VIETNAM NATIONAL UNIVERSITY - HCM
Ho Chi Minh City University of Technology
Faculty of Computer Science and Engineering

MATHEMATICAL MODELING (CO2011)

Assignment

Resource Allocation using ILP

	Name	Student's ID
1	Lê Hiếu Phương	1752431
2	Phạm Nhật Phương	1752042
3	Nguyễn Hữu Trung Nhân	1752392

Ho Chi Minh City, December 2019

Contents

1 Introduction

An integer linear programming (ILP) problem is a mathematical optimization program which contains an objective function and the constraints. Integer linear programming is widely used to model practical problems into linear program using integer variables for two main reasons:

1. The integer variables represent quantities that can only be integer. For example, in reality, it is not possible to build 6.3 cars.

2. The integer variables represent decisions and so should only take on the binary value 0 or 1. For instance, the state opened or closed of a factory.

These two considerations above occur frequently in practice. Therefore, integer linear programming can be used in many applications areas. In this mathematical modeling assignment, the main goal is to implement a program which has the ability to find an optimal solution for the mathematical model of the below practical problem:

In modeling distribution systems, decisions must be made about trade-offs between transportation costs and costs for operating distribution centers. As an example, suppose that a manager must decide which of n warehouses to use for meeting the demands of m customers for a good. The decisions to be made are which warehouses to operate and how much to ship from any warehouse to any customer. The relevant costs are given as follows:

f_i: Fixed operating cost for warehouse i, if opened (for example, a cost to lease the warehouse);

c_{ij}: Per-unit operating cost at warehouse i plus the transportation cost for shipping from warehouse i to customer j;

Following conditions must be satisfied in the model:

1. The demand d_j of each customer must be filled from the warehouses.

2. Goods can be shipped from a warehouse only if it is opened.

Determine amount which should be sent from warehouse i to customer j such that the total cost (including the operating and transportation costs) is minimum, for $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$.

From the above problem, it is easy to recognized that the solution for this problem as least required two fundamental steps. The first one is to formulate the problem under a integer linear program which contains the objective function and the constraints. The second step is to implement a program (that has the ability to solve this mathematical model) using Python programming language.

2 Background Knowledge

2.1 Convex Optimization

A convex optimization problem is one of the form

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq b_i, \qquad i = 1, ..., m \end{aligned}$$

where the functions $f_0, ..., f_m : R^n \longrightarrow R$ are convex and satisfy $f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$ for all $x, y \epsilon R^n$ and all α, $\beta \epsilon R$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$.

There are three basic properties of convex optimization problem:
1. Every local minimum is a global minimum
2. The optimal set is convex
3. If the objective function is strictly convex, then the problem has at most one optimal point

2.2 Linear Programming

An integer programming problem is a mathematical optimization in which some or all of the variables are restricted to be integers. A general linear program has the form

$$\begin{aligned} &\text{minimize} \qquad && c^T x + d \\ &\text{subject to} && Gx \leq h \\ & && Ax = b \end{aligned}$$

where $G \epsilon R^{mxn}$ and $A \epsilon R^{mxn}$. Linear programs are a part of convex optimization problems. It is possible to omit the constant d in the objective function, since it does not affect the optimal (or feasible) set. Therefore, an affine objective $c^T x + d$ can be maximized.

3 Forming The Mathematical Model

Let y_i is the state of the ith warehouse $(i = 1, 2, ..., n)$:

$$y_i = \begin{cases} 1, & \text{if warehouse } i \text{ is opened;} \\ \\ 0, & \text{if warehouse } i \text{ is not opened;} \end{cases}$$

Let x_{ij} is the amount to be sent from warehouse i to customer j
$(x_{ij} \geq 0, \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., m)$

The objective function comprises of two elements. The first element is the transportation and variable warehousing costs. The second one is fixed costs for operating warehouses:

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} f_i y_i$$

The first constraint is the condition in which each customer's demand must be met:

$$\sum_{i=1}^{n} x_{ij} = d_j \quad (j = 1, 2, ..., m)$$

In the second constraint, the summation over the shipment variables x_{ij} in the ith constraint of the second constraint is the amount of the good shipped from warehouse i. When the warehouse is not opened, $y_i = 0$ and the constraint specifies that nothing can be shipped from the warehouse. On the other hand, when the warehouse is opened and $y_i = 1$, the constraint simply states that the amount to be shipped from warehouse i can be no larger than the total demand, which is always true. Consequently, the second constraint implies the restriction number 2 as proposed above in section 1:

$$\sum_{j=1}^{m} x_{ij} - y_i (\sum_{j=1}^{m} d_j) \leq 0 \quad (i = 1, 2, ..., n)$$

In summary, the model is:

$$\text{Minimize } Z = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{i=1}^{n} f_i y_i$$

Subject to:

$$\sum_{i=1}^{n} x_{ij} = d_j, \qquad (j = 1, 2, ..., m)$$

$$\sum_{j=1}^{m} x_{ij} - y_i (\sum_{j=1}^{m} d_j) \leq 0, \qquad (i = 1, 2, ..., n)$$

$$x_{ij} \geq 0, \qquad (i = 1, 2, ..., n, \quad j = 1, 2, ..., m)$$
$$y_i = 0 \text{ or } 1, \qquad (i = 1, 2, ..., n)$$

4 Implementing The Program

Python programming language is chosen to implement the program which solves the above model. A library which helps the programming process easier is PuLP. PuLP is an linear programming modeler written in Python. This library can generate linear programming files, and call solvers to solve linear problems. The solvers which are supported by this library are: GLPK, COIN, CPLEX and GUROBI. In particularly, the GLPK package is intended for solving large-scale linear programming (LP), mixed integer programming (MIP), and other related problems.

The very first step in the beginning is to create the specification.

Customer	1	2	3	4	5	6	7	8	9	10		
Demand	30	49	23	33	19	40	20	30	38	46		
Warehouse												f
1	1.4	1.64	1.56	3.64	4.04	3.84	4.28	2.76	2.96	2.12		4.24
2	2.12	4.88	2.68	1.16	3.96	1.12	3.76	2.52	2.8	3.72		4.64
3	3.2	4.84	4.68	4.4	2.56	2.12	2.28	4.08	3.6	3.64		1.52
4	4.84	2.96	4.16	4.72	3.64	1.2	4.8	4.2	3.84	1.64		4.64
5	4.84	4.2	4.84	3.72	1.68	1.4	1.12	1.76	4	1.48		3.52

Figure 1: Input 01

```python
CUSTOMERS = [1,2,3,4,5,6,7,8,9,10]        # m customers
WAREHOUSE = ['h1','h2','h3','h4','h5']    # n warehouses

demand = {1 : 30,             # d_j
          2 : 49,
          3 : 23,
          4 : 33,
          5 : 19,
          6 : 40,
          7 : 20,
          8 : 30,
          9 : 38,
          10 : 46}

fixed_operating_cost = {'h1' : 4.24,    # f_i
                        'h2' : 4.64,
                        'h3' : 1.52,
                        'h4' : 4.64,
                        'h5' : 3.52}

transport_cost = {'h1':{1 : 1.40, 2 : 1.64, 3 : 1.56, 4 : 3.64, 5 : 4.04, 6 : 3.84, 7 : 4.28, 8 : 2.76, 9 : 2.96, 10 : 2.12},       # c_ij
                  'h2':{1 : 2.12, 2 : 4.88, 3 : 2.68, 4 : 1.16, 5 : 3.96, 6 : 1.12, 7 : 3.76, 8 : 2.52, 9 : 2.80, 10 : 3.72},
                  'h3':{1 : 3.20, 2 : 4.84, 3 : 4.68, 4 : 4.40, 5 : 2.56, 6 : 2.12, 7 : 2.28, 8 : 4.08, 9 : 3.60, 10 : 3.64},
                  'h4':{1 : 4.84, 2 : 2.96, 3 : 4.16, 4 : 4.72, 5 : 3.64, 6 : 1.20, 7 : 4.80, 8 : 4.20, 9 : 3.84, 10 : 1.64},
                  'h5':{1 : 4.84, 2 : 4.20, 3 : 4.84, 4 : 3.72, 5 : 1.68, 6 : 1.40, 7 : 1.12, 8 : 1.76, 9 : 4.00, 10 : 1.48}}
```

Figure 2: Input in Python code

The second step is to input the model of the problem in section 2 into the Python program.

```python
# The total cost
total_cost = LpProblem("ILP",LpMinimize)

# x_ij , y_i
amount_sent = LpVariable.dicts("Amount",
                               [(j,i) for j in CUSTOMERS
                                      for i in WAREHOUSE],
                               0)

is_opened = LpVariable.dicts("Opened_Warehouse",WAREHOUSE,0,1,LpBinary)

#OBJECTIVE FUNCTION
total_cost += lpSum(fixed_operating_cost[i]*is_opened[i] for i in WAREHOUSE) + lpSum(transport_cost[i][j]*amount_sent[(j,i)] for i in WAREHOUSE for j in CUSTOMERS)

#CONSTRAINTS
for j in CUSTOMERS:
    total_cost += lpSum(amount_sent[(j,i)] for i in WAREHOUSE) == demand[j]                                 #Constraint 1

for i in WAREHOUSE:
    total_cost += lpSum(amount_sent[(j,i)] for j in CUSTOMERS) - is_opened[i]*lpSum(demand[j] for j in CUSTOMERS) <= 0      #Constraint 2
```

Figure 3: Input the model in Python code

And the final step is to run the solver of the library to solve this model and print the solution to the terminal.

```
#SOLUTION
total_cost.solve()
print("Status:",LpStatus[total_cost.status])

TOL = .00001
for i in WAREHOUSE:
    if is_opened[i].varValue > TOL:
        print("The opened warehouse is",i)

for v in total_cost.variables():
    print(v.name,"=",v.varValue)

#PRINT OPTIMAL SOLUTION
print("The total cost =",value(total_cost.objective))
```

Figure 4: Run the mix integer model solver

In the end, the optimal solution is created.

```
D:\Microsoft\VisualStudio\Shared\Python37_64\python.exe
Status: Optimal
The opened warehouse is h1
The opened warehouse is h2
The opened warehouse is h5
Amount_(1,_'h1') = 30.0
Amount_(1,_'h2') = 0.0
Amount_(1,_'h3') = 0.0
Amount_(1,_'h4') = 0.0
Amount_(1,_'h5') = 0.0
Amount_(10,_'h1') = 0.0
Amount_(10,_'h2') = 0.0
Amount_(10,_'h3') = 0.0
Amount_(10,_'h4') = 0.0
Amount_(10,_'h5') = 46.0
Amount_(2,_'h1') = 49.0
Amount_(2,_'h2') = 0.0
Amount_(2,_'h3') = 0.0
Amount_(2,_'h4') = 0.0
Amount_(2,_'h5') = 0.0
Amount_(3,_'h1') = 23.0
Amount_(3,_'h2') = 0.0
Amount_(3,_'h3') = 0.0
Amount_(3,_'h4') = 0.0
Amount_(3,_'h5') = 0.0
Amount_(4,_'h1') = 0.0
Amount_(4,_'h2') = 33.0
Amount_(4,_'h3') = 0.0
Amount_(4,_'h4') = 0.0
Amount_(4,_'h5') = 0.0
Amount_(5,_'h1') = 0.0
Amount_(5,_'h2') = 0.0
Amount_(5,_'h3') = 0.0
Amount_(5,_'h4') = 0.0
Amount_(5,_'h5') = 19.0
Amount_(6,_'h1') = 0.0
Amount_(6,_'h2') = 40.0
Amount_(6,_'h3') = 0.0
Amount_(6,_'h4') = 0.0
Amount_(6,_'h5') = 0.0
Amount_(7,_'h1') = 0.0
Amount_(7,_'h2') = 0.0
Amount_(7,_'h3') = 0.0
Amount_(7,_'h4') = 0.0
Amount_(7,_'h5') = 20.0
Amount_(8,_'h1') = 0.0
Amount_(8,_'h2') = 0.0
Amount_(8,_'h3') = 0.0
Amount_(8,_'h4') = 0.0
Amount_(8,_'h5') = 30.0
Amount_(9,_'h1') = 0.0
Amount_(9,_'h2') = 38.0
Amount_(9,_'h3') = 0.0
Amount_(9,_'h4') = 0.0
Amount_(9,_'h5') = 0.0
Opened_Warehouse_h1 = 1.0
Opened_Warehouse_h2 = 1.0
Opened_Warehouse_h3 = 0.0
Opened_Warehouse_h4 = 0.0
Opened_Warehouse_h5 = 1.0
The total cost = 535.32
Press any key to continue . . .
```

Figure 5: The solution of the program

Below is the summary of the output data of the Python program.

Customer	1	2	3	4	5	6	7	8	9	10		
Warehouse												status
1	30	49	23									1
2				33		40			38			1
3												0
4												0
5					19		20	30		46		1

Figure 6: The solution summary table

5 Conclusion

Overall, the Python program has built and simulated the integer linear programming model successfully. Besides, after considering through some basic steps of solving and integer linear model procedure, the final results matches the initial assumptions. This project has helped the students who implemented the program understand more deeply about the steps in solving linear programming model as well as the fundamental characteristic of mathematical modeling - the field that plays extremely important role in solving practical problem in daily life.

References

[1] Stephen Boyd, Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[2] Frank R. Giordano, William P. Fox, Steven B. Horton, *A First Course in Mathematical Modeling 5th Edition*, Brooks/Cole Cengage Learning, 2014