Executive Summary of Data Science for e-Retail and the Sharing Economy Project
Professor Huseyin Topaloglu

Authors:
Evan Teng (et333@cornell.edu)
Negar Akbarzadeh (na332@cornell.edu)
19 May, 2022

Introduction

This report presents an integer programming model and the solution for a specific facility location problem where an online retailer is planning the locations of 3 fulfillment centers that will be used over the next 20 years. In this report, we will present the problem, describe an integer programming model and formulation, implement this model in the Gurobi Python interface, and compute the optimal solution.

Problem Description

The problem that we solved is about an online retailer who wants to plan the locations of 3 fulfillment centers (FCs) that will be used over the next 20 years. The first FC will be opened immediately and will be the only operational FC for the next 6 years. After 6 years, a second FC will be opened and the retailer will operate the two FCs concurrently for another 5 years. After a total of 11 years, a third FC will be opened and all three FCs will be operational for another 9 years. The retailer would like to choose the locations of these three FCs. In table 1, the x and y coordinates of the demand points (DPs) that the online retailer serves is presented. There are 20 DPs and each DP is served by the closest open FC. The yearly cost of serving a DP from a certain FC is simply the Euclidean distance between the DP and the FC. We would like to come up with a way to choose the locations of the first, second, and third FCs such that the total cost over the next 20 years is minimized.

$$\text{Euclidean distance between the DP and the FC} = \sqrt{(y_{FC} - y_{DP})^2 + (x_{FC} - x_{FC})^2}$$

Table 1 in the Appendix shows the x and y coordinates of the demand points, whereas Table 2 shows the x and y coordinates of the possible facility locations. There are in total 20 demand points, and 10 possible locations for facilities..

Solution

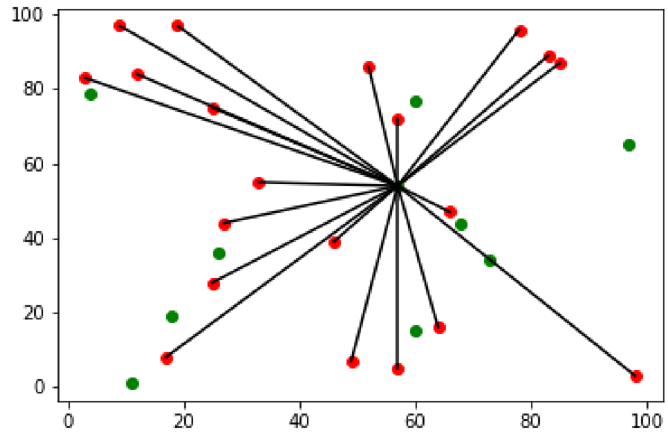
After using Gurobi to optimize our model, we found that the best facility to open was facility 3 (index starting at 0). The distances and therefore costs per year that are related to opening this fulfilment center would be:

{\$38.28, \$47.68, \$60.96, \$54.08, \$61.29, \$49.00, \$18.60, \$43.60, \$46.96, \$31.62, \$38.64, \$32.39, \$18.00,

$24.02, $11.40, $41.23, $64.44, $43.28, $65.44, $57.38}

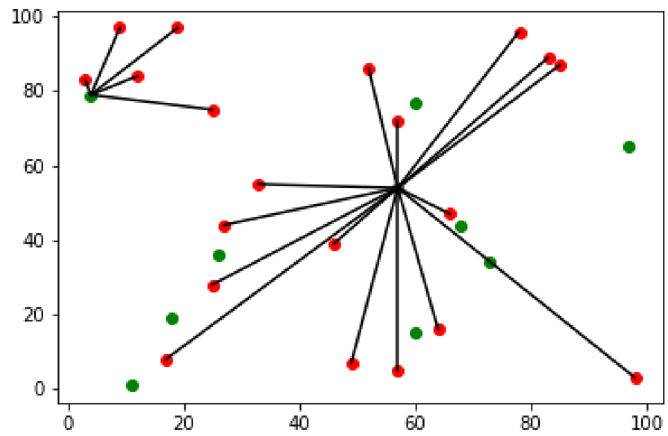
This gives us a total cost over 6 years of: 6*$848.29 = $5089.74

The following figure shows the connections between the opened facility and all the 20 different customer demand points, where the location of the possible fulfillment centers are shown with green dots and the demand points are represented by the red dots.



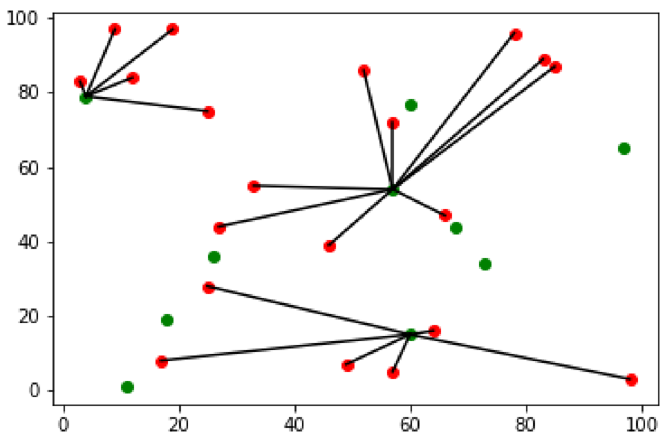
The optimal solution for the following 5 years is opening facilities y_3 and y_9. The graph below shows that some of the customers who were served by fulfillment center 3 in the first 6 years will now be served by fulfillment center 9 in the following 5 years. By adding the constraint $y_3 = 1$ to the original formulation, we can guarantee that facility center 3 in the second phase will still be opened. For phase 2, fulfillment center 3 will serve demand points 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17 and 18, and the newly opened fulfillment center 9 will serve demand points 0, 3, 4, 16 and 19.

The following graph demonstrates the results of phase 2 :



The total cost serving customers with fulfillment center 3 in the second phase is 5*$572.81 =$2864.05 and the total cost serving customers with fulfillment center 9 in the second phase is 5*$77.04 = $385.2 The total cost for these years would be $2864.05 + $385.2 = $3249.25

Finally, the last phase's optimal solution is shown below:



The optimal solution for this phase is opening facilities 0, 3, 9 . Facility 0 is paired with customers 1, 2, 5, 10, 15, 18, while facility 3 is paired with customers 6, 7, 8, 9, 11, 12, 13, 14, 17 , and facility 9 is paired with customers 0, 3, 4, 16, 19.

The total cost for this phase would be :

9*(Total cost serving by FC0 +Total cost serving by FC3 +Total cost serving by FC9) = 9*($ 148.92 +$ 269.87 + $77.05) = 9 * $495.83 = $4462.52

Therefore, the total cost over these 20 years would be the sum of total cost in phase 1, phase 2 and phase 3 which is equal to $5089.74 + $3249.25 + $4462.52 = $12801.51

Conclusions

Overall, in the first phase, facility 3 at location [57,54] should be opened, costing $5089.74 over 6 years. Then, facility 9 at location [4,79] should be opened for the next 5 years, costing $3249.25 in total to run both facilities. Finally, facility 0 at location [60,15] should be opened for the next 9 years, costing $4462.52 to run all 3 facilities. By the end of the 20 years, the total cost will be $12801.51.

Appendix

Model and formulation

Sets and Indices:

$i \in I$: Index and set of demand points.
$j \in J$: Index and set of fulfillment centers.

Parameters:

$c_{i,j} \in \mathbb{R}^+$: Distance and subsequent yearly cost between fulfillment center $j \in J$ and demand point $i \in I$.

Decision Variables:

$X_{i,j} \in \{0,1\}$: This variable is equal to 1 if demand point i is serviced by fulfillment center j ; and 0 otherwise.
$Y_j \in \{0,1\}$: This variable is equal to 1 if fulfillment center j is open; and 0 otherwise

Objective Function:

Minimizing Total cost :

$$\text{Min} \quad Z = \sum_{j \in J} \sum_{i \in I} X_{i,j} \cdot c_{i,j} \tag{1}$$

Constraints:

Demand: For each demand point $i \in I$ ensure that it is serviced by a fulfillment center.

$$\sum_{j \in J} X_{i,j} = 1 \quad \forall i \in I \tag{2}$$

Fulfillment Center limit: The sum of y_js constrained to 1 for the first 6 years, 2 for the next 5 years, and 3 for the last 9 years.
Constraint for the first 6 years:

$$\sum_{j \in J} Y_j = 1 \quad \forall j \in J \tag{3}$$

Constraint for the next 5 years:

$$\sum_{j \in J} Y_j = 2 \quad \forall j \in J \tag{4}$$

Constraint for maintaining the first opened fulfillment center:

$$Y_3 = 1 \tag{5}$$

where Y3 is the first opened fulfillment center

And constraint for the last 9 years:

$$\sum_{j \in J} Y_j = 3 \quad \forall j \in J \tag{6}$$

Constraint for maintaining the first two opened fulfillment centers:

$$Y_3 = 1 \tag{7}$$

where Y3 is the first opened fulfillment center

$$Y_9 = 1 \tag{8}$$

where Y9 is the second opened fulfillment center

In addition to:

$$X_{ij} \leq Y_j \quad \forall j \in J, \quad \forall i \in I \tag{9}$$

Formulation:

Phase 1 (for first 6 years):
Min $\quad Z_1 = 6(\sum_{j\in J}\sum_{i\in I} X_{i,j}\cdot c_{i,j})$
s.t.

$$\sum_{j\in J} Y_j = 1 \quad \forall j\in J$$

$$X_{ij} \le Y_j \quad \forall j\in J, \quad \forall i\in I$$

$$\sum_{j\in J} X_{i,j} = 1 \quad \forall i\in I$$

$$X_{ij},\, Y_j \in \{0,1\} \qquad \forall i\in I, j\in J$$

Phase 2 (for the next 5 years):
Min $\quad Z_2 = 5(\sum_{j\in J}\sum_{i\in I} X_{i,j}\cdot c_{i,j})$
s.t.

$$Y_3 = 1$$

$$\sum_{j\in J} Y_j = 2 \quad \forall j\in J$$

$$X_{ij} \le Y_j \quad \forall j\in J, \quad \forall i\in I$$

$$\sum_{j\in J} X_{i,j} = 1 \quad \forall i\in I$$

$$X_{ij},\, Y_j \in \{0,1\} \qquad \forall i\in I, j\in J$$

Phase 3 (for the last 9 years):
Min $\quad Z_3 = 9(\sum_{j\in J}\sum_{i\in I} X_{i,j}\cdot c_{i,j})$
s.t.

$$Y_3 = 1$$

$$Y_9 = 1$$

$$\sum_{j\in J} Y_j = 3 \quad \forall j\in J$$

$$X_{ij} \le Y_j \quad \forall j\in J, \quad \forall i\in I$$

$$\sum_{j\in J} X_{i,j} = 1 \quad \forall i\in I$$

$$X_{ij},\, Y_j \in \{0,1\} \qquad \forall i\in I, j\in J$$

The total cost over these 20 years will be calculated by adding total cost in the first 6 years, the following 5 years and the last 9 years. ($Z_1 + Z_2 + Z_3$)

Tables

Table1. Coordinate of Demand Points (DP)

	x	y
DP 0	25	75
DP 1	49	7
DP 2	17	8
DP 3	12	84
DP 4	3	83
DP 5	57	5
DP 6	46	39
DP 7	83	89
DP 8	78	96
DP 9	27	44
DP 10	64	16
DP 11	52	86
DP 12	57	72
DP 13	33	55
DP 14	66	47
DP 15	25	28
DP 16	9	97
DP 17	85	87
DP 18	98	3
DP 19	19	97

Table2. Coordinate of locations for the Fulfillment Centers (FC)

	x	y
FC0	60	15
FC1	26	36
FC2	73	34
FC3	57	54
FC4	18	19
FC5	11	1
FC6	60	77
FC7	68	44
FC8	97	65
FC9	4	79

Table3. Yearly costs/distances between points

	FC0	FC1	FC2	FC3	FC4	FC5	FC6	FC7	FC8	FC9
DP0	69.462220	39.012818	63.126856	38.275318	56.435804	75.312682	35.057096	53.009433	72.691127	21.377558
DP1	13.601471	37.013511	36.124784	47.675990	33.241540	38.470768	70.859015	41.593269	75.286121	84.905830
DP2	43.566042	29.410882	61.741396	60.959003	11.045361	9.219544	81.301906	62.425956	98.229324	72.180330
DP3	84.053554	50.000000	78.873316	54.083269	65.276336	83.006024	48.507731	68.818602	87.097646	9.433981
DP4	88.729927	52.325902	85.445889	61.294372	65.734314	82.389320	57.314920	75.802375	95.707889	4.123106
DP5	10.440307	43.840620	33.120990	49.000000	41.436699	46.173586	72.062473	40.521599	72.111026	91.021975
DP6	27.784888	20.223748	27.459060	18.601075	34.409301	51.662365	40.496913	22.561028	57.245087	58.000000
DP7	77.491935	77.833155	55.901699	43.600459	95.524866	113.701363	25.942244	47.434165	27.784888	79.630396
DP8	82.975900	79.397733	62.201286	46.957428	97.616597	116.249731	26.172505	52.952809	36.359318	75.927597
DP9	43.931765	8.062258	47.074409	31.622777	26.570661	45.880279	46.669048	41.000000	73.082146	41.880783
DP10	4.123106	42.941821	20.124612	38.639358	46.097722	55.081757	61.131007	28.284271	59.076222	87.000000
DP11	71.449283	56.356011	56.080300	32.388269	75.133215	94.371606	12.041595	44.944410	49.658836	48.507731
DP12	57.078893	47.507894	41.231056	18.000000	65.802736	84.599054	5.830952	30.083218	40.607881	53.460266
DP13	48.259714	20.248457	45.177428	24.020824	39.000000	58.309519	34.828150	36.687873	64.776539	37.643060
DP14	32.557641	41.484937	14.764823	11.401754	55.569776	71.700767	30.594117	3.605551	35.846897	69.771054
DP15	37.336309	8.062258	48.373546	41.231056	11.401754	30.413813	60.216277	45.880279	80.950602	55.154329
DP16	96.566040	63.324561	89.805345	64.443774	78.517514	96.020831	54.781384	79.309520	93.637599	18.681542
DP17	76.216796	77.987178	54.341513	43.278170	95.462034	113.454837	26.925824	46.238512	25.059928	81.394103
DP18	39.849718	79.202273	39.824616	65.436993	81.584312	87.022985	83.186537	50.803543	62.008064	120.880106
DP19	91.678787	61.400326	82.975900	57.384667	78.006410	96.332757	45.617979	72.180330	84.308956	23.430749
Total distances	1097.154295	935.636344	1043.768826	848.294556	1153.866952	1449.373588	919.537672	944.136744	1291.526095	1134.404497

Python code:

```
%matplotlib inline
!pip install gurobipy

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sys
import gurobipy as gp
from gurobipy import GRB
import math
from itertools import product

def read_data():
    global df1,df2
    xls = pd.ExcelFile('data.xlsx')
    DP = pd.read_excel(xls, 'DPs')
    FC = pd.read_excel(xls, 'FCs')
    return DP, FC

def dist(c_i, c_j):
    return np.linalg.norm(c_i - c_j)

DP,FC = read_data()
num_FC = len(FC) #10
num_DP = len(DP) #20
cartesian_prod = list(product(range(num_DP), range(num_FC)))

#distances bn demand point d and fulfillment center f
yearly_cost = {(d,f): dist(DP.values[d], FC.values[f]) for d, f in cartesian_prod}

m_1 = gp.Model('facility_location_fist6year')

def constructVars():
    global m_1, num_DP, num_FC , x_vars , y_vars
    for i in range( num_DP ):
        for j in range( num_FC ):
            var = m_1.addVar( vtype = GRB.BINARY , name = "x_" + str( i ) + "_" + str( j ) )
            x_vars[ i , j ] = var
    m_1.update()
    for j in range( num_FC ):
        var_2 = m_1.addVar( vtype = GRB.BINARY , name = "y_" + str( j ))
        y_vars[ j ] = var_2
    m_1.update()

def contructConstrs_6years():
    global m_1, num_DP, num_FC , x_vars , y_vars
    for i in range( num_DP ):
        constExpr_0 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_0 += 1.0 * x_vars[ i,j ]
        m_1.addConstr( lhs = constExpr_0 , sense = GRB.EQUAL , rhs = 1 )
    m_1.update()
    for i in range( num_DP ):
        constExpr_1_1 = gp.LinExpr()
        constExpr_1_2 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_1_1 = 1.0 * x_vars[i,j]
            constExpr_1_2 = 1.0 * y_vars[j]
            m_1.addConstr(lhs= constExpr_1_1, sense = GRB.LESS_EQUAL , rhs = constExpr_1_2)
```

```
    m_1.update()

    constExpr_1 = gp.LinExpr()
    for j in range( num_FC ):
        constExpr_1 += 1.0 * y_vars[ j ]
    m_1.addConstr( lhs = constExpr_1 , sense = GRB.EQUAL , rhs = 1 )
    m_1.update()

def constructObj():
    global m_1, num_DP, num_FC , x_vars , y_vars
    objExpr = gp.LinExpr()
    for i in range(num_DP):
        for j in range( num_FC ):
            objExpr += (x_vars[i,j] * yearly_cost[i,j])
        objExpr =6*objExpr
        m_1.setObjective( objExpr , GRB.MINIMIZE )
    m_1.update()

m_1.Params.Presolve = 0
x_vars = {}
y_vars = {}
for i in range(num_DP):
    for j in range(num_FC):
        x_vars[i,j] = 0
for i in range(num_DP):
    y_vars [i] = 0
constructVars()
contructConstrs_6years()
constructObj()
m_1.write('test_1.lp')
m_1.optimize()
m_1.printAttr('Xn')



m_2 = gp.Model('facility_location_11year')
def constructVars():
    global m_2, num_DP, num_FC , x_vars , y_vars
    for i in range( num_DP ):
        for j in range( num_FC ):
            var = m_2.addVar( vtype = GRB.BINARY , name = "x_" + str( i ) + "_" + str( j ) )
            x_vars[ i , j ] = var
        m_2.update()
    for j in range( num_FC ):
        var_2 = m_2.addVar( vtype = GRB.BINARY , name = "y_" + str( j ))
        y_vars[ j ] = var_2
    m_2.update()

def contructConstrs_11years():
    global m_2, num_DP, num_FC , x_vars , y_vars

    m_2.addConstr( lhs = y_vars[3] , sense = GRB.EQUAL , rhs = 1 )
    m_2.update()

    for i in range( num_DP ):
        constExpr_0 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_0 += 1.0 * x_vars[ i,j ]
        m_2.addConstr( lhs = constExpr_0 , sense = GRB.EQUAL , rhs = 1 )
    m_2.update()
    for i in range( num_DP ):
        constExpr_1_1 = gp.LinExpr()
        constExpr_1_2 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_1_1 = 1.0 * x_vars[i,j]
            constExpr_1_2 = 1.0 * y_vars[j]
            m_2.addConstr(lhs= constExpr_1_1, sense = GRB.LESS_EQUAL , rhs = constExpr_1_2)
    m_2.update()
    constExpr_1 = gp.LinExpr()
    for j in range( num_FC ):
        constExpr_1 += 1.0 * y_vars[ j ]
    m_2.addConstr( lhs = constExpr_1 , sense = GRB.EQUAL , rhs = 2 )
    m_2.update()


def constructObj():
    global m_2, num_DP, num_FC , x_vars , y_vars
    objExpr = gp.LinExpr()
    for i in range(num_DP):
        for j in range( num_FC ):
            objExpr += (x_vars[i,j] * yearly_cost[i,j])
        objExpr =5*objExpr
        m_2.setObjective( objExpr , GRB.MINIMIZE )
    m_2.update()

m_2.Params.Presolve = 0
x_vars = {}
y_vars = {}
constructVars()
contructConstrs_11years()
constructObj()
```

```
m_2.write('test_2.lp')
m_2.optimize()
m_2.printAttr('Xn')

m_3 = gp.Model('facility_location_20year')
def constructVars():
    global m_3, num_DP, num_FC , x_vars , y_vars
    for i in range( num_DP ):
        for j in range( num_FC ):
            var = m_3.addVar( vtype = GRB.BINARY , name = "x_" + str( i ) + "_" + str( j ) )
            x_vars[ i , j ] = var
        m_3.update()
    for j in range( num_FC ):
        var_2 = m_3.addVar( vtype = GRB.BINARY , name = "y_" + str( j ))
        y_vars[ j ] = var_2
    m_3.update()

def contructConstrs_20years():
    global m_3, num_DP, num_FC , x_vars , y_vars

    m_3.addConstr( lhs = y_vars[3] , sense = GRB.EQUAL , rhs = 1 )
    m_3.update()

    m_3.addConstr( lhs = y_vars[9] , sense = GRB.EQUAL , rhs = 1 )
    m_3.update()

    for i in range( num_DP ):
        constExpr_0 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_0 += 1.0 * x_vars[ i,j ]
        m_3.addConstr( lhs = constExpr_0 , sense = GRB.EQUAL , rhs = 1 )
    m_3.update()
    for i in range( num_DP ):
        constExpr_1_1 = gp.LinExpr()
        constExpr_1_2 = gp.LinExpr()
        for j in range( num_FC ):
            constExpr_1_1 = 1.0 * x_vars[i,j]
            constExpr_1_2 = 1.0 * y_vars[j]
            m_3.addConstr(lhs= constExpr_1_1, sense = GRB.LESS_EQUAL , rhs = constExpr_1_2)
    m_3.update()
    constExpr_1 = gp.LinExpr()
    for j in range( num_FC ):
        constExpr_1 += 1.0 * y_vars[ j ]
    m_3.addConstr( lhs = constExpr_1 , sense = GRB.EQUAL , rhs = 3 )
    m_3.update()


def constructObj():
    global m_3, num_DP, num_FC , x_vars , y_vars
    objExpr = gp.LinExpr()
    for i in range(num_DP):
        for j in range( num_FC ):
            objExpr += (x_vars[i,j] * yearly_cost[i,j])
        objExpr =9*objExpr
        m_3.setObjective( objExpr , GRB.MINIMIZE )
    m_3.update()

m_3.Params.Presolve = 0
x_vars = {}
y_vars = {}
constructVars()
contructConstrs_20years()
constructObj()
m_3.write('test_3.lp')
m_3.optimize()
m_3.printAttr('Xn')
```

Python code for plotting the result:

This plot is for phase 1 and the rest are the same.

```
# this array includes which fulfillment center each demand point is connected to
# for this example, dem pnt 0 is connected to full cen 1, but dem pnt 8 is connected to full cen 6
assgns = [ 0 for j in range ( nodps ) ]
assgns[0] = 3
assgns[1] = 3
assgns[2] = 3
assgns[3] = 3
assgns[4] = 3
assgns[5] = 3
assgns[6] = 3
assgns[7] = 3
assgns[8] = 3
assgns[9] = 3
assgns[10] = 3
```

```
assgns[11] = 3
assgns[12] = 3
assgns[13] = 3
assgns[14] = 3
assgns[15] = 3
assgns[16] = 3
assgns[17] = 3
assgns[18] = 3
assgns[19] = 3

for fc in range( nofcs ):
    plt.plot( fcs[ fc ][ 0 ]  , fcs[ fc ][ 1 ] , 'ro' , color = "green" , lw = 9 )

for dp in range( nodps ):
    plt.plot( dps[ dp ][ 0 ]  , dps[ dp ][ 1 ] , 'ro' , color = "red" , lw = 9 )

for dp in range( nodps ):
    dpx = dps[ dp ][ 0 ]
    dpy = dps[ dp ][ 1 ]
    fcx = fcs[ assgns[ dp ] ][ 0 ]
    fcy = fcs[ assgns[ dp ] ][ 1 ]
    plt.plot( [ dpx , fcx ], [ dpy , fcy ]  , color = "black"  )

plt.show()
```

Acknowledgements

Academic integrity is expected of all students of Cornell University at all times, whether in the presence or absence of members of the faculty. Understanding this, We declare that all of the team members contributed equally to all stages of this project.”