Simplex Algoritmh

This is the standard technique in "*Linear Programming*" to solve an optimization problem, in which you have an *"objective function"* and several functions expressed as inequalities called *"restrictions".*

In this way the vertices are tested as possible solutions.

Algebraically, this procedure consists in finding a possible basic solution, then with it, generating new possible basic solutions, so that with each of them, the value of the objective function increases.[[1]](#footnote-1)

In 1947, George B. Dantzig developed a technique to solve linear programs,[[2]](#footnote-2) inspired by the input output Matrix created by the nobel prize in economics Wassily Leontief.[[3]](#footnote-3)

Algorithm:

Subject to:

To solve a linear programming model using the Simplex method the following steps are necessary:

The simplex algorithm is an iterative process that starts from the origin of the n-D vector space  , and goes through a sequence of vertices of the polytope to eventually arrive at the optimal vertex at which the objective function is maximized. To do so, we first convert the standard form of the problem into a tableau, a table of  columns and  rows:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Basic variables* | *X1* | *X2* | *…* | *Xn* | *s1* | *s2* | *…* | *Sm* | *Basic Solution* |
| *s1* | a11 | a12 | … | a1n | 1 | 0 | … | 0 | *b1* |
| *s2* | a21 | a22 | … | a2n | 0 | 1 | … | 0 | *b2* |
| *…* | … | … | … | … | … | … | … | … | *…* |
|
| *Sm* | am1 | am2 | …. | amn | 0 | 0 | …. | 1 | *bm* |
| Z | -C1 | -C2 | … | -Cm | 0 | 0 | … | 0 | 0 |

 We select  in the “j” column of the tableau if it is most heavily weighted by  ( (- is most negative), as this   will increase   more than any other  .

Choose the constraint with least value .

Divide all elements in the “i” row of A by , so that =1.

Subtract , times the “i” row from the “k” row so that ,  .

So  becomes 1 while all other elements in the “j” column become zero. This is Gaussian elimination based on pivoting.

Repeat the steps above until all elements in the last row are non-negative.[[4]](#footnote-4) In this way the optimal feasible basic solution is found.

When there is no initial basic solution, the methods of the Big- M or the two-phase method are usually used[[5]](#footnote-5),through the use of artificial variables.

Code (Python):

Consider the following linear programming system of a company that wants to maximize the production of 2 goods, and and has the function defined:

and is subject to some restrictions of space, time and financial resources and defines its restrictions in the form of equations linear:

Using the scipy.optimize library in python:

from scipy.optimize import linprog

c=[-4,-3]

A\_ub=[[2,3],[-3,2],[0,2],[2,1]]

b\_ub=[6,3,5,4]

res=linprog(c,A\_ub,b\_ub,bounds=(0,None))

print(res)

print("The optimal value is= ",(res.fun)\*-1,"\nX:", "Where x1 = ",np.around(res.x[0],1), "and x2 = ",res.x[1])

Result:

con: array([], dtype=float64)

fun: -9.0 message: 'Optimization terminated successfully.'

nit: 5

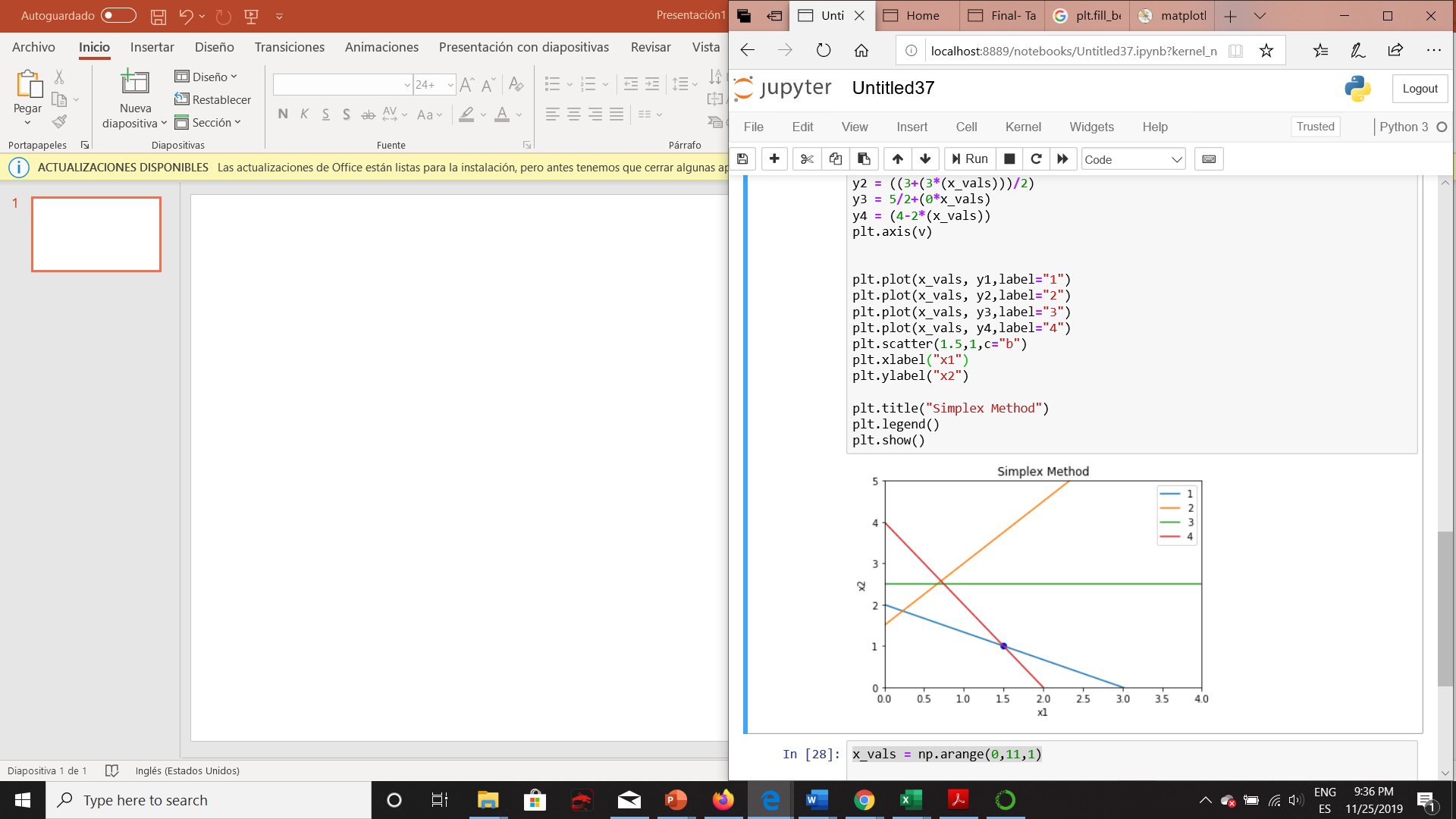
slack: array([0. , 5.5, 3. , 0. ])

status: 0

success: True x: array([1.5, 1. ])

The optimal value is= 9.0 X: Where x1 = 1.5 and x2 = 1.0

The graph:



x\_vals = np.arange(0,11,1)

v=[0,4,0,5]

y1 = (((6-2\*(x\_vals))/3))

y2 = ((3+(3\*(x\_vals)))/2)

y3 = 5/2+(0\*x\_vals)

y4 = (4-2\*(x\_vals))

plt.axis(v)

plt.plot(x\_vals, y1,label="1")

plt.plot(x\_vals, y2,label="2")

plt.plot(x\_vals, y3,label="3")

plt.plot(x\_vals, y4,label="4")

plt.scatter(1.5,1,c="b")

plt.title("Simplex Method")

plt.legend()

plt.show()

The highlighted blue area represents the set of basic feasible solutions, while the blue dot represents the point where the z function takes its maximum value.

bibliography or cybergraphy

<http://verso.mat.uam.es/~joser.berrendero/cursos/Matematicas-IO/io-tema3-16.pdf>

<http://www.mi.fu-berlin.de/wiki/pub/ABI/LinearOptimizationWS11/Simplex.pdf>

<http://repobib.ubiobio.cl/jspui/bitstream/123456789/282/3/Chavez_Abello_Rodrigo.pdf>

1. <http://repobib.ubiobio.cl/jspui/bitstream/123456789/282/3/Chavez_Abello_Rodrigo.pdf> [↑](#footnote-ref-1)
2. <https://math.mit.edu/~goemans/18310S15/lpnotes310.pdf> [↑](#footnote-ref-2)
3. <http://repobib.ubiobio.cl/jspui/bitstream/123456789/282/3/Chavez_Abello_Rodrigo.pdf> [↑](#footnote-ref-3)
4. <http://fourier.eng.hmc.edu/e176/lectures/NM/node32.html> [↑](#footnote-ref-4)
5. <http://repobib.ubiobio.cl/jspui/bitstream/123456789/282/3/Chavez_Abello_Rodrigo.pdf> [↑](#footnote-ref-5)