Optimization of Fixed Charge Problem in Python using PuLP Package

Article *in* International Journal of Control Theory and Applications · January 2017

Anand Jayakumar Arumugham
Suguna College of Engineering

IJCTA, 10(02), 2017, pp. 443-447

Optimization of Fixed Charge Problem in Python using PuLP Package

Anand Jayakumar A* Krishnaraj C and Raghunayagan P*****

Abstract : The fixed charge problem is a nonlinear programming problem of practical interest in business and industry. Yet, until now no computationally feasible exact method of solution for large problems had been developed. In this paper a numerical problem is solved using PuLP package in Python.

Keywords : Fixed charge problem, numerical problem, Python.

1. INTRODUCTION

One particularly interesting integer programming problem is the fixed – charge transportation problem. The simplicity of the problem statement and difficulty of solution makes this problem of great interest to the mathematical programming theoretician. In addition, the numerous applications in the area of distribution makes pratical solution techniques of considerable interest to the operations research practitioner. The fixed charge problem was formulated by G B Dantzig and W Hirsch in 1954. It arises in situations that involve the planning of several independent activities some or all of which have set up charges associated with them.

Fixed Charge Problems (FCP) arise in a large number of production and transportation systems. Such FCPs are typically modeled as 0-1 integer programming problems. A special case of the general FCP is Fixed Charge Transportation Problem (FCTP). The problem involves the distribution of a single commodity from a set of supply centers (sources) to a set of demand centers (destinations) such that the demand at each destination is satisfied without exceeding the supply at any source. The objective is to select a distribution scheme that has the least cost of transportation. Two kinds of costs are considered, a continuous cost which linearly increases with the amount transported between a source i and a destination j and a fixed charge which is incurred whenever a nonzero quantity is transported between source i and destination j . The fixed charge may represent toll charges on a highway; landing fees at an airport; setup costs in production systems or the cost of building roads in transportation systems. Depending on the specific applications, the importance of the fixed charge in the model will vary.

2. LITERATURE REVIEW

Leon Cooper (1), had developed a approximation method for finding optimal or near optimal solutions to the fixed charge problem. Philip Robers and Leon Cooper (2), had evaluated an approximate method of solution developed by M L Balinski for the fixed charge transportation problem by means of randomly generating test problems with known solutions. S. Molla-Alizadeh-Zavardehi et al (3), had compared the performance of genetic algorithm and simulated annealing to solve the fixed charge transportation problem. Leon B. Ellwein and Paul Gray (4), had presented a solution procedure that finds the minimal

* Dept of Mechanical Engineering SVS College of Engineering Coimbatore, India. *jay4upeople@gmail.com*

** Dept of Mechanical Engineering Karpagam College of Engineering Coimbatore, India. *krishna.kce@gmail.com*

*** Dept of Mechanical Engineering Nehru Institute of Engg and Tech Coimbatore, India. *raghuniet@gmail.com*

cost solution. They had developed an experimental computer code and presented numerical results. Paul Gray (5), presented an exact solution of this mixed integer programming problem by decomposing it into a master integer program and a series of transportation subprograms. J Haberl (6) had solved the problem through Branch and Bound algorithm. Patrick G. McKeown and Prabhakant Sinha (7), use a branch-and-bound integer programming code to solve test fixed charge problems using the setcovering formulation. Katta G. Murty (8), had described an algorithm for ranking the basic feasible solution corresponding to a linear programming problem in increasing order of linear objective function.

3. MATHEMATICAL FORMULATION

In this paper a mathematical formulation by G. Srinivasan (12), is considered. In general, we may have n demand points and m potential locations. There is a fixed cost fi of locating a facility in site *i*. There is a capacity K*i* if a facility is located in site *i*. There is a demand *dj* in point *j* and there is a transportation cost of C_{ij} between *i* and *j*. The formulation is as follows:

Let Y_i = 1 if a facility is located in site *i*.

Let X_{ij} = quantity transported from site *i* to customer *j*

The objective function minimizes the sum of the fixed cost and the transportation costs. The first constraint ensures that exactly *p* facilities are created. The second constraint ensures that items can be transported only from facilities that are created and ensures that items can be transported only from facilities that are created and that the total quantity leaving a facility is less than or equal to its capacity. The third constraint ensures that the demand of all the customers is met.

It is not absolutely necessary to fix the number of facilities created. The formulation otherwise will decide the correct number of facilities that minimize total cost.

Objective $$\sum_{i=1}^{n} f_i y_i + \sum_{i=1}^{m} C_{ij} X_{ij}$$

Subject to
$$\sum_{i=1}^{m} Y_i = p$$
$$\sum_{j=1}^{n} X_{ij} \leq K_i Y_i$$
$$\sum_{i=1}^{m} X_{ij} \geq d_j$$
$$Y_i = 0.1$$
$$X_{ij} \geq 0$$

4. NUMERICAL PROBLEM

Consider a network as shown in Fig 1 below. The fixed costs of locating facilities in the three potential locations are as follows:

Location 1 = Rs. 5,000,000

Location 2 = Rs. 4,000,000

Location 3 = Rs. 5,500,000

The capacities of the three locations are 1000000, 800000 and 1250000. The demand at the eight demand points are 200000 for the first four points and 250000 for the remaining points. The unit transportation costs are given in the Table 1 below.

Table 1
Unit Transportation Cost

	D1	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	*D8*
Location 1	4	5	5	4	4	4.2	3.3	5
Location 2	2.5	3.5	4.5	3	2.2	4	2.6	5
Location 3	2	4	5	2.5	2.6	3.8	2.9	5.5

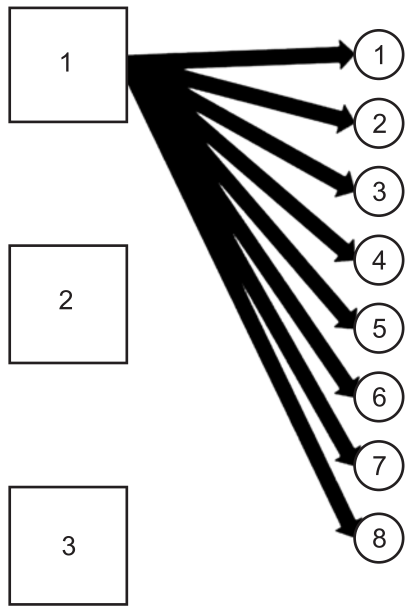

Figure 1: Network for Fixed Charge Problem

5. PYTHON PROGRAM

 # import libraries

```
from pulp import *
# variable assignment
      facility = ['f1','f2','f3']
      location = ['d1','d2','d3','d4','d5','d6','d7','d8']
             f = dict(zip(facility, [5000000, 4000000, 5500000]))
             K = dict(zip(facility, [1000000, 800000, 1250000]))
             D = dict(zip(location, [200000, 200000, 200000, 200000,
                 250000, 250000, 250000, 250000]))
             C = dict(zip(facility,[dict(zip(location, [4, 5, 5, 4, 4, 4.2, 3.3, 5])),
                 dict(zip(location, [2.5, 3.5, 4.5, 3, 2.2, 4, 2.6, 5])),
                 dict(zip(location, [2, 4, 5, 2.5, 2.6, 3.8, 2.9, 5.5]))]))
             p = 2
# decision variables   X = Lp Variable.dicts ('X%s%s', (facility, location),
           cat = 'Continuous',
      lowBound = 0,
      up Bound = None)
             Y = LpVariable.dicts('Y%s', facility,
           cat = 'Binary',
```

```
                                    lowBound = 0,
                                     upBound = 1)
# create the LP object, set up as a MINIMIZATION problem
                                        prob = LpProblem('Fixed Charge', LpMinimize)
                                        tmp1 = sum(f [i] * Y[i] for i in facility)
                                        tmp2 = sum(sum(C[i][j] * X[i][j] for j in location) for i in facility)
                                      prob + = tmp1 + tmp2
# setup constraints                   prob + = sum(Y[i] for i in facility) == p
for i in facility:                    prob + = sum(X[i][j] for j in location) <= K[i]*Y[i]
for j in location:                    prob + =  sum(X[i][j] for i in facility) >= D[j]
# save the model to a lp file
prob.writeLP("fixed-charge.lp")
# view the model
print(prob)
# solve the model
prob.solve()
print("Status:",LpStatus[prob.status])
print("Objective: ",value(prob.objective))
for v in prob.variables():
print (v.name , "=", v.varValue)
```

6. COMPUTATIONAL EFFICIENCY

An intel 2nd generation core i5 processor was used with 4GB RAM and windows 7 operating system. Python 3.5.2 :: Anaconda 4.2.0 was used. PuLP package 1.6.1 was used. The default solver was CBC.

The problem was solved in less than 1 second.

7. RESULT AND DISCUSSION

Optimal Transportation is shown below

Table 2
Optimal Transportation

	D1	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	*D8*
Loc 1	0	0	0	0	0	0	0	0
Loc2	0	200K	200K		150K			250K
Loc 3	200K			200K	100K	250K	250K	

From Location 1 no units are transported

From Location 2

200K units are transported to Demand Point 2
200K units are transported to Demand Point 3
150K units are transported to Demand Point 5
250K units are transported to Demand Point 8

From Location 3

200K units are transported to Demand Point 1
200K units are transported to Demand Point 4
100K units are transported to Demand Point 5
250K units are transported to Demand Point 6
250K units are transported to Demand Point 7

8. CONCLUSION

Thus in this paper we have found the optimal transportation for each segment using Python PuLP package. The total cost is Rs. 1,55,15,000.

9. REFERENCES

1. Leon Cooper, ―The Fixed Charge Problem – I: A New Heuristic Method‖, Computers and Maths with Applications, Vol 1, pp 89-95, 1975.
2. Philip Robers and Leon Cooper, ―A Study of the Fixed Charge Transportation Problem‖, Computers and Maths with Applications, , Vol 2, pp 125-135, 1976.
3. S. Molla-Alizadeh-Zavardehi, A. Mahmoodirad and M. Rahimian, ―Step Fixed Charge Transportation Problems via Genetic Algorithm‖, Indian Journal of Science and Technology, Vol 7 (7), pp 949-954, July 2014.
4. Leon B. Ellwein and Paul Gray, ―Solving Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints‖, A I I E Transactions, Vol 3, No 4, pp 290-298, 1971.
5. Paul Gray, ―Technical Note—Exact Solution of the Fixed-Charge Transportation Problem‖, Operations Research, Vol 19, No 6, pp 1529-1538, 1971.
6. J Haberl, ―Exact Algorithm for Solving a Special Fixed – Charge Linear Programming Problem‖, Journal of Optimization Theory and Applications, Vol 69, No 3, pp 489-528, June 1991.
7. Patrick G. McKeown and Prabhakant Sinha, ―An Easy Solution for a Special Class of Fixed Charge Problems‖, Naval Research Logistics, Vol 27, Issue 4, pp 621-624, Dec 1980.
8. Katta G. Murty, ―Solving the Fixed Charge Problem by Ranking the Extreme Points‖, Operations Research, Vol 16, No2, pp 268-279, 1968.
9. Krishnaraj, C., A. Anand Jayakumar, S. Deepa Shri, ―Solving Supply Chain Network Optimization Models Using LINGO‖, International Journal of Applied Engineering Research, Vol 10, No 19, pp14715-14718, 2015
10. Anand Jayakumar, A., C. Krishnaraj, ―Solving Supply Chain Network Gravity Location Model Using LINGO‖, International Journal of Innovative Science Engineering and Technology‖, Vol 2, No 4, pp 32-35, 2015.
11. Anand Jayakumar A, Krishnaraj C, Aravinth Kumar A, ―LINGO Based Supply Chain Network Design‖, Journal of Applied Sciences Research, Vol 11, No 22, pp 19-23, Nov 2015.
12. G Srinivasan, ―Quantitative Models in Operations Management‖, PHI.