v39, v48, v88], all zones can be reached at a fastest speed within 3 minutes, problem 2 is solved.

Conclusion

It is important to efficiently set police stations to solve emergencies and decrease crime rate. This model not only can be applied to the patrol service platform, but also can be widely applied to the layout of emergency service facilities such as fire stations and hospitals. Due to this project, we can see the network and modelling are useful in solving real life problems.

In question1， we calculate the shortest path for all the points with the Fordy method, and then we use the 0,1 to find the minimum weight for all the points. Actually, we can improve this model by assign the points considering duty balance, and traffic jams together. As well as the scale of staff in each police station. Furthermore, we may take the maximum flow into consider, and give every path a maximum capacity, then we may make a choice more realistic.

In the question 2, the multi-objective analytic hierarchy process (AHP) is used to separate the different objectives. If the weight can be determined, we can use the comprehensive evaluation and analysis model to get the optimal solution.

APPENDIX

Appendix1:

```
count=0;

Record_data = cell(92,1); %we create a 92 matrix to store every path weight o each
    points

for i = 1 :92

de_index = Routine_data(find(Routine_data(:,1)==i),2);    % find the points has the
direct path to every points

de_index = [Routine_data(find(Routine_data(:,2)==i),1); de_index]; % store every
location result into matrix
```

```
de_index = de_index(find(de_index<=92)); % make sure we get every index within
92 points
        n = length(de_index);
      count = count+n;    % get the scale of a matrix
Record_data{i} = zeros(n,2); % store every adjoin points and length matrix for each
points
for j = 1 : n
Record_data{i}(j,1) = de_index(j); % give the first column for index
Record_data{i}(j,2) = 0.1*sqrt((de_data(i,1) -
de_data(de_index(j),1))^2+(de_data(i,2) - de_data(de_index(j),2))^2); % calculate the
distinct between every points for each point
end
end
Adjoin_matrix = zeros(count,3);
index_adj = 1;
fori = 1 :92
        [n1,n2] = size(Record_data{i});    % find the adjoin points for each points
        n = n1;
for j = 1 : n
%
Adjoin_matrix(index_adj,:) = [i,Record_data{i}(j,1),Record_data{i}(j,2)]; % store a
matrix i is stands for pints and give the matrix for each points
```

```
index_adj = index_adj + 1;

end

end

mat=zeros(92,92); % find the initial matrix A

fori=1:92

for j=1:280

if Adjoin_matrix(j,1)==I % reorder the 92 matrix into one matrix and all the direct

weight path matrix

                    mat(i,Adjoin_matrix(j,2))=Adjoin_matrix(j,3);

end

end

end

mat(mat==0)=inf; % if there n path set the minimum weight 0 into infinite

for i=1:92

      mat(i,i)=0;

end

function [D,path]=flora (blinjie) % we create a Flora method function

n=size(blinjie,1); % find the initial size of matrix

D=blinjie;

path=zeros(n,n);

fori=1:n

for j=1:n
```

```
if D(i,j)~=inf    % if there are path between two points then we can use the direct path
path(i,j)=j;
end
end
end
for k=1:n
fori=1:n
for j=1:n
if D(i,k)+D(k,j)<D(i,j)    % calculate all the path for each pints to 91 points
D(i,j)=D(i,k)+D(k,j);      % set all minimum weight in to weight matrix
path(i,j)=path(i,k);
end
end
end
end
end
```

input:

[D,path]=flora(mat)   % input initial matrix and path matrix to get the all the shortest path

Output: D—-A matrix；path—-P matrix

% we use the method to plot the graph

Appendix1:

```
d=D(1:20,:)% take the first 20 station
x=[];
    for j=1:92
    x=[x,find(d(:,j)==min(d(:,j)))]% find the shortest distance of 92 point to 20 stations
    end
S1=find(x==1)%find all the zones in the charge of first station
S2=find(x==2)
S3=find(x==3)
S4=find(x==4)
S5=find(x==5)
S6=find(x==6)
S7=find(x==7)
S8=find(x==8)
S9=find(x==9)
S10=find(x==10)
S11=find(x==11)
S12=find(x==12)
S13=find(x==13)
S14=find(x==14)
S15=find(x==15)
S16=find(x==16)
S17=find(x==17)
S18=find(x==18)
S19=find(x==19)
S20=find(x==20)
```

Appendix2:

```
S1(find(d(1,S1)>3))=[]% for 20 stations removing the distance that arrival time id longer than 3
minutes
S2(find(d(2,S2)>3))=[]
S3(find(d(3,S3)>3))=[]
S4(find(d(4,S4)>3))=[]
S5(find(d(5,S5)>3))=[]
```

```
S6(find(d(6,S6)>3))=[]
S7(find(d(7,S7)>3))=[]
S8(find(d(8,S8)>3))=[]
S9(find(d(9,S9)>3))=[]
S10(find(d(10,S10)>3))=[]
S11(find(d(11,S11)>3))=[]
S12(find(d(12,S12)>3))=[]
S13(find(d(13,S13)>3))=[]
S14(find(d(14,S14)>3))=[]
S15(find(d(15,S15)>3))=[]
S16(find(d(16,S16)>3))=[]
S17(find(d(17,S17)>3))=[]
S18(find(d(18,S18)>3))=[]
S19(find(d(19,S19)>3))=[]
S20(find(d(20,S20)>3))=[]

X=[413    403 383.5      381 339 335 317 334.5      333 282 247 219 225 280 290 337 415 432 418
444 251 234 225 212 227 256 250.5      243 246 314 315 326 327 328 336 336 331 371 371 388.5
411 419 411 394 342 342 325 315 342 345 348.5      351 348 370 371 354 363 357 351 369 335 381
391 392 395 398 401 405 410 408 415 418 422 418.5      405.5      405 409 417 420 424 438 438.5
434 438 440 447 448 444.5      441 440.5      445 444];
Y=[359    343 351 377.5      376 383 362 353.5      342 325 301 316 270 292 335 328 335 371 374
394 277 271 265 290 300 301 306 328 337 367 351 355 350 342.5      339 334 335 330 333 330.5
327.5      344 343 346 342 348 372 374 372 382 380.5      377 369 363 353 374 382.5      387 382 388
395 381 375 366 361 362 359 360 355 350 351 347 354 356 364.5      368 370 364 370 372 368 373
376 385 392 392 381 383 385 381.5      380 360];
    h1=plot(X(1:20),Y(1:20),'bo');
```

```
hold on
h2=plot(X(21:92),Y(21:92),'r*');

legend([h1,h2],'Police Station','Zone','Location','southeast');
for k=1:92
   text(X(k)+1,Y(k)+1,sprintf('%d',k) );
end

D=zeros(92,92);
D(1,69)=1;D(1,74)=1;D(1,75)=1;D(1,78)=1;
D(2,40)=1;D(2,43)=1;D(2,44)=1;D(2,70)=1;
D(3,44)=1;D(3,45)=1;D(3,55)=1;D(3,65)=1;
D(4,39)=1;D(4,57)=1;D(4,62)=1;D(4,63)=1;
D(5,47)=1;D(5,49)=1;D(5,50)=1;
D(6,47)=1;D(6,50)=1;
D(7,15)=1;D(7,32)=1;D(7,34)=1;D(7,47)=1;
D(8,9)=1;D(8,33)=1;D(8,46)=1;D(8,47)=1;
D(9,8)=1;D(9,34)=1;D(9,35)=1;
D(10,26)=1;D(10,34)=1;
D(11,25)=1;D(11,26)=1;D(11,22)=1;
D(12,25)=1;D(12,27)=1;
D(13,22)=1;D(13,23)=1;D(13,24)=1;
```

D(14,16)=1;D(14,21)=1;

D(15,7)=1;D(15,28)=1;D(15,31)=1;

D(16,14)=1;D(16,36)=1;D(16,38)=1;

D(17,41)=1;D(17,40)=1;D(17,42)=1;D(17,81)=1;

D(18,73)=1;D(18,80)=1;D(18,81)=1;D(18,83)=1;

D(19,77)=1;D(19,79)=1;

D(20,85)=1;D(20,86)=1;D(20,89)=1;

D(21,14)=1;D(21,22)=1;

D(22,11)=1;D(22,13)=1;D(22,21)=1;

D(23,13)=1;

D(24,13)=1;D(24,25)=1;

D(25,11)=1;D(25,12)=1;D(25,24)=1;

D(26,10)=1;D(26,11)=1;D(26,27)=1;

D(27,12)=1;D(27,26)=1;

D(28,15)=1;D(28,29)=1;

D(29,28)=1;D(29,30)=1;

D(30,29)=1;D(30,48)=1;

D(31,15)=1;D(31,32)=1;D(31,34)=1;

D(32,7)=1;D(32,31)=1;D(32,33)=1;

D(33,8)=1;D(33,32)=1;D(33,34)=1;

D(34,9)=1;D(34,10)=1;D(34,7)=1;D(34,31)=1;D(34,33)=1;D(34,37)=1;

D(35,9)=1;D(35,36)=1;D(35,45)=1;

D(36,16)=1;D(36,35)=1;D(36,37)=1;D(36,39)=1;

D(37,34)=1;D(37,36)=1;

D(38,16)=1;D(38,39)=1;D(38,41)=1;

D(39,4)=1;D(39,36)=1;D(39,38)=1;D(39,40)=1;

D(40,2)=1;D(40,17)=1;D(40,39)=1;

D(41,17)=1;D(41,38)=1;D(41,92)=1;

D(42,17)=1;D(42,43)=1;

D(43,2)=1;D(43,42)=1;D(43,70)=1;D(43,72)=1;

D(44,2)=1;D(44,3)=1;D(44,67)=1;

D(45,3)=1;D(45,35)=1;D(45,46)=1;

D(46,8)=1;D(46,45)=1;D(46,55)=1;

D(47,5)=1;D(47,6)=1;D(47,7)=1;D(47,8)=1;D(47,48)=1;

D(48,30)=1;D(48,47)=1;D(48,61)=1;

D(49,5)=1;D(49,50)=1;D(49,53)=1;

D(50,5)=1;D(50,6)=1;D(50,49)=1;D(50,51)=1;D(50,59)=1;

D(51,50)=1;D(51,52)=1;D(51,59)=1;

D(52,51)=1;D(52,53)=1;D(52,56)=1;

D(53,49)=1;D(53,52)=1;D(53,54)=1;

D(54,53)=1;D(54,55)=1;D(54,63)=1;

D(55,3)=1;D(55,46)=1;D(55,54)=1;

D(56,52)=1;D(56,57)=1;

D(57,4)=1;D(57,56)=1;D(57,58)=1;D(57,60)=1;

D(58,57)=1;D(58,59)=1;

D(59,50)=1;D(59,51)=1;D(59,58)=1;

D(60,57)=1;D(60,61)=1;D(60,62)=1;

D(61,48)=1;D(61,60)=1;

D(62,4)=1;D(62,60)=1;D(62,85)=1;

D(63,4)=1;D(63,54)=1;D(63,64)=1;

D(64,63)=1;D(64,65)=1;D(64,76)=1;

D(65,3)=1;D(65,64)=1;D(65,66)=1;

D(66,65)=1;D(66,67)=1;D(66,76)=1;

D(67,44)=1;D(67,66)=1;D(67,68)=1;

D(68,67)=1;D(68,69)=1;D(68,75)=1;

D(69,1)=1;D(69,68)=1;D(69,70)=1;D(69,71)=1;

D(70,2)=1;D(70,43)=1;D(70,69)=1;

D(71,69)=1;D(71,72)=1;D(71,74)=1;

D(72,43)=1;D(72,71)=1;D(72,73)=1;

D(73,18)=1;D(73,72)=1;D(73,74)=1;

D(74,1)=1;D(74,71)=1;D(74,73)=1;D(74,80)=1;

D(75,1)=1;D(75,68)=1;D(75,76)=1;

D(76,64)=1;D(76,66)=1;D(76,75)=1;D(76,77)=1;

D(77,19)=1;D(77,76)=1;D(77,78)=1;

D(78,1)=1;D(78,77)=1;D(78,79)=1;

D(79,19)=1;D(79,78)=1;D(79,80)=1;

```
D(80,74)=1;D(80,79)=1;D(80,18)=1;
D(81,17)=1;D(81,18)=1;D(81,82)=1;
D(82,81)=1;D(82,83)=1;D(82,90)=1;
D(83,18)=1;D(83,82)=1;D(83,84)=1;
D(84,83)=1;D(84,85)=1;D(84,89)=1;
D(85,62)=1;D(85,84)=1;D(85,20)=1;
D(86,20)=1;D(86,87)=1;D(86,88)=1;
D(87,86)=1;D(87,88)=1;D(87,92)=1;
D(88,86)=1;D(88,87)=1;D(88,88)=1;D(88,91)=1;
D(89,20)=1;D(89,84)=1;D(89,90)=1;D(89,88)=1;
D(90,89)=1;D(90,82)=1;D(90,91)=1;
D(91,88)=1;D(91,90)=1;D(91,92)=1;D(91,87)=1;
D(92,41)=1;D(92,91)=1;D(92,87)=1;

for i=1:92
    for j=1:92
        if  D(i,j)==1
            line([X(i),X(j)],[Y(i),Y(j)])    %%
            %%line([1,2],[3,4])½
            D(i,j)=sqrt((X(i)-X(j))^2+(Y(i)-Y(j))^2);
        end
    end
```

```
end
```