Math 381

Homework 4

Oct 29, 2021

Haochen Wang

I found 10 places that I'd like to visit: Beijing, Tianjin, Harbin, Daqing, Xi'an, Kunming, Shanghai, Jilin, Tokyo, and Kyoto. All the cities are in either China or Japan.

Below is a map for the location of all these ten cities.

Map for 10 places that I'd like to visit:

Now, I am going to present the set of distances as a chart in matrix form. I get the distance between any two cities mentioned above from the website "www.jisuan.info". Check the table below.

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Matrix for the set of distances:

In km	Beijing	Tianjin	Harbin	Daqing	Xi'an	Kunming	Shanghai	Jilin	Tokyo	Kyoto
Beijing	0	110	1058	1024	912	2090	1080	947	2100	1792
Tianjin	110	0	1069	1055	914	2074	972	942	2025	1708
Harbin	1058	1069	0	141	1970	3143	1691	219	1582	1432
Daqing	1024	1055	141	0	1932	3123	1739	327	1722	1569
Xi'an	912	914	1970	1932	0	1189	1220	1852	2803	2451
Kunming	2090	2074	3143	3123	1189	0	1955	3014	3722	3354
Shanghai	1080	972	1691	1739	1220	1955	0	1473	1765	1396
Jilin	947	942	219	327	2851	3014	1473	0	1447	1261
Tokyo	2100	2025	1582	1722	2803	3722	1765	1447	0	368
kyoto	1792	1708	1432	1569	2451	3354	1396	1261	368	0

Problem:

I want to find the shortest path connecting 5 of the cities above.

I am going to solve this problem by using a LP:

Define all the cities from the matrix at the first row (except the "In km" square at the first column) from left to right as city 1 to city 10 in order.

Let x_{ij} be the binary variable that $x_{ij} = 1$ if and only if I am at city j at step i (start at step 1).

Let e_{ab} be the binary variable that $e_{ab} = 1$ if and only if I travel from city a to b in one stop.

Let d_{ab} be the distance, in kilometers, from city a to city b.

Let f_{iab} be the binary variable that $f_{iab} = 1$ iff I visit city a at step i and city b at step i+1.

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Then, to find the shortest path, I have to minimize my object function for the total distance:

$$\sum_{a=1}^{10}\sum_{b=1}^{10} e_{ab}d_{ab}.$$

There are five constraints in this LP problem ($a, b, i, j \in Z^+$):

(1) We need to visit exactly one city on each of the 5 steps, so I have the constraint

$\sum_{j=1}^{10} x_{ij} = 1$ for all $i \in [1, 5]$, since there are only 5 total steps.

(2) Each city an be visited at most once, so I have the following constraint:

$\sum_{i=1}^{5} x_{ij} \leq 1$ for all $j \in [1, 10]$, since there are only 10 total cities.

(3) By the definition of f_{iab}, I have to make $f_{iab} = 0$ if I don't visit city a at step i or I don't visit city b at step i+1, so I have the following constraint:

$3f_{iab} \leq 1 + x_{ia} + x_{(i+1)b}$ for all $i \in [1, 4]$, since there are only 5 total steps.

(4) By the definition of f_{iab}, I have to make $f_{iab} = 1$ if I visit city a at step i and I visit city b at step i+1, so I have the following constraint:

$3f_{iab} \geq -1 + x_{ia} + x_{(i+1)b}$ for all $i \in [1, 4]$, since there are only 5 total steps.

(5) If I travel from city a to city b in a particular step, then by definition, $e_{ab} = 1$ and one of f_{iab} will be 1, which means $\sum_{i=1}^{4} f_{iab} = 1$. Otherwise, I will have $e_{ab} = \sum_{i=1}^{4} f_{iab} = 0$. This means e_{ab} should always be equal to the sum of f_{iab}. Also, I can not travel from one city to the same city. Thus, I have the following constraint:

$e_{ab} = \sum_{i=1}^{4} f_{iab}$ for all $a, b \in [1, 10]$ and $a \neq b$, since there are only 10 total cities that are available for me to travel.

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Thus, my LP is:

$$\text{Minimize } \sum_{a=1}^{10}\sum_{b=1}^{10} e_{ab}d_{ab}$$

Subject to

$$\sum_{j=1}^{10} x_{ij} = 1 \text{ for all } 1 \le i \le 5 \quad (1)$$

$$\sum_{i=1}^{5} x_{ij} \le 1 \text{ for all } 1 \le j \le 10 \quad (2)$$

$$3f_{iab} \le 1 + x_{ia} + x_{(i+1)b} \text{ for all } 1 \le i \le 4 \quad (3)$$

$$3f_{iab} \ge -1 + x_{ia} + x_{(i+1)b} \text{ for all } 1 \le i \le 4 \quad (4)$$

$$e_{ab} = \sum_{i=1}^{4} f_{iab} \text{ for all } 1 \le a, b \le 10 \quad (5)$$

$$x_{ij}, e_{ab}, f_{iab} \in [0, 1] \text{ for all } 1 \le i \le 5, 1 \le a, b, i \le 10 \quad (6)$$

Coding:

After finishing to build my LP, I will use the LP solver to solve this LP problem.

<u>Below is the code I write in Java to generate the lpsolve input file:</u>

```
1 public class math381hw3 {
2   // This gives the number of cities I want to visit.
3   public static int n = 5;
4   public static void main (String[] args) {
5     // This will assign the distance between any two cities from the map into an array.
6     int [][] distance = {{0, 110, 1058, 1024, 912, 2090, 1080, 947, 2100, 1792},
7                          {110, 0, 1069, 1055, 914, 2074, 972, 942, 2025, 1708},
8                          {1058, 1069, 0, 141, 1970, 3143, 1691, 219, 1582, 1432},
9                          {1024, 1055, 141, 0, 1932, 3123, 1739, 327, 1722, 1569},
10                         {912, 914, 1970, 1932, 0, 1189, 1220, 1852, 2803, 2451},
11                         {2090, 2074, 3143, 3123, 1189, 0, 1955, 3014, 3722, 3354},
12                         {1080, 972, 1691, 1739, 1220, 1955, 0, 1473, 1765, 1396},
13                         {947, 942, 219, 327, 2851, 3014, 1473, 0, 1447, 1261},
14                         {2100, 2025, 1582, 1722, 2803, 3722, 1765, 1447, 0, 368},
15                         {1792, 1708, 1432, 1569, 2451, 3354, 1396, 1261, 368, 0}};
16    // Print out the object function of the problem that will minimize the total distance among the cities I visit.
17    System.out.print("min: ");
18    for (int i = 1; i <= 10; i++) {
19      for (int j = 1; j <= 10; j++) {
```

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

```
20          if (i != j) {
21            if (i == 10 && j == 9) {
22              System.out.println(distance[i-1][j-1] + "e_" + i + "_" + j + ";");
23            } else {
24              System.out.print(distance[i-1][j-1] + "e_" + i + "_" + j + "+");
25            }
26          }
27        }
28      }
29      // Print out the fifth constraint from the problem.
30      for (int i = 1;i <= 10; i++) {
31        for (int j = 1;j <= 10; j++) {
32          if (i != j) {
33            System.out.print("e_" + i + "_" + j + "=");
34            for (int k = 1; k <= n - 1; k++) {
35              if (k == n - 1) {
36                System.out.println("f_" + k + "_" + i + "_" + j + ";");
37              } else {
38                System.out.print("f_" + k + "_" + i + "_" + j + "+");
39              }
40            }
41          }
42        }
43      }
44      // Print out the third constraint from the problem.
45      for (int i = 1; i <= n - 1; i++) {
46        for (int j = 1; j <= 10; j++) {
47          for (int k = 1; k <= 10; k++) {
48            if (j != k) {
49              System.out.println("3f_" + i + "_" + j + "_" + k + "<=1+x_" + i
50                      + "_" + j + "+x_" + (i+1) + "_" + k + ";");
51            }
52          }
53        }
54      }
55      // Print out the fourth constraint from the problem.
56      for (int i = 1; i <= n - 1; i++) {
57        for (int j = 1; j <= 10; j++) {
58          for (int k = 1; k <= 10; k++) {
59            if (j != k) {
60              System.out.println("3f_" + i + "_" + j + "_" + k + ">=-1+x_" + i
61                      + "_" + j + "+x_" + (i+1) + "_" + k + ";");
62            }
63          }
64        }
65      }
```

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

```
66    // Print out the first constraint from the problem.
67    for (int i = 1; i <= n; i++) {
68      for (int j = 1; j <= 10; j++) {
69        if (j == 10) {
70          System.out.println("x_" + i + "_" + j + "=1;");
71        } else {
72          System.out.print("x_" + i + "_" + j + "+");
73        }
74      }
75    }
76    // Print out the second constraint from the problem.
77    for (int j = 1; j <= 10; j++) {
78      for (int i = 1; i <= n; i++) {
79        if (i == n) {
80          System.out.println("x_" + i + "_" + j + "<=1;");
81        } else {
82          System.out.print("x_" + i + "_" + j + "+");
83        }
84      }
85    }
86    // Print the requirement that every variable must be binary.
87    System.out.print("bin ");
88    for (int i = 1; i <= n; i++) {
89      for (int j = 1; j <= 10; j++) {
90        System.out.print("x_" + i + "_" + j + ",");
91      }
92    }
93    for (int i = 1; i <= 10; i++) {
94      for (int j = 1; j <= 10; j++) {
95        if (i != j) {
96          System.out.print("e_" + i + "_" + j + ",");
97        }
98      }
99    }
100    for (int i = 1; i <= n - 1; i++) {
101      for (int j = 1;j <= 10; j++) {
102        for (int k = 1; k <= 10; k++) {
103          if (j != k) {
104            if(i == n - 1 && j == 10 && k == 9) {
105              System.out.println("f_" + i + "_" + j + "_" + k + ";");
106            } else {
107              System.out.print("f_" + i + "_" + j + "_" + k + ",");
108            }
109          }
110        }
111      }
```

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

```
112     }
113   }
114 }
```

Below is my LP input file to solve the LP problem:

Min:110e_1_2+1058e_1_3+...+1261e_10_8+368e_10_9
(90 lines of the following type: ensure that e_a_b and the sum of f_i_a_b are the same for all cities a and b)
e_1_2=f_1_1_2+f_2_1_2+f_3_1_2+f_4_1_2
...
e_10_9=f_1_10_9+f_2_10_9+f_3_10_9+f_4_10_9;
(720 lines of the following type: ensure that f_{iab} = 1 if I visit city a at step i and city b at step i+1. Otherwise f_{iab} = 0)
3f_1_1_2<=1+x_1_1+x_2_2;
...
3f_4_10_9>=-1+x_4_10+x_5_9;
(5 lines of the following type: ensure that I visit exactly one city on each of 5 steps)
x_1_1+x_1_2+x_1_3+x_1_4+x_1_5+x_1_6+x_1_7+x_1_8+x_1_9+x_1_10=1;
...
x_5_1+x_5_2+x_5_3+x_5_4+x_5_5+x_5_6+x_5_7+x_5_8+x_5_9+x_5_10=1;
(10 lines of the following type: ensure that each city could be visited at most once)
x_1_1+x_2_1+x_3_1+x_4_1+x_5_1<=1;
…
x_1_10+x_2_10+x_3_10+x_4_10+x_5_10<=1;
(ensure all the variables are binary)
bin x_1_1,x_1_2, ..., f_4_10_8,f_4_10_9;

Below is the result I get from lp_solve after generating the LP input file above:

Actual values of the variables:

e_1_2	1
e_2_8	1
e_3_4	1
e_8_3	1
f_1_1_2	1
f_2_2_8	1
f_4_3_4	1
f_3_8_3	1
x_1_1	1
x_2_2	1
x_3_8	1
x_4_3	1
x_5_4	1

./lp_solve -ia hw3.txt 1.72s user 0.01s system 98% cpu 1.758 total
all other variables have a value of 0

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Conclusion:

According to the result I got from lp_solve, the shortest possible path that could connect 5 cities is 1--2--8--3--4 (Beijing -- Tianjin -- Jilin -- Harbin -- Daqing), and the total distance should be 1412km with an average distance of 1412/4 = 353km for an intermediate path.

I drew a line that could show my route for my solution. Check the graph below (the arrow shows the direction where my route starts at from and end in Daqing):

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Challenge 1:

I will change the cities I want to visit from 5 to 2 through 10. I repeat the steps I listed above and record the answers into a table (when the number of cities needed to be visited exceeds 7, the running time is too long, so I can't collect related data). Check the table below.

Number of cities I will visit	2	3	4	5	6	7	8	9	10
Total distance	110km	360km	1271km	1412km	2324km	3123km			
Path	1-2	4-3-8	1-2-8-3	1-2-8-3-4	4-3-8-2-1-5	9-10-8-3-4-1-2			
Running time	0.01s	0.18s	0.56s	1.72s	16.59s	2071.05s	over 2 hours	over 2 hours	over 2 hours

For n cities ($2 \leq n \leq 10$), there will be 10!/(10-n)! possibilities. This means total possibilities and running time will only be longer as the number of cities I want to visit increases.

We could see the solution for 3 cities and 2 cities are totally different: the distance between city 1 and city 2 is shortest but both of them are relatively far away from other cities, and the distance among cities 3, 4, and 8 are much shorter. This could also explain why the solutions from 2 cities and 3 cities are the base for the solutions for 4 cities and 5 cities.

City 5 appears in the solution for 6 cities but is replaced by city 9 and city 10 in the solution for 7 cities. A possible reason could be that the distance between cities 9 and 10 is short, but both of them are far away from other cities when city 5 is relatively closer to other cities.

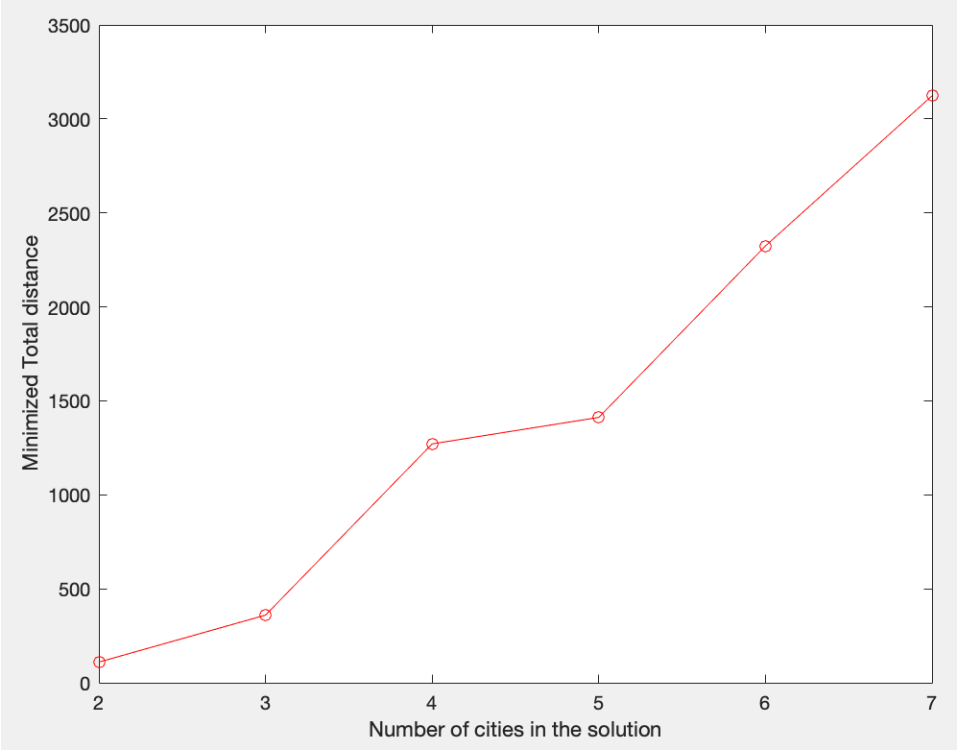
Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Plot for total distance versus the number of cities in my solution:



We could see that there is a positive relationship between the number of cities and minimized total distance in the solution. Yet, this phenomenon could be easily proved.

Define *the minimized total distance for the solution of n cities is a* and the minimized total distance for the solution of n+1 cities is b. Suppose there isn't such positive relationship, then $b \leq a$ must be true. Yet, we're able to pick n cities from the n+1 cities with a total distance of $c < b$, so $c < a$, which is a contradiction to the origin statement (Italic part).

So the relationship between the number of cities and minimized total distance must be positive.

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Challenge 2:

Now, I am going to assign a rating to each city (I will still visit 5 cities only).

Here is my rating for each city: Beijing -- 9, Tianjin -- 7.5, Harbin -- 8, Daqing -- 6, Xi'an -- 6.5, Kunming -- 7, Shanghai -- 8.5, Jilin -- 6, Tokyo -- 10, Kyoto -- 9.5.

Design an LP to solve this challenge:

The variables and related constraints are exactly the same as what I wrote at the end of page 2.

Since I want to maximize the sum of ratings for the cities i visit, my object function should be:

$\sum^{j=10}_{1}(\sum^{i=5}_{1}x_{ij})y_j$ where y_j represents the rating for city j.

I need to add an upper bound for total distance traveled, so I have the following new constraint:

$\sum^{10}_{a=1}\sum^{10}_{b=1}e_{ab}d_{ab} \leq 2000$ where 2000km is the current upper bound for total distance.

Thus, my LP is:

$$\text{Maximize } \sum^{j=10}_{1}(\sum^{i=5}_{1}x_{ij})y_j$$

Subject to

$$\sum^{10}_{j=1}x_{ij} = 1 \text{ for all } 1 \leq i \leq 5 \quad (1)$$

$$\sum^{5}_{i=1}x_{ij} \leq 1 \text{ for all } 1 \leq j \leq 10 \quad (2)$$

$$3f_{iab} \leq 1 + x_{ia} + x_{(i+1)b} \text{ for all } 1 \leq i \leq 4 \quad (3)$$

$$3f_{iab} \geq -1 + x_{ia} + x_{(i+1)b} \text{ for all } 1 \leq i \leq 4 \quad (4)$$

$$e_{ab} = \sum^{4}_{i=1}f_{iab} \text{ for all } 1 \leq a, b \leq 10 \quad (5)$$

$$\sum^{10}_{a=1}\sum^{10}_{b=1}e_{ab}d_{ab} \leq 2000 \quad (6)$$

$$x_{ij}, e_{ab}, f_{iab} \in [0, 1] \text{ for all } 1 \leq i \leq 5, 1 \leq a, b, i \leq 10 \quad (7)$$

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Coding:

After finishing to build my LP, I will use the LP solver to solve this LP problem.

I change the code to generate the lpsolve input file (delete all code from line 16 to 28 from the origin input file and add all except the last two lines of code below, then add the last two lines of the code below after line 3):

```
18    // This will assign ratings for all the cities I may visit.
19    double [] rating = {9, 7.5, 8, 6, 6.5, 7, 8.5, 6, 10, 9.5};
20    // Print out the object function of the problem that will maximize the total ratings among the cities I visit.
21    System.out.print("max: ");
22    for (int i = 1; i <= n; i++) {
23      for (int j = 1; j <= 10; j++) {
24        if (i == n && j == 10) {
25          System.out.println(rating[j-1] + "x_" + i + "_" + j + ";");
26        } else {
27          System.out.print(rating[j-1] + "x_" + i + "_" + j + "+");
28        }
29      }
30    }
31    // Print out the newly assigned constraint that will limit the total traveled distance.
32    System.out.print("distance=");
33    for (int i = 1; i <= 10; i++) {
34      for (int j = 1; j <= 10; j++) {
35        if (i != j) {
36          if (i == 10 && j == 9) {
37            System.out.println(distance[i-1][j-1] + "e_" + i + "_" + j + ";");
38          } else {
39            System.out.print(distance[i-1][j-1] + "e_" + i + "_" + j + "+");
40          }
41        }
42      }
43    }
44    System.out.println("distance<=" + m + ";");
 4  // This gives the upper bound for the number of distance I could travel in km.
 5  public static int m = 2000;
```

The following will be added after deleting the first line from the original LP input file:

max:
9.0x_1_1+7.5x_1_2+8.0x_1_3+6.0x_1_4+6.5x_1_5+7.0x_1_6+8.5x_1_7+6.0x_1_8+10.0x_1_9+9.5x_1_10+9.0x_2_1+7.5x_2_

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

2+8.0x_2_3+6.0x_2_4+6.5x_2_5+7.0x_2_6+8.5x_2_7+6.0x_2_8+10.0x_2_9+9.5x_2_10+9.0x_3_1+7.5x_3_2+8.0x_3_3+6.0x_3_4+6.5x_3_5+7.0x_3_6+8.5x_3_7+6.0x_3_8+10.0x_3_9+9.5x_3_10+9.0x_4_1+7.5x_4_2+8.0x_4_3+6.0x_4_4+6.5x_4_5+7.0x_4_6+8.5x_4_7+6.0x_4_8+10.0x_4_9+9.5x_4_10+9.0x_5_1+7.5x_5_2+8.0x_5_3+6.0x_5_4+6.5x_5_5+7.0x_5_6+8.5x_5_7+6.0x_5_8+10.0x_5_9+9.5x_5_10;
distance=
110e_1_2+1058e_1_3+1024e_1_4+912e_1_5+2090e_1_6+1080e_1_7+947e_1_8+2100e_1_9+1792e_1_10+110e_2_1+1069e_2_3+1055e_2_4+914e_2_5+2074e_2_6+972e_2_7+942e_2_8+2025e_2_9+1708e_2_10+1058e_3_1+1069e_3_2+141e_3_4+1970e_3_5+3143e_3_6+1691e_3_7+219e_3_8+1582e_3_9+1432e_3_10+1024e_4_1+1055e_4_2+141e_4_3+1932e_4_5+3123e_4_6+1739e_4_7+327e_4_8+1722e_4_9+1569e_4_10+912e_5_1+914e_5_2+1970e_5_3+1932e_5_4+1189e_5_6+1220e_5_7+1852e_5_8+2803e_5_9+2451e_5_10+2090e_6_1+2074e_6_2+3143e_6_3+3123e_6_4+1189e_6_5+1955e_6_7+3014e_6_8+3722e_6_9+3354e_6_10+1080e_7_1+972e_7_2+1691e_7_3+1739e_7_4+1220e_7_5+1955e_7_6+1473e_7_8+1765e_7_9+1396e_7_10+947e_8_1+942e_8_2+219e_8_3+327e_8_4+2851e_8_5+3014e_8_6+1473e_8_7+1447e_8_9+1261e_8_10+2100e_9_1+2025e_9_2+1582e_9_3+1722e_9_4+2803e_9_5+3722e_9_6+1765e_9_7+1447e_9_8+368e_9_10+1792e_10_1+1708e_10_2+1432e_10_3+1569e_10_4+2451e_10_5+3354e_10_6+1396e_10_7+1261e_10_8+368e_10_9;
distance<=2000;

Below is the result I get from lp_solve after generating the LP input file above:

Value of objective function: 39.50000000

Actual values of the variables:

x_1_4	1
x_2_3	1
x_3_8	1
x_4_10	1
x_5_9	1
distance	1989
e_3_8	1
e_4_3	1
e_8_10	1
e_10_9	1
f_2_3_8	1
f_1_4_3	1
f_3_8_10	1
f_4_10_9	1

./lp_solve -ia hw3_2000.txt 4.45s user 0.01s system 99% cpu 4.474 total
all other variables have a value of 0

The output shows that if I want to travel through 5 cities when the upper bound for total traveled distance is 2000km, the path 4-3-8-10-9 (Daqing - Harbin - Jilin - Kyoto - Tokyo) with the total distance of 1989km could maximize the sum of ratings, which is 39.5.

Math 381

Homework 4

Oct 29, 2021

Haochen Wang

I will change the upper bound for total traveled distance. I will increase the upper bound for total traveled distance from 1412km (since the minimum total traveled distance for traveling through 5 cities is 1412km, any upper bound below 1412 km will make the LP problem infeasible) to 4400km by 200km per step (188km for the first step only). I record the data with related results in a table. Check the table below for specific information.

Maximal Total Distance	Actual Total Distance	Sum of rating	Path	Running Time
1412km	1412km	36.5	1-2-8-3-4	3.04s
1600km	1412km	36.5	1-2-8-3-4	3.98s
1800km	1647km	36.5	1-2-3-4-8	5.17s
2000km	1989km	39.5	4-3-8-10-9	4.45s
2200km	1989km	39.5	9-10-8-3-4	5.72s
2400km	2346km	39.5	9-10-3-8-4	7.62s
2600km	2483km	39.5	3-8-4-10-9	9.55s
2800km	2681km	42	1-2-8-10-9	3.11s
3000km	2846km	44.5	1-2-7-10-9	0.41s
3200km	2846km	44.5	1-2-7-10-9	0.46s
3400km	2846km	44.5	1-2-7-10-9	0.58s
3600km	2846km	44.5	1-2-7-10-9	0.75s
3800km	2846km	44.5	1-2-7-10-9	0.54s
4000km	3902km	45	3-1-7-10-9	0.48s
4200km	4088km	45	7-1-3-9-10	0.4s
4400km	4271km	45	3-1-7-9-10	0.28s

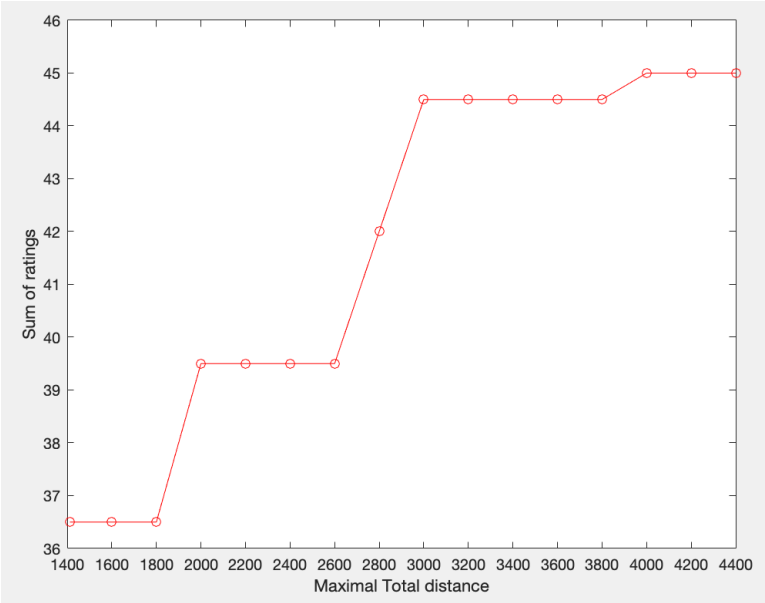
Math 381

Homework 4

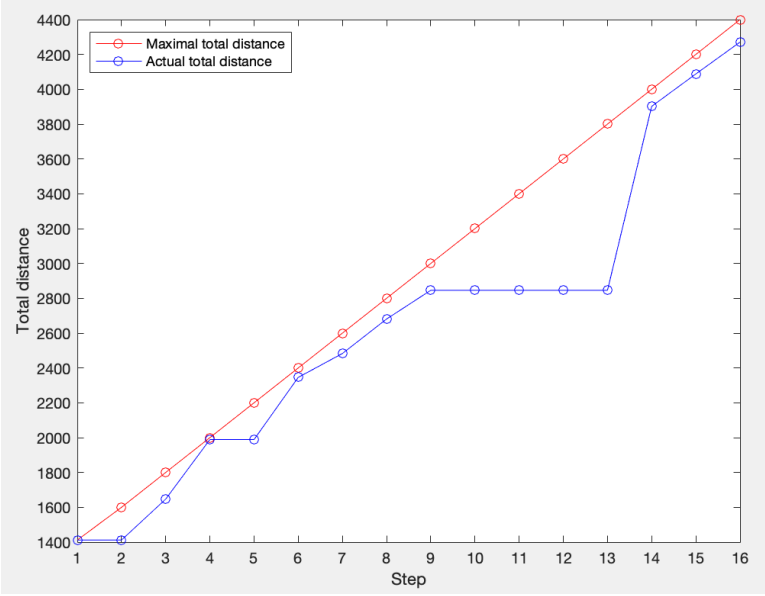
Oct 29, 2021

Haochen Wang

Below is the graph for Maximal total distance v.s. Sum of ratings:



Below is the graph for Maximal total distance v.s. Actual total distance by steps (16 total):



Math 381

Homework 4

Oct 29, 2021

Haochen Wang

Based on the table and two graphs above, I find some interesting facts:

- The sum of ratings will never decrease as maximal total distance increases.
- After the maximal total distance becomes no shorter than 4000km, the sum of ratings stays at 45. According to the rating I assigned, the five highest-rated cities have a rating of 10, 9.5, 9, 8.5, and 8 respectively. This means the maximal total rating is 45. Thus, I could enjoy the maximal total ratings if I could travel no smaller than 3902km (the minimum total distance with the highest possible total rating occurs when the maximal total distance becomes 4000km, which is 3902km).
- The relationship between running time and maximal traveled distance is neither strictly increasing nor strictly decreasing: when maximal total distance increase from 1412km to 2600km and from 3000km to 3600km, running time increases monotonically, but running time decreases monotonically when maximal total distance increases from 2600km to 3000km and from 3600km to 4400km.
- The solution has the same number of sum of rating and same cities in the path when maximal total distance changes from 1412km to 1800km, or from 2000km to 2600km, or from 3000km to 3800km, or from 4000km to 4400km.
- Actual total distance never decreases when maximum total distance increases.
- Solutions with the same number of sum of rating and same cities in the path could have different actual total distance due to the arrangement of cities in the path.