CentraleSupélec

MASTER DSBI
DECISION MODELING
PRACTICAL WORKS

LINEAR PROGRAMING & PREFERENCES

1 Introduction to Linear programming with Python

Linear programming is the technique used to maximize or minimize a function. The idea is to optimize a complex function by best representing them with linear relationships. In simpler terms, we try to optimize (to maximize or minimize) a function denoted in linear terms and bounded by linear constraints.

Use case - Miracle worker

Let's try to formalize an use-case and carry it forward throughout the article. Suppose you are a magical healer and you goal is to heal anyone who asks for help. The more you are able to heal someone, the better. Your secret behind the healing is 2 medicines, each of which uses special herbs. To create one unit of medicine 1, you need 3 units of herb A and 2 units of herb B. Similarly, to create one unit of medicine 2, you need 4 and 1 units of herb A and B respectively. Now medicine 1 can heal a person by 25 unit of health (whatever it is) and medicine 2 by 20 units. To complicate things further, you only have 25 and 10 units of herb A and B at your disposal. Now the question is, how many of each medicine will you create to maximize the health of the next person who walks in ?

Modeling the problem

First let's try to identify the objective (what we want to do and how) and constraint (the bounding functions) of the stated problem.

As it's clear from the problem, we want to increase the health by as many units as possible. And medicines are the only thing which can help us with it. What we are unsure of, is the amount to each medicines to create. Going by a mathematician's logic, lets say we create x units of medicine 1 and y units of medicine 2. Then the total health restored can be given by,

$$25 \times x + 20 \times y = \text{Health Restored}$$

Where

$x =$ Units of medicine 1 created
$y =$ Units of medicine 2 created

This is the objective function, which we want to maximize. Now both the medicines are dependent on the herbs which we have in finite quantity. Let's understand the constraints. If we create x and y units of medicine 1 and 2,

- We use $3x + 4y$ units of herb A. But we only have 25 units of it, hence the constraint is, our total usage of herb A should not exceed 25, denoted by,
$$3x + 4y \leq 25$$
- We use $2x + 1y$ units of herb B. We have 10 units of it, hence the constraint is, our total usage of herb B should not exceed 10, denoted by,
$$2x + y \leq 10$$
- Also the amount of medicines created cannot be negative (doesn ?t make sense) hence, they should be equal or greater than zero, denoted by,
$$x \geq 0; \; y \geq 0$$

Solution - Graphical representation

One way to solve the problem is by representing it into graph which requires plotting the functions, constraints and finding any point of interest. Let's plot our functions and see what we can infer from it.

The point of intersection, as obvious, from the plot is $(3, 4)$, which says, If we create 3 units of medicine 1 and 4 units of medicine 2, considering the constraints on herbs, we are best equipped to heal the next patient. Intuitively, we wanted to find a solution which satisfy all of our constraints. As the constraint have few variables (only x and y), transforming the problem into graph of lower dimension, we can visualize it as a 2D plot. With constraints on herbs being transformed into lines, our solution now transforms into a point of intersection. To make matter things better, it was on the positive quadrant, satisfying our 3rd constraint.

But what about problems with larger variable count ? or with more constraints ? wouldn't it be difficult to plot and visualize them all ? And do you always want to plot and solve these kinds of problems ? Lets try to leverage modern computing prowess.

Solution - Python Programming

Python has a nice package named PuLP which can be used to solve optimization problems using Linear programming. To start with we have to model the functions as variables and call PuLP ?s solver module to find optimum values. Here it goes,

```python
1 # import PuLP
2 from pulp import *
3
4 # Create the 'prob' variable to contain the problem data
5 prob = LpProblem("The Miracle Worker", LpMaximize)
6
7 # Create problem variables
8 x=LpVariable("Medicine_1_units",0,None,LpInteger)
9 y=LpVariable("Medicine_2_units",0, None, LpInteger)
10
11 # The objective function is added to 'prob' first
12 prob += 25*x + 20*y, "Health restored; to be maximized"
13 # The two constraints are entered
14 prob += 3*x + 4*y <= 25, "Herb A constraint"
15 prob += 2*x + y <= 10, "Herb B constraint"
16
17 # The problem data is written to an .lp file
18 prob.writeLP("MiracleWorker.lp")
19
20 # The problem is solved using PuLP's choice of Solver
21 prob.solve()
```

Lets try to understand the code,

- Line 1-2 : We import the PuLP package.
- Line 4-5 : We define our problem by giving a suitable name, also specifying that our aim is to maximize the objective function.
- Line 7-9 : Here, we define LpVariable to hold the variables of objective functions. The next arguments specifies the lower and upper bound of the defined variable. As per 3rd constraint, its non-negative, hence made second argument as 0 and 3rd as None (in-place of infinity). The last argument says if the required output is discrete or continuous.
- Line 11-12 : Denotes the objective function in terms of defined variables, along with a short description.
- Line 13-15 : Entering the herb's constraints in terms of variables.
- Line 17-18 : Output the objective and constraints into a file for portability and reuse. (Next time just load and run)
- Line 20-21 : This calls the solver module and optimizes the functions.

Now lets review the output,

```
# The status of the solution is printed to the screen
print("Status:", LpStatus[prob.status])
# Output=
# Status: Optimal

# Each of the variables is printed with it's resolved optimum value
for v in prob.variables():
    print(v.name, "=", v.varValue)
# Output=
# Medicine_1_units = 3.0
# Medicine_2_units = 4.0

# The optimised objective function value is printed to the screen
print("Total Health that can be restored = ", value(prob.objective))
# Output=
# Total Health that can be restored =  155.0
```

As evident, the solution was optimum and similar to what we got from graphical representation.

More details about the use of the Pulp package can be found at `http://benalexkeen.com/linear-programming-with-python-and-pulp/`.

Two others Python modules could be used to solve linear programs :

- Scipy : `https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.optimize.linprog.html`
- Numpy : `https://www.pythonprogramming.in/numpy-tutorial-with-examples-and-solutions.html`

Exercise

A toy manufacturing organization manufactures two types of toys A and B. Both the toys are sold at 25 € and 20 € respectively. There are 2000 resource units available every day from which the toy A requires 20 units while toy B requires 12 units. Both of these toys require a production time of 5 minutes. Total working hours are 9 hours a day. What should be the manufacturing quantity for each of the pipes to maximize the profits ?

2 How to visit Paris ?

John Doe, an American researcher, goes to Paris to present an article at a conference. It is his first time in the French capital. He arrives on Monday and he intends to spend the whole week at the conference. However, the conference ends on Friday, and Mr. Doe leaves only on Sunday in the end evening. So he has the opportunity to visit the city during the weekend. He gives himself 12 hours on the two days to fully visit Paris with a maximum budget of 65 €.

By doing his own research (via some social medias for instance), he finds that it is interesting to visit : *La* Tour Eiffel (TE)*, Le Musée du louvre (ML), l'Arc de triomphe (AT), le Musée d'Orsay (MO), le Jardin des tuileries (JT), les Catacombes (CA), le Centre Pompidou*

(CP), la Cathédrale Notre Dame de Paris (CN), la Basilique du Sacré-Coeur (BS), la Sainte Chapelle (SC), La Place de la Concorde (PC), la Tour Montparnasse (TM) and l'Avenue des Champs-Elysées (AC).

For each of these places or monuments, he obtained the following information which could help him to make his choices :

- the duration of a visit (in hours)
- the global appreciation of Internet users (given by a qualitative score : from ⋆ to ⋆⋆⋆⋆⋆)
- the price for a single visit.

	Duration (in hours)	Appreciations	Price (in euros)
TE	9/2	⋆⋆⋆⋆⋆	15.50
ML	3	⋆⋆⋆⋆	12
AT	1	⋆⋆⋆	9.50
MO	2	⋆⋆	11
JT	3/2	⋆⋆⋆	0
CA	2	⋆⋆⋆⋆	10
CP	5/2	⋆	10
CN	2	⋆⋆⋆⋆⋆	5
BS	2	⋆⋆⋆⋆	8
SC	3/2	⋆	8.50
PC	3/4	⋆⋆⋆	0
TM	2	⋆⋆	15
AC	3/2	⋆⋆⋆⋆⋆	0

Mr. Doe was also able to determine the distances (in kms of walking) between two sites :

Sites	TE	ML	AT	MO	JT	CA	CP	CN	BS	SC	PC	TM	AC
TE	0	3.8	2.1	2.4	3.5	4.2	5.0	4.4	5.5	4.2	2.5	3.1	1.9
ML		0	3.8	1.1	1.3	3.3	1.3	1.1	3.4	0.800	1.7	2.5	2.8
AT			0	3.1	3.0	5.8	4.8	4.9	4.3	4.6	2.2	4.4	1.0
MO				0	0.900	3.1	2.5	2.0	3.9	1.8	1.0	2.3	2.1
JT					0	4.2	2.0	2.4	2.7	2.0	1.0	3.4	2.1
CA						0	3.5	2.7	6.5	2.6	3.8	1.3	4.9
CP							0	0.850	3.7	0.900	2.7	3.4	3.8
CN								0	4.5	0.400	2.8	2.7	3.9
BS									0	4.2	3.3	5.7	3.8
SC										0	2.5	2.6	3.6
PC											0	3.0	1.2
TM												0	2.1
AC													0

Our aim is to help Mr Doe to choose some sites to visit during these two days. You can use some linear programming models or another approach to do this. Use a python code to answer the following questions :

1. It is assumed that Mr. Doe gives equal importance to each tourist site, and he wants to visit the maximum number of sites. Which list(s) of places could you recommend to him ? This solution will be called **ListVisit 1**.
2. Actually, Mr. Doe has some preferences among these tourist sites and he expresses them as follows :
   - **Preference 1 :** If two sites are geographically very close (within a radius of 1 km of walking), he will prefer to visit these two sites instead of visiting only one.
   - **Preference 2 :** He absolutely wants to visit the Eiffel Tower (TE) and Catacombes (CA).
   - **Preference 3 :** If he visits Notre Dame Cathedral (CN) then he will not visit the Sainte Chapelle (SC).
   - **Preference 4 :** He absolutely wants to visit Tour Montparnasse (TM).
   - **Preference 5 :** If he visits the Louvre (ML) Museum then he must visit the Pompidou Center (CP).

   (a) For each of the five preferences above, suggest to Mr. Doe, one or more lists of tourist sites to visit. Are the obtained lists different from the solution **ListVisit 1** ?

(b) If Mr. Doe wishes, at the same time, to take into account **Preference 1** and **Preference 2**, which list(s) would you recommend to him ?

(c) If Mr. Doe wishes, at the same time, to take into account **Preference 1** and **Preference 3**, which list(s) would you recommend to him ?

(d) If Mr. Doe wishes, at the same time, to take into account **Preference 1** and **Preference 4**, which list(s) would you recommend to him ?

(e) If Mr. Doe wishes, at the same time, to take into account **Preference 2** and **Preference 5**, which list(s) would you recommend to him ?

(f) If Mr. Doe wishes, at the same time, to take into account **Preference 3** and **Preference 4**, which list(s) would you recommend to him ?

(g) If Mr. Doe wishes, at the same time, to take into account **Preference 4** and **Preference 5**, which list(s) would you recommend to him ?

(h) If Mr. Doe wishes, at the same time, to take into account **Preference 1**, **Preference 2** and **Preference 4**, which list(s) would you recommend to him ?

(i) If Mr. Doe wishes, at the same time, to take into account **Preference 2**, **Preference 3** and **Preference 5**, which list(s) would you recommend to him ?

(j) If Mr. Doe wishes, at the same time, to take into account **Preference 2**, **Preference 3**, **Preference 4** and **Preference 5**, which list(s) would you recommend to him ?

(k) If Mr. Doe wishes, at the same time, to take into account **Preference 1**, **Preference 2**, **Preference 4** and **Preference 5**, which list(s) would you recommend to him ?

(l) If Mr. Doe wishes, at the same time, to take into account **Preference 1**, **Preference 2**, **Preference 3**, **Preference 4** and **Preference 5**, which list(s) would you recommend to him ?

(m) Is the solution **ListVisit1** close to one or more solutions found above with the preferences ? Why ?

3. Let be $\succsim$ the ranking of the touristic sites obtained from the appreciations observed (see the column “Appreciation” of the Table above). Compare this ranking to recommendations lists elaborated in the previous question. There are some differences ?

4. Now, we assumed that Mr. Doe has no restriction (about hour, budget, distance between sites). Nevertheless, he wishes to make choices by maximizing the criterion “Appreciations” and minimizing the two criteria “Duration” and “Price”. In addition, it gives an importance to each of these three criteria by assigning them the following weights :

	Duration (in hours)	Appreciations	Price (in euros)
Weight	4	2	3

Could you provide to him a recommendation of places to visit ? Justify the approach you used (your approach could be different to other student(s), do not be afraid). Could you implement your approach in order to recommend some touristic visits in Paris ?