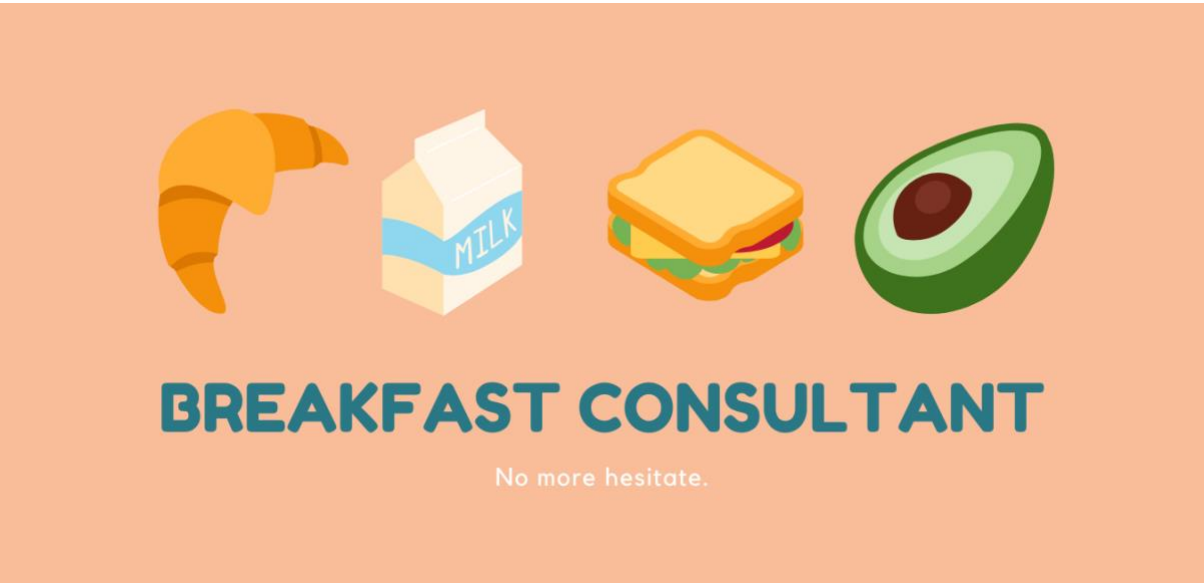
Operation Research Project Report

Breakfast Consultant



b05501012 薛功囷　b05501053 施欣承

b05501089 黃欣穎　b05501095 林永安

b05501108 陳妤華　b05501115 劉燕宜

b04208002 陳思彤

TABLE OF CONTENTS

Introduction

The topic that we have chosen for the Operation Research Final project is a model where it can tell you what to eat for breakfast on that given day. As we all know, breakfast is the most important meal of the day, and is a crucial one so you can start your day energized. Sometimes, choosing what to eat for breakfast might be difficult for many people, including students, teachers, professional workers, etc. There are many things that need to be considered when choosing what to eat for breakfast. Factors that need to be considered are the amount of money one is willing to spend for breakfast, the location of the breakfast store, how far it is from where they currently live and how far it is from the restaurant/store to their class location, the weather, and many more. That is why we decided to create this model, to solve your morning worries.

Literature Review

For this model, we would take into account, we would be using methods that we have learned throughout the whole semester. We would take into account what the user inputted and choose the available option from the restaurants we have carefully selected. What the user input to the model would be the constraints for our model. For example, if the user would input a budget of 60NTD for breakfast, automatically, all the breakfast options that are above 60 NTD would be eliminated for the solution. If the user inputted the maximum calorie intake is 400 calories, then all breakfast food above 400 calories will be removed for the solutions. Once all the user input has been made to become constraints, we would use GUROBI solver to solve for the best option. For this model, we will have three different modes, celebration mode, diet mode, and lazy mode.

The objective function of the whole model for the celebration and diet mode is listed in the formula below.

max. $\Sigma H_{ij}y_{ij}$, where H_{ij} represents the degree of appropriateness (某店某餐爽度) and y_{ij} represents the menu item j in restaurant i.

The objective function for the lazy mode is listed below.

The decision variables are x_i, which represents the restaurant, and y_{ij} represents the menu item j in restaurant i.

max. $\Sigma(H_{ij} - 0.01Ca_{ij} - 0.1Pr_{ij})y_{ij}$, where H_{ij} represent the degree of appropriateness (某店某餐爽度), Ca_{ij} represents the calories contained, Pr_{ij} represents the price of the item, and y_{ij} represents the menu item j in restaurant i.

The parameters for every single mode in the model is as listed below.

Starting point P_s, s=1,2, 3...,7
Ending point/classroom P_e, e=1,2, 3...,8
Departing time Tg_{si}
Arriving time Tb_{ie}
Waiting time Tw_i
Class time T_c
Store opening time Ti_{open}
Store closing time Ti_{close}

Date in the system T_d
Time in the system T_s
Degree of appropriateness H_{ij}
Calorie for the menu item Ca_{ij}
Price for the menu item Pr_{ij}
Any rest days C_{ik}, k=1,2,3,4,5,6,7 (k meaning day of the week, 1 = Sunday)

Since we have three different modes, the constraints for each mode are different in some aspects.

For the diet mode, the constraint is listed below.

#total journey must be completed before class time

$$\Sigma\ (Tg_{si}+Tb_{ie}+Tw_i)x_i \leq T_c - T_s \quad \text{for i=1,...,N}$$

#Can only eat from one restaurant or not eat breakfast at all (also includes if the restaurant has rest days during the week)

$$C_{ik} * x_i \leq 1, \text{ for i =1,...,N, k} = T_d$$

#Restaurant must be open

$$T_s+Tg_{si} > Ti_{open} * x_i$$
$$T_s+Tg_{si} * x_i < Ti_{close}$$

#only choose meals from the selected restaurant

$$y_{ij} \leq x_i, \text{ for i = 1,...,N, j = 1, 2, 3}$$

#can only choose 1 meal

$$y_{ij} \leq 1, \text{ for i =1..., N, for j=1, 2, 3}$$

#Budget constraint

$$Pr_{ij}y_{ij} \leq B$$

#Calorie constraint

$$Ca_{ij}y_{ij} \leq K$$

For the lazy mode the constraint is listed below.

#total journey must be completed before class time

$$\Sigma\ (Tg_{si}+Tb_{ie}+Tw_i)x_i \leq T_c - T_s \quad \text{for i=1,...,N}$$

#Can only eat from one restaurant or not eat breakfast at all (also includes if the restaurant has rest days during the week)

$$C_{ik} * x_i \leq 1, \text{ for i =1,...,N, k} = T_d$$

#Restaurant must be open

$$T_s+Tg_{si} > Ti_{open} * x_i$$
$$T_s+Tg_{si} * x_i < Ti_{close}$$

#only choose meals from the selected restaurant

$$y_{ij} \leq x_i, \text{ for i = 1,...,N, j = 1, 2, 3}$$

#can only choose 1 meal

y_{ij} ≤ 1, for i =1..., N, for j=1, 2, 3

The constraints for the celebration mode is listed below.

#total journey must be completed before class time

Σ (Tg_{si}+Tb_{ie}+Tw_i)x_i≤ T_c -T_s for i=1,...,N

#Can only eat from one restaurant or not eat breakfast at all（also includes if the restaurant has rest days during the week）

C_{ik} * x_i ≤ 1, for i =1,...,N, k = T_d

#Restaurant must be open

T_s+Tg_{si}>Ti_{open}*x_i

T_s+Tg_{si}*x_i<Ti_{close}

#only choose meals from the selected restaurant

y_{ij} ≤ x_i, for i = 1,...,N, j = 1, 2, 3

#can only choose 1 meal

y_{ij} ≤ 1, for i =1..., N, for j=1, 2, 3

Although the constraints for the celebration mode and the lazy mode is similar, we would put greater emphasis for lower distance in the lazy mode.

The Model

This program is created for students and professors who studies or works in the National Taiwan University area. The flowchart of our program is as listed below in figure 1.

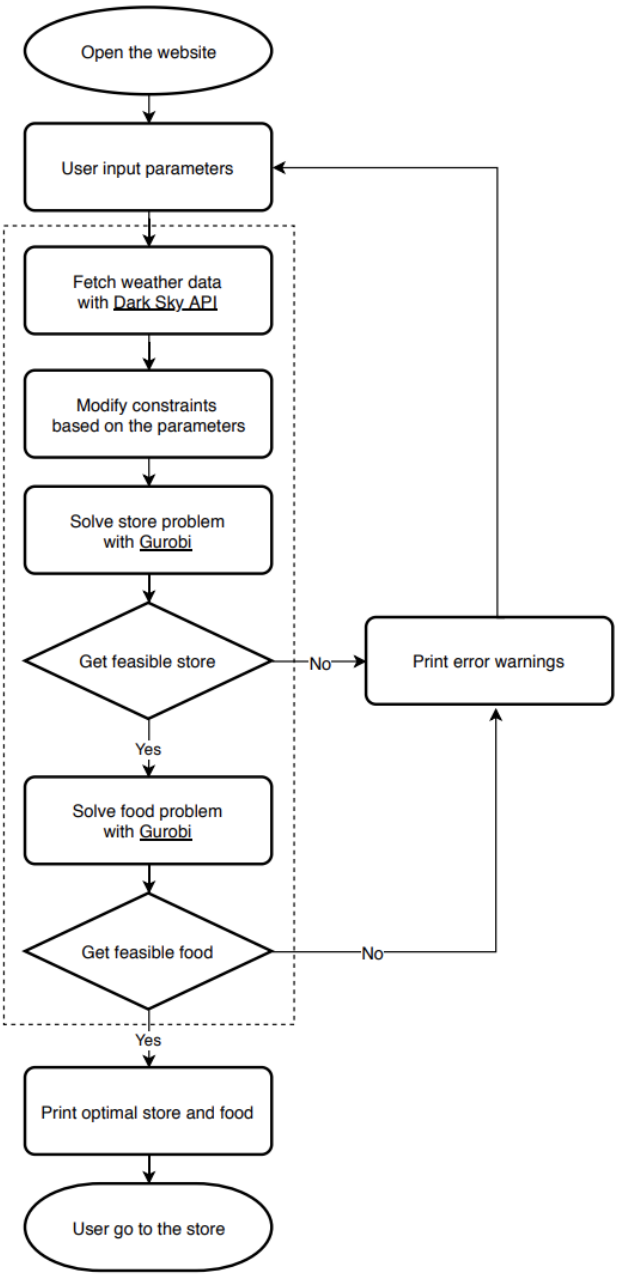


Figure 1 – Flowchart for Model

First of all, the program will ask the user to input some parameters, such as his/her budget, the starting location and ending location. Afterwards, the program will fetch the weather data for that morning by using the Dark Sky API and gain the exact data according to the user's location. Also, the parameters that are inputted by the user will be used to modify

the constraints that are set-up. Then, the program will run and search for the best possible breakfast option for the user and print it for the user. If the user does not like the option, the program will re-run it until the user is satisfied. After that, the user should go to the store and buy it.

We have also set up three different modes to for the user to choose from. These three modes are lazy mode, celebration mode, and diet mode. Celebration mode are for those who are not on a tight budget, thus the breakfast stalls with a higher price range will be considered during the selection process. The diet mode takes into account the calories each breakfast option has. If the user input exceeds the food listed, it will not be shown as a result for the potential breakfast option. Lazy mode is for those who does not want to spend too much time on the road and would prefer the restaurant with the shortest possible distance from their starting point to their ending point.

With every program, there are assumptions that we have made for this program. First of all, since the audience for this program are National Taiwan University students or professors, the starting locations that are available to be chosen from are Gong guan MRT station, Chang Xing BOT, 9th women's dorm, Shuiyuan BOT, technology building, South Xin sheng road parking lot, and Fu bell. Likewise, the available classroom that can be chosen from the options are the NTU sports center, Core subjects building, Liberal education building, Freshman classroom, common subjects building, engineering building, Department of civil engineering building, Civil engineering research building, and the computer and information networking center/foreign language teaching center. Also, the available restaurants that are in consideration for the program are also listed below. We chose all the restaurants that are in the gong guan area that are open from 7am. In total, there are 43 options available. Although there are probably more restaurants that are in the area, we believe that the restaurants that we selected have great variety and competitive price range. It is the stalls that most people would go to and buy food from. It is listed in figure 2. For each restaurant, we took data of each restaurant's distance from the starting location and ending location listed above, the three main items that they sell along with their calorie intake and their price, and the approximate waiting time. After that, we would put them in array format in the python coding process, which is show in figure 3.

```
store_list = {1:'7-eleven Zhangxing Store 7-eleven 長興店',
              2:'Asakusa Japanese rice balls 淺草日式飯糰',
              3:'Anhaoshih Heping Store 安好食 和平店',
              4:'NTU Black Ricaball台大黑飯糰',
              5:'My Kitchen Breakfast 我家廚房早餐吧',
              6:'Liang Liang Hao 倆倆號',
              7:'MOS BURGER Zhangxing Store 摩斯漢堡長興店',
              8:'FamilyMart Zhangxing Store 全家長興店',
              9:'7-eleven Siyuan Store 7-eleven水源店',
              10:'Subway Siyuan Store Subway 水源店',
              11:'STARBUCKS Gounguan Store 星巴克 公館店',
              12:'Liu\'s Fried Bun 劉家水煎包',
              13:'Snowballer Cafe Gongguan Store 雪球咖啡 公館店',
              14:'Toast Toast 土司吐司',
              15:'Eat Ricaball 呷飯團',
              16:'Eat Toast Bar 吃吐吧',
              17:'The House Breakfast 好食',
              18:'Bread Société 布列德麵包',
              19:'Snowballer Cafe Fuxing Store 雪球咖啡 復興店',
              20:'Subway Fuxing South Store Subway 復興南店',
              21:'ZhenFang 真芳早餐',
              22:'Mc Donald\'s Fuxing 2nd Store 麥當勞 科技大樓店',
              23:'7-eleven Heping East Store 7-eleven 和平東路店',
              24:'7-eleven MRT Technology Building Station 7-eleven 科技大樓店',
              25:'Xiao Bakery 小福麵包店',
              26:'Xiaofu Bistro 小福比司多',
              27:'MOS BURGER Xiaofu Store 摩斯漢堡小福店',
              28:'Mc Donald\'s NTU 麥當勞 活大店',
              29:'FamilyMart New NTU Store 全家活大店',
              30:'Chunyi Bakery 活大麵包店 (純義西點麵包坊) ',
              31:'NTU Freshman Female Dorm Bistro 大一女比司多',
              32:'Shine Mood NTU 小木屋鬆餅',
              33:'Pasta & Toast 趴兔',
              34:'Xiaoxiaofu 3rd Store  第三小吃部(小小福)',
              35:'Xiaoxiaofu 2nd Store 第二小吃部(小小福)',
              36:'Xiaoxiaofu Fried Chicken 大叔韓國炸鷄(小小福)',
              37:'7-eleven NTU Elite store 7-eleven 鹿鳴店',
              38:'Hi-Life Guoching Store萊爾富 國青店',
              39:'FamilyMart NTU Maple Store 全家 女九店',
              40:'NTU Female Dorm #9 Restaurant 女九早餐',
              41:'Zhuei1 Breakfast Restaurant 豬a早餐店',
              42:'FamilyMart Xinsheng South Store 全家 新生南路店',
              43:'Anhaoshih Xinsheng South Store 安好食 新生南路店'}
```

Figure 2 – List of Restaurants that will be used

```
Dg = np.array(
     [[1050,   0, 700,1600, 950,1350, 900],
      [1500, 700, 600,2000, 450,1000,1100],
      [1500, 700, 600,2000, 450,1100,1200],
      [1400, 850, 500,1900, 550, 950,1000],
      [1200,  50, 850,1750,1100,1500,1050],

      [ 350,1200,1000, 400,1900, 950, 600],
      [1050,   0, 700,1600, 950,1350, 900],
      [1200,  50, 850,1750,1100,1500,1050],
      [ 600,1600,1200,  10,2300, 800, 750],
      [ 650,1700,1300,  80,2300, 900, 850],

      [ 350,1300,1000, 300,2000, 600, 550],
      [ 500,1400,1100, 240,2000, 700, 650],
      [ 350,1300,1100, 350,1900, 850, 650],
      [ 400,1300,1100, 500,1900,1000, 650],
      [ 280,1200,1000, 450,1900, 850, 600],

      [ 280,1200,1000, 450,1900, 850, 600],
      [2000,1100,1400,2600, 350,1900,1800],
      [1700,1100,  900,2100, 170,1300,1400],
      [1500,  800, 900,2100, 150,1500,1300],
      [1700,1100,  950, 220, 130,1400,1500],

      [1800,1000,1000,2200, 110,1500,1500],
      [1800,1000,1100,2300, 130,1600,1600],
      [1750,1100,1000,2100, 140,1450,1450],
      [1800,1100,1200,2400, 140,1700,1700],
      [ 750, 900, 300,1100, 900, 200, 250],

      [ 750, 900, 300,1100, 900, 200, 250],
      [ 750, 900, 300,1100, 900, 200, 250],
      [ 850, 700, 250,1250, 600, 500, 500],
      [ 800, 650, 300,1200, 650, 450, 450],
      [ 850, 700, 250,1250, 600, 500, 500],

      [ 100,1100, 950, 750,1550, 550, 350],
      [ 300, 800, 550,1050,1150, 550, 350],
      [ 210, 850, 650, 850, 800, 850, 240],
      [ 230, 850, 700, 800,1500, 900, 300],
      [ 230, 850, 700, 800,1500, 900, 300],

      [ 230, 850, 700, 800,1500, 900, 300],
      [ 400, 950, 750, 950,1600,1000, 450],
      [1400, 600, 750,1900, 550,1300,1200],
      [ 800, 700,  60,1200,1000, 700, 500],
      [ 800, 700,  60,1200,1000, 700, 500],

      [ 750,1300, 750, 700,1600, 120, 550],
      [ 700,1300, 700, 650,1600, 190, 450],
      [ 700,1400, 800, 700,1800, 300, 600]])
```

Figure 3 – Example Matrix in Array form.

The objective of this model is that with the data base that we have, from the 43 restaurants that we have selected, we can find the optimum breakfast option for the user which satisfies his or her wants, including low budget, low calories, or short distances.

The constraints that we have regarding this program are the budget constraints, calorie constraints, and time constraints, which all explains the three different modes that we offer. Here below is the coding for the diet mode, which is for breakfast options with lower calorie intake.

```
    menu_array_to_unicode = (s.decode(encoding = 'utf-8') for s in menu_array)

    Pr_ave = np.zeros(N)
    for i in range(N):
        Pr_ave[i] = (Pr[i][0]+Pr[i][1]+Pr[i][2])/3
    Ca_ave =np.zeros(N)
    for i in range(0,N):
        Ca_ave[i] = (Ca[i][0]+Ca[i][1]+Ca[i][2])/3
    m = Model('lp1')
    m.setParam( 'OutputFlag', False )
    x = {}
    for i in range(0,N):
        x[i] = m.addVar(name = 'x_%s' % (i + 1),vtype = 'i')

    m.update()

    m.setObjective(quicksum(x[i]*(H[i] - 0.002*(Dg[i,s-1] + Db[i,e-1])) for i in range(0,N)), GRB.MAXIMIZE)

    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i]*peak[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V >= 420) and (Ts + Dg[i,s-1])/V <= 540)) <= (Tc-Ts))
    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V < 420) or (Ts + Dg[i,s-1])/V > 540)) <= (Tc-Ts))

    m.addConstr(quicksum(C[i,day-1]*x[i] for i in range(0,N)) <= 1)

    #價格限制
    m.addConstr(quicksum((Pr_ave[i]*x[i])for i in range(0,N))<=B)
    #熱量限制
    m.addConstr(quicksum((Ca_ave[i]*x[i])for i in range(0,N))<=K)

    m.update()

    for i in range(0,N):
        m.addConstr(x[i] <= C[i,day-1])

    for i in range(0,N):
        m.addConstr(T_open[i] * x[i] <= Ts + Dg[i,s-1]/V)
        m.addConstr((Ts + Dg[i,s-1]/V) * x[i] <= T_close[i])


    m.optimize()

    count = 0
    record = 0
    for i in range(0,N):
        if(m.getVars()[i].x == 1):
            print store_list[i+1]
            for j in range(3):
                if Ca[i][record]>=Ca[i][j]:
                    record = j
            if Pr[i][record]>B:
                print 'Spend a little bit more for less calories^^'
            print menu_array[i][record], 'Price: %d Calories: %d'% (Pr[i][record],Ca[i][record])
            print 'Moving distance = %d m' %(Dg[i,s-1] + Db[i,e-1])
            count = 1
            break
    if count == 0:
        print 'You should rush to the classroom, or you\'ll be late!'
else:
    print 'You are late!!!'
```

Figure 4 – Constraint Coding for Diet Mode

Here is the coding for the constraints of the lazy mode, for breakfast options with fewest distances.

```
    m = Model('lp1')
    m.setParam( 'OutputFlag', False )
    x = {}
    for i in range(0,N):
        x[i] = m.addVar(name = 'x_%s' % (i + 1),vtype = 'i')

    m.update()

    m.setObjective(quicksum(x[i]*(H[i] - 0.002*(Dg[i,s-1] + Db[i,e-1])) for i in range(0,N)), GRB.MAXIMIZE)

    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i]*peak[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V >= 420) and (Ts + Dg[i,s-1])/V <= 540)) <= (Tc-Ts))
    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V < 420) or (Ts + Dg[i,s-1])/V > 540)) <= (Tc-Ts))

    m.addConstr(quicksum(C[i,day-1]*x[i] for i in range(0,N)) <= 1)

    #價格限制
    m.addConstr(quicksum((Pr[i]*x[i])for i in range(0,N))<=B)
    #熱量限制
    #m.addConstr(quicksum((Ca[i]*x[i])for i in range(0,N))<=K)

    m.update()

    for i in range(0,N):
        m.addConstr(x[i] <= C[i,day-1])

    for i in range(0,N):
        m.addConstr(T_open[i] * x[i] <= Ts + Dg[i,s-1]/V)
        m.addConstr((Ts + Dg[i,s-1]/V) * x[i] <= T_close[i])


    m.optimize()

    count = 0
    for i in range(0,N):
        if(m.getVars()[i].x == 1):
            print store_list[i+1]
            if Ca[i]>500:
                print 'This meal is abit fat'
            print menu_list[i+1], 'Price: %d Calories: %d'% (Pr[i],Ca[i])
            print 'Moving distance = %d m' %(Dg[i,s-1] + Db[i,e-1])
            count = 1
            break
    if count == 0:
        print 'You should rush to the classroom, or you\'ll be late!'

else:
    print 'You are late!!!'
```

Figure 5 – Constraint Coding for Lazy Mode

Here is the coding for the constraints of the celebration mode, which also includes breakfast options with a higher price ranges.

```
    m = Model('lp1')
    m.setParam( 'OutputFlag', False )
    x = {}

    for i in range(0,N):
        x[i] = m.addVar(name = 'x_%s' % (i + 1),vtype = 'i')

    m.update()

    m.setObjective(quicksum(x[i]*(H[i] - 0.005*(Dg[i,s-1] + Db[i,e-1])) for i in range(0,N)), GRB.MAXIMIZE)

    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i]*peak[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V >= 420) and (Ts + Dg[i,s-1])/V <= 540)) <= (Tc-Ts))
    m.addConstr(quicksum(((Dg[i,s-1] + Db[i,e-1])/V + Tw[i])*x[i] for i in range(0,N) if (((Ts + Dg[i,s-1])/V < 420) or (Ts + Dg[i,s-1])/V > 540)) <= (Tc-Ts))

    m.addConstr(quicksum(C[i,day-1]*x[i] for i in range(0,N)) <= 1)

    m.update()

    for i in range(0,N):
        m.addConstr(x[i] <= C[i,day-1])

    for i in range(0,N):
        m.addConstr(T_open[i] * x[i] <= Ts + Dg[i,s-1]/V)
        m.addConstr((Ts + Dg[i,s-1]/V) * x[i] <= T_close[i])


    m.optimize()

    count = 0
    for i in range(0,N):
        if(m.getVars()[i].x == 1):
            print store_list[i+1]
            print 'Moving distance = %d m' %(Dg[i,s-1] + Db[i,e-1])
            count = 1
            break
    if count == 0:
        print 'You should rush to the classroom, or you\'ll be late!'

else:
    print 'You are late!!!'
```

Figure 6 – Constraint Coding for Celebration Mode

As you can see as well, if the person is running a bit late, we would also show that he or she should skip breakfast or else the user would be late for class.

Results

The pictures below will be the result of some test runs that we have done. For all the test runs of each mode, we decided to have the starting position as Shui yuan BOT and the ending position is the engineering building. Figure 7 is the result for the lazy mode. Figure 8 is the result for the celebration mode. Lastly, figure 9 is the result for the diet mode.

```
Current time: 08:12

Please input when will your class start today. hour:9

minute:10

Please input your place now:
1:Gongguan, 2:Changxing BOT, 3:9th Women's Dorm, 4:Shuiyuan BOT, 5:Technology Building, 6:Xinsheng S Rd Parking Lot, 7:Fu Bell
4

Please input your classroom:
1:Sports Center
2:Core Subject Classroom/Liberal Education Classroom
3:Freshman Classroom
4:Common Subjects Classroom
5:Engineering Building
6:Dept. of Civil Engineering
7:Civil Engineering Research Building
8:Computer and Information Networking Center/Foreign Language Teaching Center
5

Please input your breakfast budget for today:60
MOS BURGER 小福店
火腿歐姆蛋堡+紅茶 Price: 60 Calories: 447
Moving distance = 1350 m
```

Figure 7 – Lazy Mode Test Run

```
Current time: 08:09

Please input when will your class start today. hour:9

minute:10

Please input your place now:
1:Gongguan, 2:Changxing BOT, 3:9th Women's Dorm, 4:Shuiyuan BOT, 5:Technology Building, 6:Xinsheng S Rd Parking Lot, 7:Fu Bell
4

Please input your classroom:
1:Sports Center
2:Core Subject Classroom/Liberal Education Classroom
3:Freshman Classroom
4:Common Subjects Classroom
5:Engineering Building
6:Dept. of Civil Engineering
7:Civil Engineering Research Building
8:Computer and Information Networking Center/Foreign Language Teaching Center
5
小福比司多
Moving distance = 1350 m
```

Figure 8 – Celebration Mode Test Run

```
Current time: 08:15

Please input when will your class start today. hour:9

minute:10

Please input your place now:
1:Gongguan, 2:Changxing BOT, 3:9th Women's Dorm, 4:Shuiyuan BOT, 5:Technology Building, 6:Xinsheng S Rd Parking Lot, 7:Fu Bell
4

Please input your classroom:
1:Sports Center
2:Core Subject Classroom/Liberal Education Classroom
3:Freshman Classroom
4:Common Subjects Classroom
5:Engineering Building
6:Dept. of Civil Engineering
7:Civil Engineering Research Building
8:Computer and Information Networking Center/Foreign Language Teaching Center
5

Please input your breakfast budget for today:60

Please input your espected calories intake for breakfast:
(Max calories per day: boys= 2000, girls= 1600)
300
劉家水煎包
高麗菜水煎包+豆漿 Price: 30 Calories: 150
Moving distance = 1340 m
```

Figure 9 – Diet Mode Test Run

As you can see from the test result above, first, the system will use to the coding to call what time it is. Then it will ask what time our next class time, asking for the hours and the minutes. Afterwards, it will ask the user to input his or her starting location and ending location from the given selections that the program. After this is where the differences lie.

For the lazy mode, it will ask the user to input his or her budget. Then it will select a restaurant from our restaurant list that fits the requirement of the user. It will print out the restaurant name, the item for you to order, the price, and the calories. Lastly, it will show you the total distance that is covered.

For the celebration mode, since this is the mode where it suits users who have unlimited budget without any calorie intake constraints, this is where the restaurants that will appear is somewhat random. After running the program, the result will print out just the name of the restaurant and the total distance that you will need to cover if going to the restaurant. Notice that since for this mode there is not any budget constraint, the program will not list out the item that they need to order as the user can order anything the user want off the menu.

For the diet mode, it is quite similar to the lazy mode, but placing more importance on the calories that the food has instead of the distance that the user has to cover. It will ask the user to input the budget as well. But after that, there is an additional step, which is to input the maximum number of calories that the user wants to eat for breakfast. This adds an additional constraint for the model to solve. Afterwards, after the program runs, it will print out the name of the restaurant, the item on the menu that the user has to order along with the calorie intake for the breakfast item. Lastly, it will also show the total distance the user will cover if he or she goes to that restaurant.

We also created a website for this program to run to. It is to make the user experience a lot more convenient. Results of this would be shown on the next page.

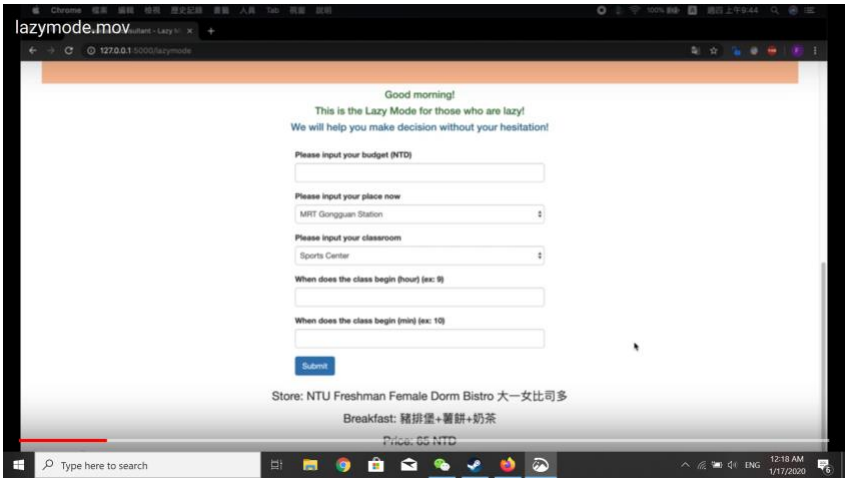


Figure 10 – Website test run for Lazy Mode

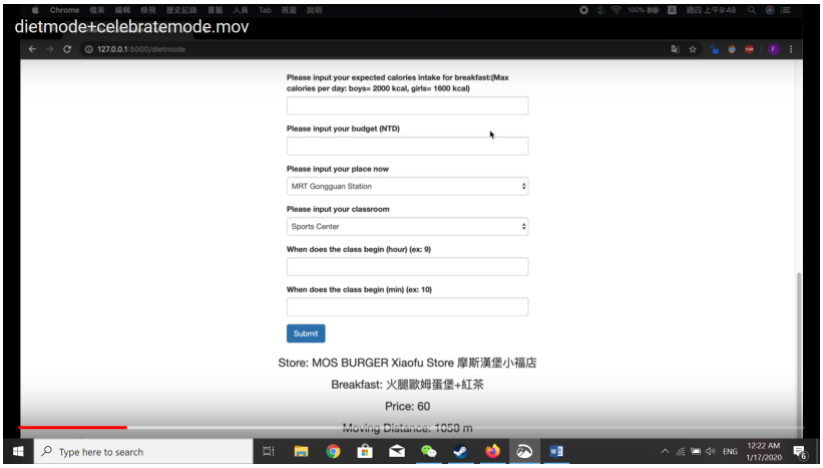


Figure 11 – Website test run for Diet mode

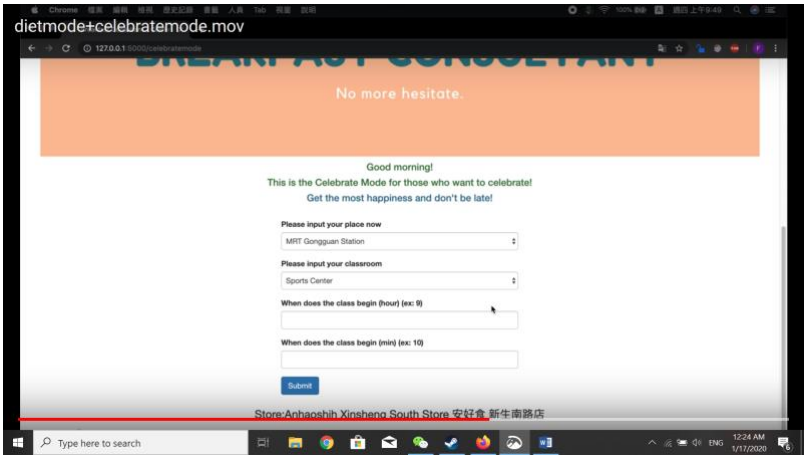


Figure 12 – Website test run for Celebration mode

Furthermore, since we use some coding to get the real actual date and time when the user uses the program, we can also tell if he or she has enough time to go buy breakfast and make it to class on time. We believe that buying your breakfast should not be the reason why you are late for class. For this reason, if the user has insufficient time, the program will say "you should rush to the classroom, or you'll be late" or "you are late".

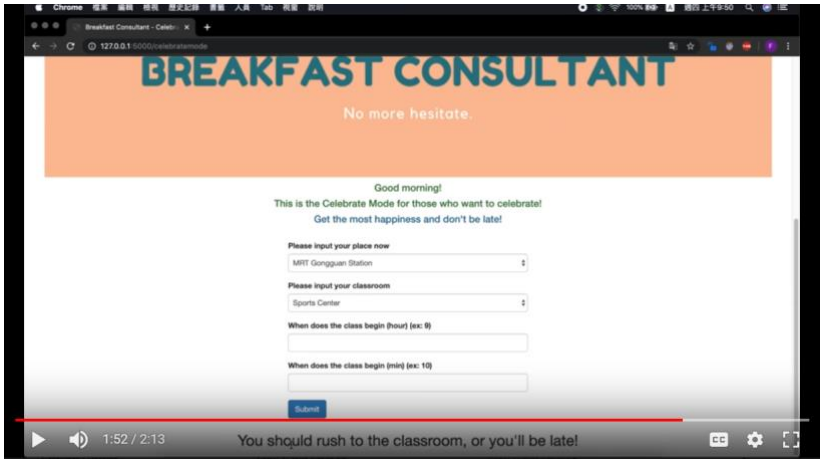


Figure 14 – "You should rush to the classroom, or you'll be late" example

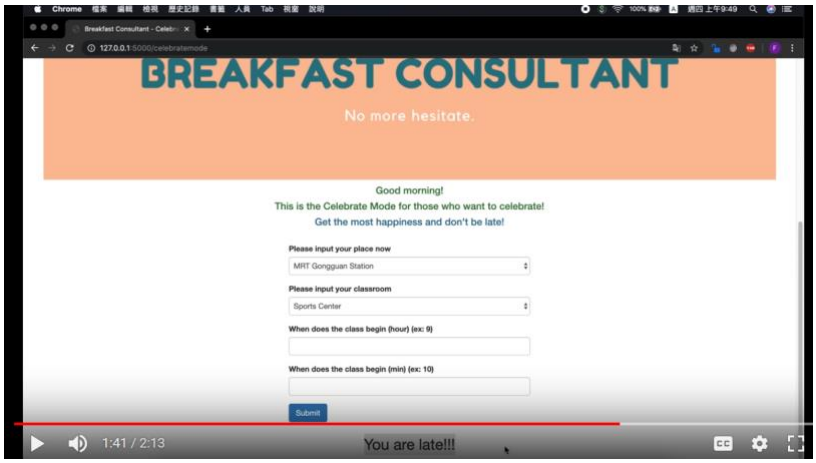


Figure 14 – "You are late" example

Due to some financial and technical problem, we cannot make the website online at this time. Therefore, we record the process we tried out our website. The recorded videos can be access through the following links.

- The lazy mode:

https://drive.google.com/file/d/1m6SLKHTfBDgpkFgXQxTDxQ1x_CiPNZlo/view?usp=sharing

- The diet mode and the celebration mode:

https://drive.google.com/file/d/14bUhZkx7NixBZuajJm0H1So3H00yc7Vi/view?usp=sharing

Reflection

What we have learned as a group from working in this project is that apart from the obvious which is that typing a lot for the coding process, especially for the numerous amounts of restaurants that we have, along with the respective distances, price, and timing variables, it does take quite some time to do so. In addition to that, we have to be really careful when typing in the data as a little mistake when filling in the matrix for the arrays and typing the constraints can have huge repercussions for the results.

We also realize that turns out there are really a lot of options for breakfast in the Gong guan area. Usually, as students, we would probably go to the same restaurants twice or three times a week. Some would also just go to the nearby seven eleven or family mart convenience store to grab a quick breakfast. However, during the selection process, after doing quite some research in the area, turns out there are a lot of restaurants that we ourselves has not even tried before. To most of us, it feels like we just moved in into a new area and there are so many more food options to try for breakfast.

Lastly, through this project, we as a group realize how programming can really make a big mathematical problem and solve it within seconds. It just goes to show how much technology has improved over the past two decades. Imagine solving this by paper and 100% of the people would just give up and go to the nearest convenience store. Nobody twenty years ago would have thought to make a program to solve for what to eat for breakfast, but here we are now. It's quite astounding that we can use a mathematical approach to solve our problems.

Work Distribution

b05501012 薛功囷: Collecting data for restaurant, making the equations and constraints & converting it into python form (14.3%)

b05501053 施欣承: Collecting data for restaurant, making the equations and constraints (14.3%)

b05501089 黃欣穎: Collecting data for restaurant, revision of model and converting it into python form (14.3%)

b05501095 林永安: Collecting data for restaurant, making the report (14.2%)

b05501108 陳妤華: Collecting data for restaurant, making the website, making equations from the problem (14.3%)

b05501115 劉燕宜: Collecting data for restaurant, coding, making the equations and constraints (14.3%)

b04208002 陳思彤: Collecting data for restaurant, coding, making the equations and constraints, making equations from the problem (14.3%)

Extra Credit

We believe that we deserve extra credit for the creativeness of the idea, which is to solve a problem that a lot of people have but are not willing to admit. We think that this model will help a lot of students choose the best restaurants available for them to their respective needs and likings. We also believe that inputting more than forty restaurants to choose from, three items from each restaurant with details about their price and calories, finding their distances from their starting locations and to their respective classrooms is somewhat of a tiring job and should show our dedication to the project.

Also, we also did three modes to satisfy the user's needs, such as limiting his or her calorie intake, or wants to spend as little time on the road on the way to class. This could be very useful as if it was perhaps a rainy day, most people would opt to go for the lazy mode, which means that they would have to spend less time on the road, meaning they have a smaller chance of getting soaked from the rain. If one has a reason to celebrate in morning, for example after getting a very high score in his or her midterms, he or she might want to splash out on a big breakfast just to celebrate and use the celebration mode. Celebration mode could also be used for those who are stressed and just want to eat a lot to prepare for the upcoming exams that they may face. We believe that giving three modes to choose from offers some flexibility for the user.

We also designed a web page for the program to run on, which also consumes some time as we believe that without a webpage, nobody would really want to use the program we made. It has to be simple yet user-friendly and appealing to the eye.

References

1. https://www.canva.com
2. https://hackmd.io/@shaoeChen/ByofdR1XG?type=view&fbclid=IwAR3NkbuZlN4Mk_5dWwdpS0na9ntqhtxoj412bpwWO-dSddZ98VBeiuFv_3Q
3. https://andy6804tw.github.io/2018/01/06/bootstrap-tutorial(1)/?fbclid=IwAR0YEuc3arcuLLkn83seBTmUb2lB7g3lhCIPzQbGffyAkKbwwqsBej6R20s
4. https://bootstrap.hexschool.com/?fbclid=IwAR3Epl9SSTAU1yxT7fG_K791vMCYwffD9s00auVCwFXRwbZJG2xjUnJjiwc
5. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture3_ORClass.pdf
6. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture4_ORPost.pdf
7. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture5_ORPost.pdf
8. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture6_ORClass.pdf
9. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture7_ORClass.pdf
10. https://ceiba.ntu.edu.tw/course/fbb9d3/content/lecture8_ORClass.pdf