**<https://www.section.io/engineering-education/simplex-method-in-python/>**

[**https://www.section.io/engineering-education/authors/stanley-juma/**](https://www.section.io/engineering-education/authors/stanley-juma/)

**Getting Started with the Simplex Method Algorithm**

The simplex method is an algorithm used in linear programming problems to determine the optimal solution for a given optimization problem.

For example, this method is used when a linear optimization problem is subjected to inequality constraints. In this article, we shall look at how this algorithm work.

To understand how this algorithm works, let us consider the following problem:

* A bicycle manufacturer makes touring, racing, and model bicycles.
* These bicycles are made of aluminum and steel.
* The company has 91800 steel units and 42000 aluminum units.
* The racing, touring, and mountain models require 17, 27, and 15 steel units and 12, 21, and 15 aluminum units.
* A company makes 8 dollars per racing bike, 12 dollars per touring bike, and 22 dollars per mountain bike.

In order to maximize profits, how many of each type should be produced? What is the maximum profit that can be made?

Define our variables:

* R: Number of racing bikes.
* T: Number of touring bikes.
* M: Number of mountain bikes.

We are optimizing the profit function.

P = 8R + 12T + 22M

Since we have two categories of resources with the respective possible number of production units made out of them, we will thus have two constraints.

These constraints are:

17R+27T+34M≤91800

12R+21T+15M≤42000

Additionally, we need to understand that any product produced can be zero when nothing is produced or greater than one when at least a single product is produced. Therefore on top of the above two constraints, we will add the following as well:

R≥0,T≥0,M≥0

Now, we can combine the following and develop the following optimization problem:

Maximize −8R−12T−22M+P=0

 17R+27T+34M≤91800

 12R+21T+15M≤42000

R≥0,T≥0,M≥0

Case 1

import numpy as np

import scipy as sp

# Get matrices

c = [-8, -12, -22]

A = [[17, 27, 34], [12, 21, 15]]

b = [91800, 42000]

# define the upper bound and the lower bound

R = (0, None)

T = (0, None)

M = (0, None)

# Implementing the Simplex Algorithm

from scipy.optimize import linprog

# Solve the problem by Simplex method in Optimization

res = linprog(c, A\_ub=A, b\_ub=b,

bounds=(R, T, M), method='simplex', options={"disp": True})

# linear programming p[roblem

print(res)

This program returns:

Optimization terminated successfully.

Current function value: -59400.000000

Iterations: 3

con: array([], dtype=float64)

fun: -59400.0

message: 'Optimization terminated successfully.'

nit: 3

slack: array([ 0., 1500.])

status: 0

success: True

x: array([ 0., 0., 2700.])

From this output, it is clear that the optimal action is to build 0 touring bikes, 0 racing bikes, and 2700 mountain bikes. If this action is practiced, the company will realize an optimal profit of $59,400 dollars.

Now, let us proceed and solve the second problem.

Suppose we are given the following linear programming problem:

*Using the Simplex method:* *Maximize*, Z=30x+40y

Subject to,

x+y≤50

 4x++2y≤150

50x+100y≤4500

 x≥0, y≥0

From this problem, we can have the following three arrays:

c=[−30,−40]

A=[11 42 50100]

b=[50,150,4500]

Case 2

import numpy as np

import scipy as sp

# Get matrices

c = [-30, -40]

A = [[1, 1], [4, 2], [50, 100]]

b = [50, 150, 4500]

# define the upper bound and the lower bound

x = (0, None)

y = (0, None)

# Implementing the Simplex Algorithm

from scipy.optimize import linprog

# Solve the problem by Simplex method in Optimization

res = linprog(c, A\_ub=A, b\_ub=b, bounds=(x, y),

method='simplex', options={"disp": True})

# linear programming p[roblem

print(res)

Executing this program yields:

Optimization terminated successfully.

Current function value: -1900.000000

Iterations: 3

con: array([], dtype=float64)

fun: -1900.0

message: 'Optimization terminated successfully.'

nit: 3

slack: array([ 0., 30., 0.])

status: 0

success: True

x: array([10., 40.])

In the result, the value of the objective function, i.e., fun, is -1900. This value is computed for the minimization problem.

In the case of a maximization problem, we omit the negative sign. Therefore, the solution for our maximization is 1900. Also, from the results, we can see that the value for the x and y that will lead to an optimal solution are 10 and 15, respectively.