Paper of problem B: Search and Find

Shenzhen Middle School, China
Advisor: Wentao Zhang
Team members: Gong Qianyi, Feng Kengcheng, Song Yuanmingqing, Zou Xinhui

Abstract

In this paper, we undertake the search and find problem. In two parts of searching, we use different way to design the model, but we use the same algorithm to calculate the main solution. In Part 1, we assume that the possibilities of finding the ring in different paths are different. We give weight to each path according to the possibility of finding the ring in the path. Then we simplify the question as pass as more weight as possible in limited distance. To simplify the calculating, we use Greedy algorithm and approximate optimal solution, and we define the values of the paths(according to the weights of paths) in Greedy algorithm. We calculate the possibility according to the weight of the route and to total weights of paths in the map. In Part 2, firstly, we limit the moving area of the jogger according to the information in the map. Then we use Dijkstra arithmatic to analysis the specific area of the jogger may be in. At last, we use greedy algorithm and approximate optimal solution to get the solution.

Key words: weight of path, Greedy algorithm, approximate optimal solution, Dijkstra arithmatic

Catalogue

Part 1

Restatement

Hopkinton State Park is located in the towns of Hopkinton and Ashland and includes the picturesque Hopkinton Reservoir.The park offers a variety of year round recreational activities including life guarded swimming beaches, stocked fishing, reservable group picnic sites, tree shaded, first come first served, picnic areas, 10 miles of marked trails, open field space and a concrete boat launching ramp for non-motorized watercraft. It can contain hundreds of poeple. It is not easy to find a ring at night in it.

Problem analysis

For a seeker who wants to find a small object in a area, it is the best to search every small places and corner. However, the problem is the seeker cannot search the whole area, and his radius of action is limited. If the possibility of finding object is constant everywhere, the seeker just need to search random places which are not repeated, but the question is the possibility is often different in different places in the real life. Therefore, we need to analysis and define different possibility of each path to organize the most efficient route.

Assumptions

1. There's no accident to delay the searching, such as dangerous cases, pavement behavior, vehicles, eating and drinking;

2. The seeker starts at the main entrance of the park, but he does not have fixed terminal point;

3. The power of the pen light flashlight is enough for the seeker to search for at least 2 hours;

4. The ring only lost in the park, and it can be found on the paths of the park;

5.The possibility of finding out the ring on each path depends on its the feature(position, condition, length, etc);

Symbols

C: the lengths of paths.

D: the weights of paths.

R: the values of paths.

$x;y;U;V;W$: the symbols of crosses.

P:the possibility of finding out the ring.

a: the parameter of weight of the path.

d: the total weight of one kind of public facility.

Solve the problem

Diagram and Starting point
We assume that the seeker starts at the main entrance of the map. To simplify solving the problem. We draw a diagram of the paths.

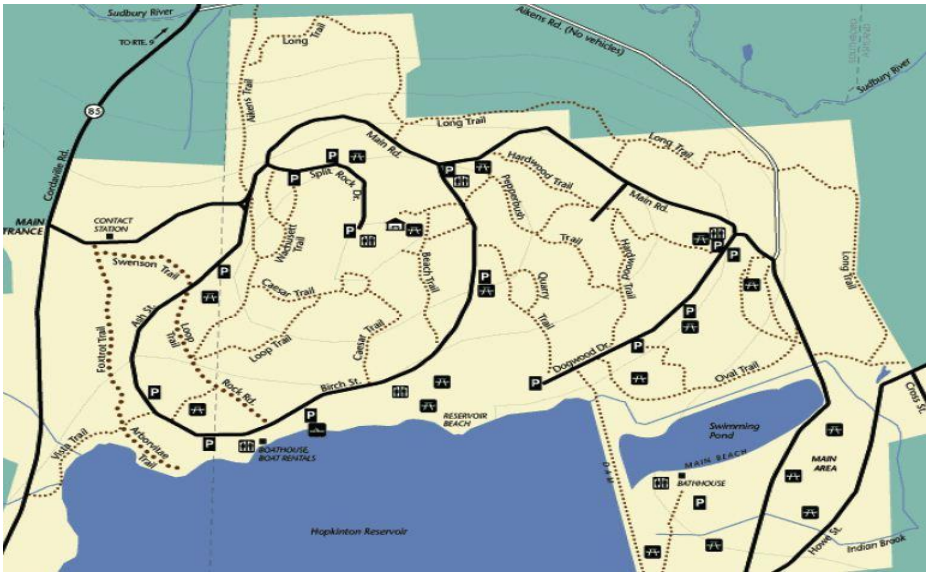

Figure 1

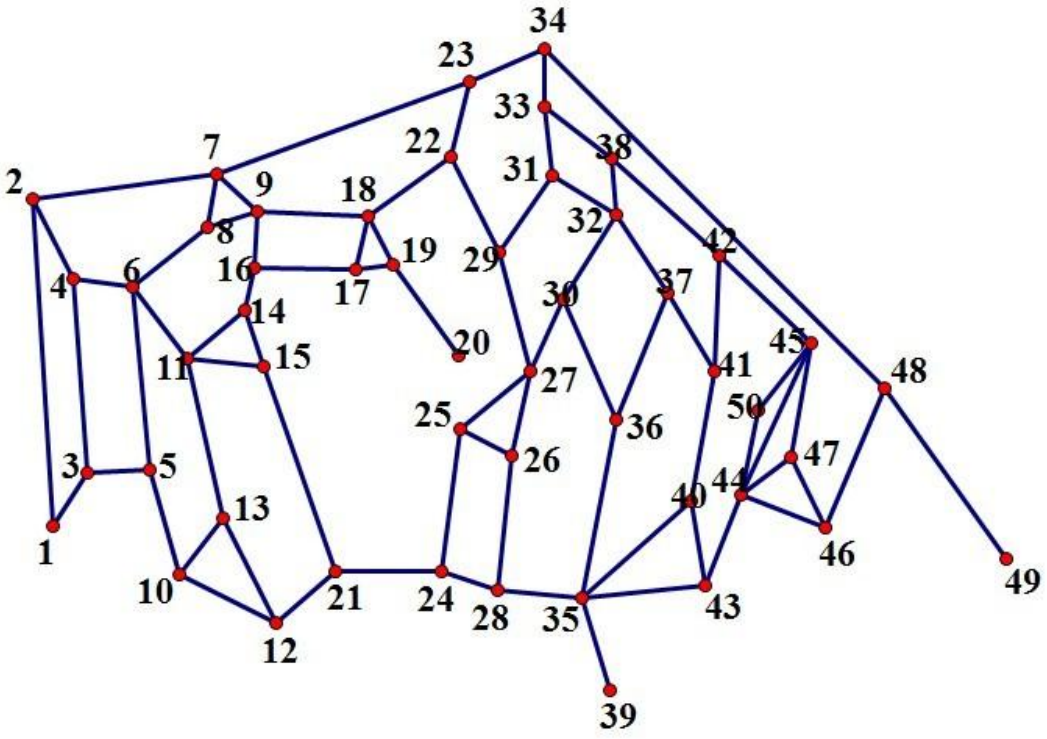

Figure 2

As **figure 1** and **figure 2** show above, the vertex which is most approach to the entrance is vertex 2. Therefore, we assume that the seeker is starting at vertex 2.

The weight of paths

Our objective was to create a model to determine the efficiency of various route chart designs and use the model to find an optimal design. We decided that optimal route occurs when seeker can find the most valuable route to search and use as less time as possible. To determine how our model could best meet the optimum, we compute the weighs of paths.

Path	Weights
Paved roud	27.4/mile
Hiking trail and self-guided nature trail	21.9/mile
restroom	4 each
Picnic area	2 each
Parking and boat ramp	3 each
Pavilion	1.5 each

Table 1

We assume the public facilities around the path can influence the weight of the path. According the table above you can see that we assume weight of the path depends on the condition and the length of the path. The weights of the roads is larger than that of the pavements, because the possibility of finding the class ring on roads is larger than that on pavements since roads are less rugged than pavements. The longer the road or the pavement, the larger weight it has, but their difference is not large. Thus, we set weights for roads and pavements, and data of roads is little bigger than that of pavements. To find the weight of the road or the pavement, we use the coefficient of weight multiplied by the length of the road or the pavement.

More importantly, since the flow of the people is much larger than roads and pavements, the value is larger if there are several public facilities. The larger flow of the people, the easier to lose a small object. Due to the flow of the people, we set different weight of these public facilities. The weightb of the restroom is largest. The weight of the parking lot and boat ramp are larger than that of the picnic area, and the weight of pavilion is smallest.

We set that $C(x,y)$ represents the length of path between cross x and y, and $D(x,y)$represents the weight of path between cross x and y.

We establish **equation 1**

$$D=aC+\sum d$$

a is according the kind of path(27.4or21.9), d is the total weight of the pubic facilities around the path(quantity*parameter)

Then we get a set of data D and C. *(in appendix 1)*

Greedy algorithm

As the restatement shows above, we have to find the most efficent route which contain the biggest weights of connecting paths in the limited distance. Because of the large calculating work in listing the possible route and comparing their weight, we use greedy algorithm to simplify the calculating work. Greedy algorithm is that each step chooses parital optimal solution, and at last we can get the approximate optimal solution.

The optimal solution within two steps

According to **figure 3**, set the adjacent vertex set as adj[*U*], adj[*U*]={*V1*,*V2*,*V3*,…}. Assume the seeker arrives at vertex *U*. Now he continues for two steps in random ways, and we can calculate the optimal cross *V*. Then *V* is the optimal vertex within a two-step.

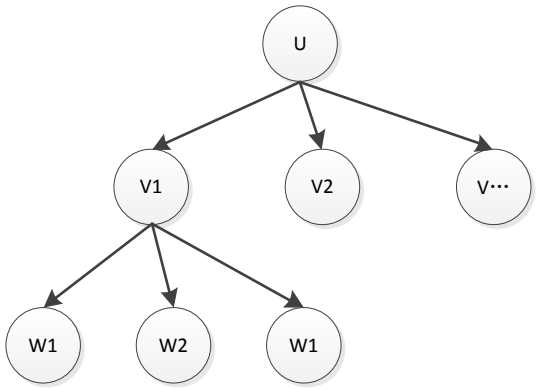

Figure 3

We define the value of the path *R=D/C*, and the value of each vertex is the same as the biggest value of the adjacent path which has not been passed. The two-step value from *U* to *V1* is the value of path *(U,V1)* plus the value of vertex *V1*. Similarly, the two-step value from *U* to *V2* is the value of path *(U,V2)* plus the value of vertex *V2*. By analogy, we can calculate other two-step values. Then we choose path (*U* to *V1*, *U* to *V2*, *U* to *V3*....*U* to *Vn*) which owns the biggest two-step value as the optimal solution.

Method explanation

When we arrive at a vertex, we calculate its optimal solution within two steps. Then we get the path and go to next vertex. Begin at the new cross and calculate its optimal solution within two steps again. Repeat the action, until the the distance is longer than the longest distance that the seeker can arrive, or the two-step value on a vertex is zero.*(pseudocode in appendix 2)*

Then, we get the solution

2----7----8----6----5----10----12----21----24----28----35----40----41----42----38----32----31----29----27----25----26----27----29----22----18----19----20----19----17----16----14----11----13----10----9----23----34----48----46----44----45----50----44----43----35----36----37----41----40----43----35----39

Possibility of finding the ring

According to the greedy algorithm shows above, we can get the most efficient route. According to our assumption that the ring can be found on the paths of the park, the seeker can 100% find the ring if he walk pass all the paths. Because we define the weight of the path according to possibility of finding the ring, we can calculate the possibility of finding the ring in the most efficient route by **equation 2**

$$P=(\sum D)/D_{total}$$

$\sum D$ is the total weight of paths in the most efficient route(if repeatedly pass a path, only calculate it once). D_{total} is the total weight of paths which we define on the map. Then we can get the solution of finding the ring.

Therefore, according to the solution of route, we get the answer.

$$\sum D=308.87$$

$$D_{total}=415.67$$

$$P=(\sum D)/D_{total}=308.87/415.67=74.3\%$$

Part 2

Problem analysis

Finding a jogger in the park is always a difficuilt job, because the man is often move so it is impossble to determine his real time position in a map. As a matter of fact, the seeker do not need to search every places, he just need to search some place is suitable for himself. Because of the area for the jogging man to jog is limited, we can ignore see area in the map to simplify our calculating.

Assumption

1.Regardless of whether the jogger is conscious or not, he cannot go for more than 5 miles.

2.The jogger comes by car and park his car in parking area.

3.The most paths that the jogger jogs are small trails.

4.The jogger cannot cross the BLM public lands border in the map.

Symbols

A;B;C: the symbols of an area.

G:connected graph

V:set of total poins

S;U:set of points

E:set of paths

x;y: the symbols of crosses.

C:length of path

P:set of paths in the route

Solve the problem

Area chooses

Because the jogger cannot cross the lands border and jog on the road for a long while, we assume that the jogger does not jog in the big road and we separate the map into three area.

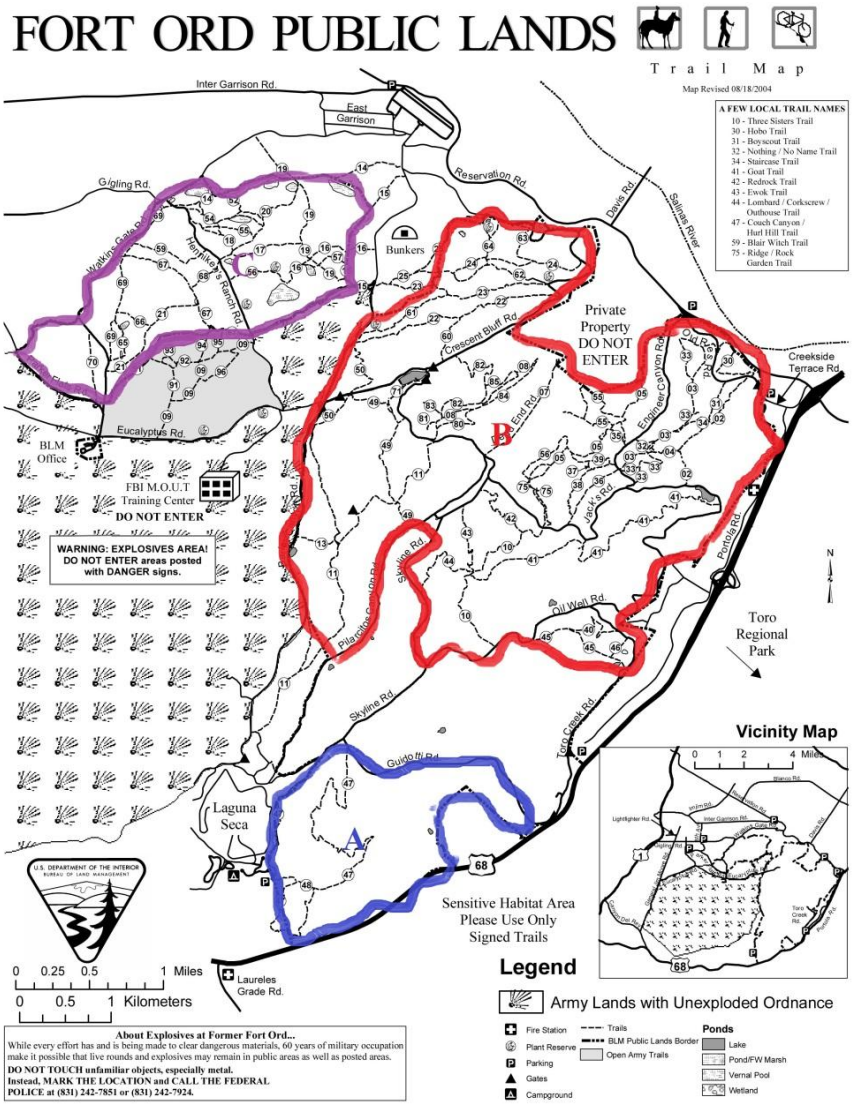

Figure 4

According to **figure 4**, area C does not have parking area, so the jogger cannot be there. Area A is too small for the jogger to run for 5 miles. Thus, the best choice to look for the jogger is looking for him in area B and it has most possibility to find him, because area B has 2 parking area which has the most parking places in the map.

Diagram and starting points

We assume that the jogger starts at parking area. To simplify solving the problem. We draw a diagram of the paths in area B.

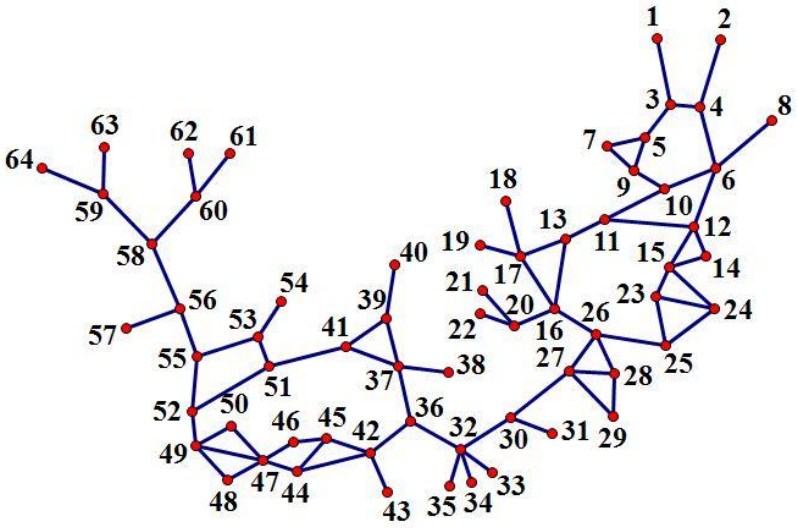

Figure 5

As figure 4 and figure 5 show above, point 1 and point 2 are the parking area, so we assume that they are the starting points.

Dijkstra's algorithm

Set G=(V,E) is a connected graph, and divide the points in the graph into two sets, set S and set U. Set S contain the points which are already calculated the shortest distance between them and the starting point O. Set U At first, set S only contain point O. Set U contain the points which have not calculated the shortest distance from point O. Each time the problem calculate a shorest path of a point to point O, and then move the poin from set U to set S. When all the points move to set S, the problem will stop. In the process of move points, always ensure that the shortest distance between point O and the points in set S do not longer than any the shortest distance between point O and the points in set U In addition, each point corresponds a distance. The distances of points in set S are the shortest distance between O and those points. The distances of point in set U are the shortest distance which between O and those points, only containing the points in set S as intermediate points.

We need to ignore some points which the shortest distance between them and the starting points are longer than five miles, because the jogger is impossble to be there. Then we use Dijkstra arithmatic to calculate the shortest distance between each points and starting points O.

According to figure 4 and figure 5, we can get a set of data C(x,y), the length of each path*(in appendix 3)*.Then we can calculate the acceptable points.(figure 6)*(in appendix 4)*

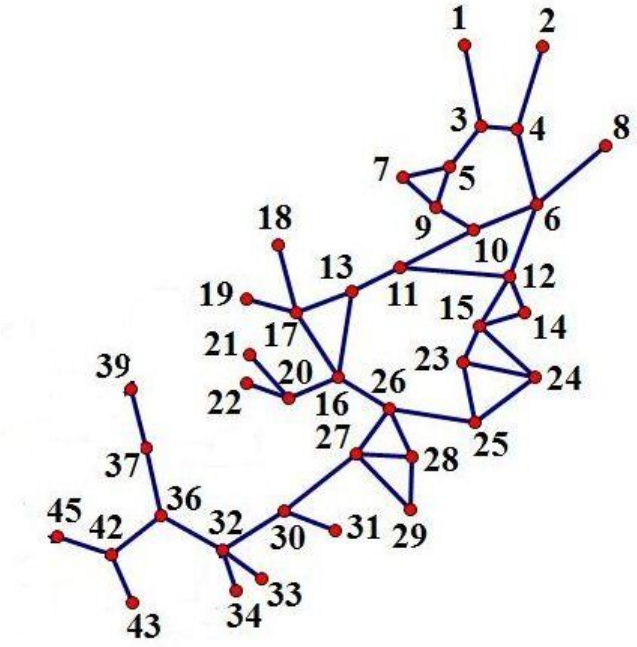

Figure 6

Solution

To simplify the searching route, we can use *Greedy algorithm*(in Part 1) and *The optimal solution within two steps*(in Part 1). We assume that the lengths of the paths are the values of the paths.

Then we get the solutions of two starting points

P_1:1---3---4---6---12---15---24---23---25---26---27---30---32---36---42---45

P_2:2---4---6---12---15---24---23---25---26---27---30---32---36---42---45

Because of the length of P_1 longer than that of P_2

So we select P_1 as the searching route.

Appendix 1

Explanation: the datas in C(*x*,y) times 3.43/64 are the lengths of paths in mile.

C(1,2)=13.97;C(1,3)=5.24;C(2,4)=0.47;C(2,7)=5.46;C(3,4)=7.99;
C(3,5)=1.47;C(4,6)=4.16;C(7,8)=0.39;C(7,9)=0.46;C(5,6)=6.37;
C(6,8)=4.25;C(8,9)=0.50;C(6,11)=3.27;C(5,10)=0.72;C(10,13)=2.34;
C(10,12)=3.45;C(11,13)=1.58;C(9,10)=2.67;C(9,23)=15.43;C(12,21)=4.1;
C(11,15)=5.4;C(11,14)=3.88;C(14,16)=1.37;C(16,17)=2.03;C(17,18)=2.85;
C(9,18)=1.67;C(15,21)=5.89;C(14,15)=2.79;C(17,19)=2.47;C(18,19)=0.4;
C(18,22)=6.09;C(19,20)=4.83;C(21,24)=2.85;C(24,25)=5.14;C(25,27)=2.50;
C(25,26)=1.85;C(24,28)=4.86;C(26,28)=1.27;C(28,35)=6.86;C(26,27)=0.71;
C(27,29)=1.95;C(22,29)=1.90;C(23,34)=8.97;C(29,31)=1.84;C(27,30)=4.89;
C(31,32)=3.55;C(31,33)=5.16;C(33,34)=1.45;C(33,38)=0.81;C(32,38)=1.39;
C(30,32)=2.50;C(30,36)=4.48;C(32,37)=0.70;C(36,37)=3.63;C(35,36)=3.32;
C(35,39)=17.32;C(35,43)=4.17;C(35,40)=1.38;C(37,41)=4.92;C(38,42)=4.52;
C(34,48)=26.13;C(41,42)=0.79;C(40,41)=5.22;C(40,43)=3.92;C(43,44)=4.64;
C(42,45)=1.48;C(44,50)=1.00;C(45,50)=4.29;C(44,45)=3.91;C(45,47)=2.91;
C(44,46)=0.35;C(44,47)=0.76;C(46,47)=0.86;C(46,48)=0.50;C(48,49)=23.52;

D(1,2)=13.97;D(1,3)=3.93;D(2,4)=0.35;D(2,7)=5.46;D(3,4)=6.00;
D(3,5)=1.10;D(4,6)=3.12;D(7,8)=0.39;D(7,9)=0.46;D(5,6)=11.37;
D(6,8)=9.25;D(8,9)=0.50;D(6,11)=4.45;D(5,10)=0.72;D(10,13)=3.75;
D(10,12)=10.45;D(11,13)=1.19;D(9,10)=2.00;D(9,23)=11.57;D(12,21)=11.1;
D(11,15)=4.05;D(11,14)=2.91;D(14,16)=1.03;D(16,17)=1.52;D(17,18)=2.14;
D(9,18)=1.25;D(15,21)=4.42;D(14,15)=2.09;D(17,19)=4.85;D(18,19)=3.4;
D(18,22)=6.09;D(19,20)=19.83;D(21,24)=10.85;D(24,25)=3.86;D(25,27)=6.00;
D(25,26)=1.39;D(24,28)=9.86;D(26,28)=1.27;D(28,35)=8.15;D(26,27)=0.71;
D(27,29)=8.95;D(22,29)=4.90;D(23,34)=6.73;D(29,31)=6.84;D(27,30)=5.67;
D(31,32)=4.66;D(31,33)=5.16;D(33,34)=1.09;D(33,38)=0.81;D(32,38)=1.39;
D(30,32)=1.88;D(30,36)=3.36;D(32,37)=0.53;D(36,37)=2.72;D(35,36)=2.49;
D(35,39)=19.99;D(35,43)=5.13;D(35,40)=4.38;D(37,41)=11.69;D(38,42)=12.52;
D(34,48)=19.60;D(41,42)=10.79;D(40,41)=13.22;D(40,43)=4.94;D(43,44)=3.48;
D(42,45)=1.48;D(44,50)=1.00;D(45,50)=4.97;D(44,45)=4.93;D(45,47)=2.91;
D(44,46)=0.26;D(44,47)=0.57;D(46,47)=0.86;D(46,48)=0.50;D(48,49)=29.52;

Appendix 2

Explanation: use the symbol in Part 1

Foreach V $V \mid adj[U]$

```
{
        int BestNextStep = 0;
          double BestNextRate = 0;
        If(C[U,V]!=0)
    {
            double NextRate = CalBestRateFromV(U, V);
            if (D[U, V] != 0)
 NextRate += R[U, V];

              if (BestNextRate < NextRate)
                    {
                          BestNextRate = NextRate ;
                          BestNextStep = V;
                    }
}

              if (BestNextStep == 0)
                    return;
               if (BestNextRate == 0)
                    return;

double NewLengthSum = LengthSum + C[U, BestNextStep];
                double NewValueSum = ValueSum + D[U, BestNextStep];
                D[U, BestNextStep] = 0;
                D[BestNextStep, U] = 0;

              Cal(BestNextStep, NewLengthSum, NewValueSum, Route + "----" +
BestNextStep.ToString());

}

    private double CalBestRateFromV(int U, int V)
          {
              double BestValue=0;
              for (int i = 1; i <= 50; i++)
              {
                  if (D[V, i] != 0
                  {
```

```
                    if (i == U) continue;
                    if (BestValue < R[V, i])
                        BestValue = R[V, i];
                }
            }
            return BestValue;
        }
```

Appendix 3

Explanation: Table 1 is the shortest distance between each point and point 2,table 2 is the shortest distance between each point and point 1. The datas in both table over 6.43 are the length in mile.

Table 1

1====6.56	2====0	3====4.72
4====1.12	5====5.36	6====4.04
7====6.52	8====5.91	9====6.94
10====5.34	11====6.51	12====7.08
13====7.85	14====7.84	15====8.56
16====8.6	17====8.87	18====10.9
19====10.7	20====10.28	21====11.19
22====11.86	23====9.27	24====10.11
25====10.56	26====11.3	27====13.52
28====12.34	29====12.94	30====16.98
31====19.85	32====20	33====24.09
34====23.88	35====34.44	36====27.12
37====30.77	38====35.73	39====31.23
40====33.14	41====33.74	42====31.41
43====31.89	44====33	45====31.96
46====32.95	47====33.78	48====35.09
49====35.86	50====34.63	51====36.75
52====40.37	53====50.08	55====45.69
56====50.28	57====50.86	58====51.85
59====52.84	60====53.96	61====56.21
62====55.82	63====53.59	64====53.44

Table 2

1====0	2====6.56	3====1.84
4====5.44	5====2.48	6====7.76
7====3.64	8====9.63	9====4.06
10====6.46	11====7.63	12====8.2
13====8.97	14====8.96	15====9.68
16====9.72	17====9.99	18====12.02
19====11.82	20====11.4	21====12.31
22====12.98	23====10.39	24====11.23
25====11.68	26====12.42	27====14.64
28====13.46	29====14.06	30====18.1
31====20.97	32====21.12	33====25.21
34====25	35====35.56	36====28.24
37====31.89	38====36.85	39====32.35

40====34.26	41====34.86	42====32.53
43====33.01	44====34.12	45====33.08
46====34.07	47====34.9	48====36.21
49====36.98	50====35.75	51====37.87
52====41.49	53====51.2	55====46.81
56====51.4	57====51.98	58====52.97
59====53.96	60====55.08	61====57.33
62====56.94	63====54.71	64====54.56

Appendix 4

Explanation: the datas in C(*x*,y) over 6.43 are the lengths of paths in mile.

(1,3)=1.84;(3,4)=3.6;(2,4)=1.12;
(5,3)=0.64;(4,6)=2.92;(6,8)=1.87;
(5,7)=1.16;(5,9)=1.58;(7,9)=1.92;
(9,10)=2.4;(6,10)=1.3;(10,11)=1.17;
(11,12)=0.57;(6,12)=3.16;(11,13)=1.34;
(12,14)=0.76;(12,15)=1.48;(14,15)=1.86;
(13,16)=0.75;(13,17)=1.02;(17,18)=2.03;
(17,19)=1.83;(16,17)=0.51;(16,20)=1.68;
(20,21)=0.91;(20,22)=1.58;(15,23)=0.71;
(15,24)=5.1;(23,24)=0.84;(23,25)=1.29;
(24,25)=0.64;(16,26)=2.76;(25,26)=0.74;
(26,27)=3.77;(26,28)=1.04;(27,28)=1.18;
(27,29)=0.61;(28,29)=0.6;(27,30)=3.46;
(30,31)=2.87;(30,32)=3.02;(32,33)=4.09;
(32,34)=3.88;(32,35)=14.44;(32,36)=7.12;
(36,37)=3.65;(37,38)=4.96;(37,39)=0.46;
(39,40)=1.91;(39,41)=7.36;(36,41)=6.62;
(36,42)=4.29;(42,43)=0.48;(42,44)=2.3;
(42,45)=0.55;(44,45)=1.04;(45,46)=0.99;
(46,47)=1.13;(44,47)=0.78;(47,48)=1.31;
(48,49)=2.93;(47,49)=2.08;(47,50)=0.85;
(49,50)=1.52;(41,51)=3.01;(50,52)=5.74;
(51,52)=4.81;(53,55)=4.39;(52,55)=5.32;
(55,56)=4.59;(56,57)=0.58;(56,58)=1.57;
(58,59)=0.99;(58,60)=2.11;(60,61)=2.25;
(60,62)=1.86;(59,63)=0.75;(59,64)=0.6