ISE 426 - Optimization Models and Applications

Class Project - Navigation Problem

Phase 3 – Final Report

Team Hydra

December 2019

Team Member

Jia Wo jiw416@lehigh.edu

Jingyu Liu jilb19@lehigh.edu

Qianfan Ji qij319@lehigh.edu

Tianjiu Yu tiy419@lehigh.edu

Hua Dong hud219@lehigh.edu

Outline

1. Executive Summary

Inspired by the navigation system for rideshare services like Uber and Lyft, we reformulated the navigation problem based on the shortest path problem. Meanwhile, we built a new model using greedy approach. The model we built aims to find the optimal sequence to send each customer to his or her desired destination and minimize the total distance travelled. The model based on the shortest path problem takes all nodes into consideration and find the total shortest distance. The model based on the greedy approach always serve the nearest customer first. Each time we add a new customer in the system, the driver will compare his location and the locations of all the unserved customers and search for the nearest one. As we're adding customers in later stage, the system elevated from a deterministic model to a dynamic model. Initially, we want to compare the result of the two methods. Since the two methods are based on different heuristic, the two models made different decision since the first stage. Overall the optimal solution model based on the shortest path problem is 52. Meanwhile, the overall the optimal solution model based on the greedy approach is 75. Basically, the model that considers all the nodes in the system yields the best result.

2.Introduction

In stage 0, there are one driver and three customers in the system. In later stages, we gradually add more customers in the system. In stage 1, there are one driver and four customers in the system. In stage 2, there are one driver and five customers in the system. The pickup locations and the drop-off locations for the three customers in stage 0, the starting-point and the end-point for the driver are all deterministic. We enforce constraints such as the customer must be picked up first then dropped off, and all three customers must finish their trip in the end. After picking a customer, the driver will take the customer to the destination immediately. After finishing all three trips, the driver will return to the end-point. In later stage, we give the models new information about the new customers' location and their desired destinations. The two models will make separate decisions based on the position of the driver and each of the remaining unserved customers in the system.

Below is a graph presentation of our problem:

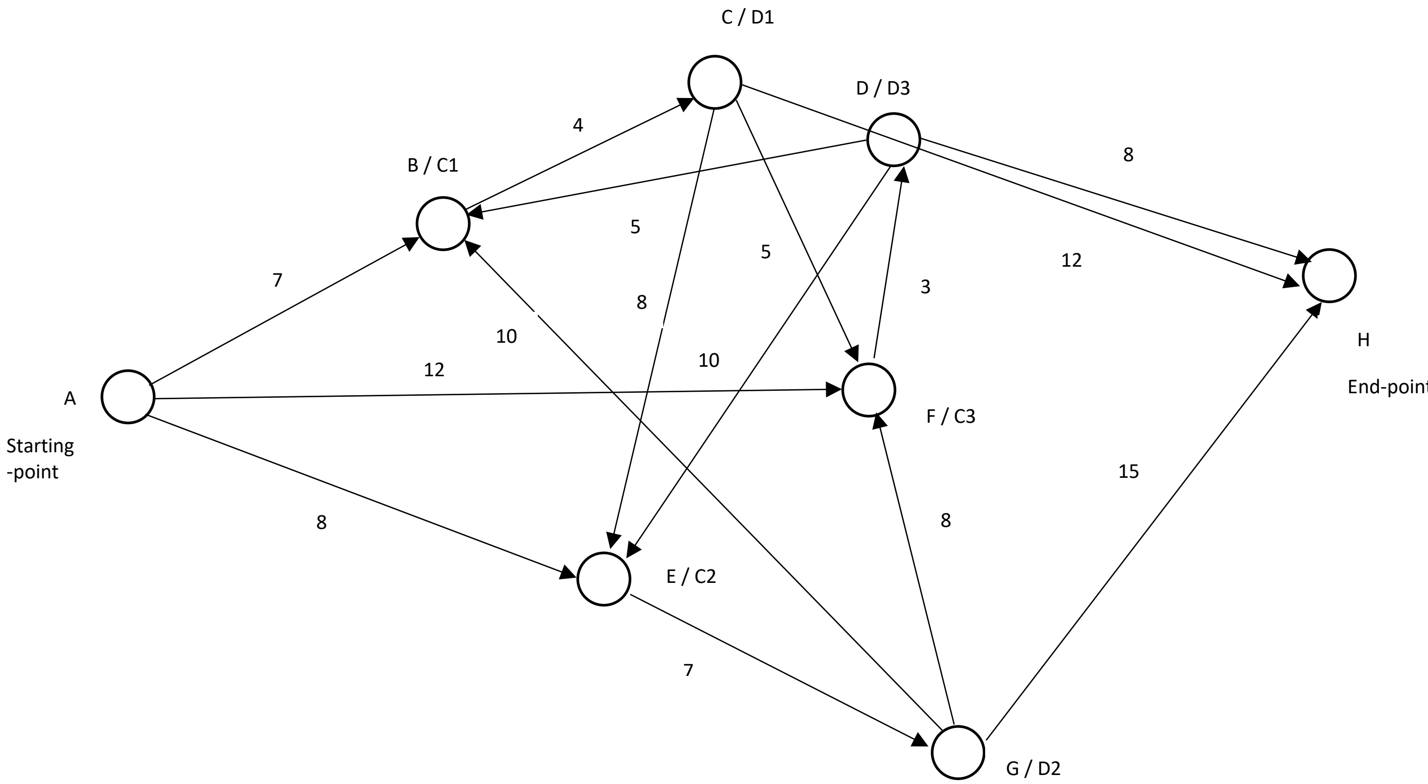

Figure 1 Graph Presentation of Stage 0

3.Model

3.1 Data

We generated the data using random number generators from MATLAB, including the distance from the driver's starting-point A to each customer, the distance from each customer C_i to his or her desired destination D_i, the distance from the destination D_i to the next customer C_i, and the distance from destination D_i to the end-point H. The distance we are using in the two models are all Euclidean distances.

3.2 Decision Variables

A: a set of nodes {A, B……, H}

X_{ij}: binary varible, 1 if travel on arc (i, j), 0 otherwise.

D_{ij}: the distance of arc (i, j).

3.3 Formulation

3.3.1 Objective Function: minimize the total distance travelled from A to H

$$\text{Min} \sum_{i,j \in \mathrm{A}} X_{ij} \cdot D_{ij}$$

3.3.2 Constraints:

Customer 1 must be picked up: $X_{\mathrm{ab}} + X_{\mathrm{db}} + X_{\mathrm{gb}} = 1;$

Customer 2 must be picked up: $X_{\mathrm{ae}} + X_{\mathrm{ce}} + X_{\mathrm{de}} = 1;$

Customer 3 must be picked up: $X_{\mathrm{gf}} + X_{\mathrm{cf}} + X_{\mathrm{af}} = 1;$

Driver must pick up a customer at the starting-point: $X_{\mathrm{ab}} + X_{\mathrm{ae}} + X_{\mathrm{af}} = 1;$

Customer 1 must be dropped off: $X_{\mathrm{bc}} = 1;$

Customer 2 must be dropped off: $X_{\mathrm{eg}} = 1;$

Customer 3 must be dropped off: $X_{\mathrm{fd}} = 1;$

After dropping off customer 1, the driver should either pick up customer 2 or customer 3 or go to the end-point: $X_{\mathrm{ce}} + X_{\mathrm{cf}} + X_{\mathrm{ch}} = 1;$

After dropping off customer 2, the driver should either pick up customer 1 or customer 3 or go to the end-point: $X_{\mathrm{gb}} + X_{\mathrm{gf}} + X_{\mathrm{gh}} = 1;$

After dropping off customer 3, the driver must pick up customer 1 or customer 2 or go to the end-point: $X_{\mathrm{db}} + X_{\mathrm{de}} + X_{\mathrm{dh}} = 1;$

The driver must return to the end-point after finishing all three trips: $X_{\mathrm{ch}} + X_{\mathrm{gh}} + X_{\mathrm{dh}} = 1.$

4. Solutions and Analysis

In stage 0, the optimal solution for the model based on the shortest path problem is picking up customer 2, then drop off customer 2. The decision made is based on minimizing the overall distance travelled. The model considered the distances from the driver to all the customers, from the destinations to the next customers and distances from the last destination to the end-point. The optimal solution for the model based on the greedy approach is picking up customer 1, then drop off customer 1. The model only considered the distances from the driver to all the customers.

In stage 1, the optimal solution for the model based on the shortest path problem is picking up customer 4, then drop off customer 4. Again, the model considered the distances from the driver to all the unserved customers, from the destinations to the next customers and distances from the last destination to the end-point. The optimal solution for the model based on the greedy approach is picking up customer 3, then drop off customer 3. Again, the model only considered the distances from the driver to all the unserved customers.

In stage 2, the optimal solution for the model based on the shortest path problem is picking up customer 1, then drop off customer 1; picking up customer 3, then drop off customer 3; picking up customer 5, then drop off customer 5; and eventually return to the end-point. The total distance travelled is 52. Again, the model considered the distances from the driver to all the unserved customers, from the destinations to the next customers and distances from the last destination to the end-point. The optimal solution for the model based on the greedy approach is picking up customer 2, then drop off customer 2; picking up customer 4, then drop off customer 4; picking up customer 5, then drop off customer 5; and eventually return to the end-point. The total distance travelled is 61. Again, the model only considered the distances from the driver to all the unserved customers.

As we mentioned earlier in the summary part, we want to compare the total distance travelled between two models. However, since they took different routes in the very beginning, the overall distance travelled is not comparable.

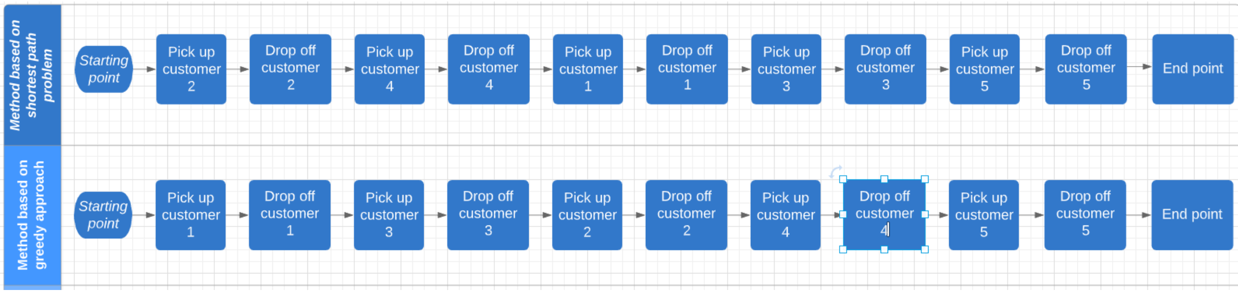

Figure 2 Graph Presentation of Optimal Solution

5. Conclusions

We formulated the navigation problem based on different heuristic. The shortest path problem considers the overall distances of the starting point, customers, destinations and the end point. However, the greedy approach only considers the distance between the driver and the customers. In our project, we elevated the problem from a deterministic model to a dynamic model by adding new customers in the system. We wrote the code using both AMPL and MATLAB. Again, it is difficult for us to compare the two approaches in terms of total distances travelled. But we can say that both approaches can solve real-world navigation problems given the conditions. In the future, we can further improve our model and add more customers and drivers in the system to make the model more dynamic.

6. Appendix

AMPL code

Stage2.mod:

```
var xab >= 0;
var xae >= 0;
var xaf >= 0;
var xbc >= 0;
var xce >= 0;
var xcf >= 0;
var xch >= 0;
var xdb >= 0;
var xde >= 0;
var xdh >= 0;
var xeg >= 0;
var xfd >= 0;
var xgb >= 0;
var xgf >= 0;
var xgh >= 0;
var xci >= 0;
var xdi >= 0;
var xgi >= 0;
var xij >= 0;
var xjb >= 0;
var xjf >= 0;
var xjh >= 0;
var xjk >= 0;
var xkl >= 0;
var xlb >= 0;
var xlf >= 0;
var xlh >= 0;
var xck >= 0;
var xdk >= 0;
param dab >= 0;
param dae >= 0;
param daf >= 0;
param dbc >= 0;
param dce >= 0;
param dcf >= 0;
param dch >= 0;
param ddb >= 0;
param dde >= 0;
param ddh >= 0;
```

```
param deg >= 0;
param dfd >= 0;
param dgb >= 0;
param dgf >= 0;
param dgh >= 0;
param dci >= 0;
param ddi >= 0;
param dgi >= 0;
param dij >= 0;
param djb >= 0;
param djf >= 0;
param djh >= 0;
param djk >= 0;
param dkl >= 0;
param dlb >= 0;
param dlf >= 0;
param dlh >= 0;
param dck >= 0;
param ddk >= 0;
minimize distance:
dab*xab+dae*xae+daf*xaf+dbc*xbc+dce*xce+dcf*xcf+dch*xch+ddb*xdb+dde*xde+ddh*xd
h+deg*xeg+dfd*xfd+dgb*xgb+dgf*xgf+dgh*xgh+dci*xci+ddi*xdi+dgi*xgi+dij*xij+djb*xjb+dj
f*xjf+djh*xjh+djk*xjk+dkl*xkl+dlb*xlb+dlf*xlf+dlh*xlh+dck*xck+ddk*xdk;
cons1: xjb+xjf+xjk = 1;
cons2: xdh+xch+xlh = 1;
cons3: xlb+xdb+xjb = 1;
cons4: xlf+xcf+xjf = 1;
cons5: xjk+xck+xdk = 1;
cons6: xlb+xlf+xlh = 1;
cons7: xdb+xdk+xdh = 1;
cons8: xcf+xck+xch = 1;
cons9: xae = 1;
cons10: xeg = 1;
cons11: xgi = 1;
cons12: xij = 1;
cons13: xbc = 1;
cons14: xfd = 1;
cons15: xkl = 1;
```

Stage2.dat:

```
param dab:= 7;
param dae:= 8;
param daf:= 12;
param dbc:= 4;
param dce:= 8;
```

```
param dcf:= 5;
param dch:= 12;
param ddb:= 5;
param dde:= 10;
param ddh:= 8;
param deg:= 7;
param dfd:= 3;
param dgb:= 10;
param dgf:= 8;
param dgh:= 15;
param dci:= 10;
param ddi:= 12;
param dgi:= 3;
param dij:= 5;
param djb:= 5;
param djf:= 10;
param djh:= 15;
param djk:= 9;
param dck:= 8;
param ddk:= 4;
param dkl:= 5;
param dlb:= 12;
param dlf:= 8;
param dlh:= 3;
```

Stage2.log:

```
ampl: model stage2.mod;
ampl: data stage2.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 12.9.0.0: optimal solution; objective 52
2 dual simplex iterations (0 in phase I)
ampl:
display xab,xae,xaf,xbc,xce,xcf,xdb,xde,xdh,xeg,xfd,xgb,xgf,xgh,xci,xdi,xgi,xij,xjb,xjf,xjh,xjk,xck,xdk,xkl,xlb,xlf,xlh,distance;
xab = 0
xae = 1
xaf = 0
xbc = 1
xce = 0
xcf = 1
xdb = 0
xde = 0
xdh = 0
xeg = 1
```

xfd = 1
xgb = 0
xgf = 0
xgh = 0
xci = 0
xdi = 0
xgi = 1
xij = 1
xjb = 1
xjf = 0
xjh = 0
xjk = 0
xck = 0
xdk = 1
xkl = 1
xlb = 0
xlf = 0
xlh = 1
distance = 52

MATLAB code

PROJECT1.m

```
clc,clear
Costume = [7 8 12];
S=min(Costume);
F1position=find(Costume==(min(Costume)));
 H=[4 10 5;8 7 10;5 8 6];
 D=[0,0,0];
 M=0;
 T=S;
 y=1;
  H
 for i=1:4
T=M+H(F1position,F1position)+T
H(F1position,F1position)=999;

fprintf('Picking and sending custome %d\n',F1position);
 prompt = ('Is any new costume appears Y/N \n');
 str=input(prompt,'s');
 if (strcmp(str, 'N') ==1)
     D=H(:,F1position)
     H(F1position,:)=999;
     H(:,F1position)=999;
     M=min(D)
     F1position=find(D==(min(D)));
H
 else
     p1 = 'what is the distance\n';
     c = input(p1);
     H=[H c];
     p2 = 'what is the distance\n';
     r=input(p2);
     H=[H;r]
     D=H(:,F1position)
     H(F1position,:)=999;
     H(:,F1position)=999;
     M=min(D)
     H
     F1position=find(D==(min(D)));
     y=y+1;
 end
 end
 T=M+H(F1position,F1position)+T
 F1position=find(D==(min(D)))
 T=M+H(F1position,F1position)+T;
 fprintf('Picking and sending custome %d\n',F1position);
  fprintf('total distance %d\n',T);
```

Figure 3 MATLAB Code

Matlab Ouput:

```
Command Window

H =

     4    10     5
     8     7    10
     5     8     3


T =

    11

Picking and sending custome 1
Is any new costume appears Y/N
Y
what is the distance
[999;5;10]
what is the distance
[999 6 12 5]

H =

   999    10     5   999
     8     7    10     5
     5     8     3    10
   999     6    12     5


D =

   999
     8
     5
   999


M =

     5


H =

   999   999   999   999
   999     7    10     5
   999     8     3    10
   999     6    12     5


T =

    19

Picking and sending custome 3
Is any new costume appears Y/N
Y
what is the distance
[999;6;999;8]
what is the distance
[999 10 999 9 5]
```

Figure 4 MATLAB Output 1

```
H =

   999   999   999   999   999
   999     7    10     5     6
   999     8   999    10   999
   999     6    12     5     8
   999    10   999     9     5


D =

   999
    10
   999
    12
   999


M =

    10


H =

   999   999   999   999   999
   999     7   999     5     6
   999   999   999   999   999
   999     6   999     5     8
   999    10   999     9     5


T =

    36

Picking and sending custome 2
Is any new costume appears Y/N
N

D =

   999
   999
   999
     6
    10


M =

     6


H =

   999   999   999   999   999
   999   999   999   999   999
   999   999   999   999   999
   999   999   999     5     8
   999   999   999     9     5
```

Figure 5 MATLAB Output 2

```
T =

    47

Picking and sending custome 4
Is any new costume appears Y/N
N

D =

   999
   999
   999
   999
     9


M =

     9


H =

   999   999   999   999   999
   999   999   999   999   999
   999   999   999   999   999
   999   999   999   999   999
   999   999   999   999     5


T =

    61


F1position =

     5

Picking and sending custome 5
total distance 75
```

Figure 6 MATLAB Output 3

7. Citation

[1] Medium. (2019). *Greedy Algorithm and Dynamic Programming*. [online] Available at: https://medium.com/cracking-the-data-science-interview/greedy-algorithm-and-dynamic-programming-a8c019928405 [Accessed 14 Dec. 2019].

[2] Hillier, F. and Lieberman, G. (2015). *Introduction to operations research*. New York, NY: McGraw-Hill.