



Luis Fernando Flores Diaz

Evidencia puntos extra P1: Solución con Simplex a un PPL.

UA: Investigación de Operaciones

Se ejecutará el programa en Python que recibe como argumentos lo siguiente:

 [LuisDiaz-epsilon / Investigacion-Operaciones](#) [Public](#)

nfo -- Naturaleza de la funcion objetivo ('max' o 'min')
fo -- arreglo numpy con los coeficientes de la función objetivo
s_a -- matriz numpy con los coeficientes de las restricciones
b -- arreglo numpy con los términos independientes de las restricciones
signos -- lista de operadores de las restricciones ('>=', '<=', o '=')
nv -- Naturaleza de las variables que recibe solo ('>=', '<=', "eR")

Paquetes y funciones:

[estándar_form.py](#)

- ▶ **forma_estandar(nfo, fo, s_a, b, signos, nv)**
Descripción: Retorna la forma estandar de un PPL.

[simplex_format.py](#)

- ▶ **simplex_tab_format(fo_std, s_a_std, b_std)**
Descripción: Recibe un PPL en su forma estandar y prepara el tablero inicial.

[simples_operation.py](#)

- ▶ **simplex_method_p1_canonic(tablero)**
Descripción: Para el caso unico cuando se recibe el tablero se requiere conocer sus variables basicas y pivotar
- ▶ **criterio_optimalidad(tablero)**
Verificamos si cumple el criterio de optimalidad
- ▶ **operar_tablero(tablero)**
Opera el tablero para las variables que necesitan entrar y salir.

Se ejecuto con el siguiente PPL

$$\begin{aligned} P: \min z &= -3x + 2y \\ s. a \\ x + y &\leq 5 \\ 2x + y &\leq 15 \\ x, y &\geq 0 \end{aligned}$$

Utilizando los paquetes escritos en Python en un archivo para ejecutarlos se representa de la siguiente manera:

```
import numpy as np
import standar_form
import simplex_format
import simplex_operation

fo = np.array([-1, 2,])
s_a = np.array([[1, 2,], [2, 1]])
b = np.array([0, 5, 15])
signos = ["<=", "<="]
#Naturaleza de las variables
nv = [">=", ">="]

"""
Paso 1 colocar en forma canonica la matriz inciial con simplex_method_p1_canonic
Paso 2 verificar con criterio_optimalidad en caso de cumplir finzalir, en caso de fallas
conrinuar con paso 3
Paso 3 operar_tablero
"""

fo_std, s_a_std, b_std, nfo_std, nv_std= standar_form.forma_estandar("max", fo, s_a, b,
signos, nv)

#Darle formato al tablero
tablero_inicial = simplex_format.simplex_tab_format(fo_std, s_a_std, b_std)

#operar el tablero
tablero = simplex_operation.simplex_method_p1_canonic(tablero_inicial)
tablero, flag_c_o = simplex_operation.criterio_optimalidad(tablero)

while not flag_c_o:
    tablero, flag_c_o = simplex_operation.criterio_optimalidad(tablero)
    if not flag_c_o:
        tablero = simplex_operation.operar_tablero(tablero)
        tablero, flag_c_o = simplex_operation.criterio_optimalidad(tablero)
    else:
        break
```

A continuación se muestra la impresión por pantalla:

```
C:\Users\Win\Desktop\Investigacion-Operaciones>py test1.py
```

```
Matriz inicial:
```

```
[[1 2]
 [2 1]]
```

```
Procedimiento:
```

```
Se agrego una variable de Holgura
```

```
[[1. 2. 1.]
 [2. 1. 0.]]
```

```
Se agrego una variable de Holgura
```

```
[[1. 2. 1. 0.]
 [2. 1. 0. 1.]]
```

```
-----Forma estandar:
```

```
min [ 1 -2]
```

```
s.a
```

```
[[1. 2. 1. 0.]
 [2. 1. 0. 1.]]
```

```
b: [ 0 5 15]
```

```
Naturaleza de las variables
```

```
['x_0=> 0', 'x_1=> 0', 'x_3=> 0', 'x_4=> 0']
```

```
[ 1 -2  0]
```

```
[ 1 -2  0]
```

```
[ 1 -2  0 0]
```

```
Primer tablero simplex
```

```
[[ 1. -2.  0.  0.  0.]
 [ 1.  2.  1.  0.  5.]
 [ 2.  1.  0.  1. 15.]]
```

```
Se encontraron las siguientes variables basicas.
```

```
[2 3]
```

```
Se tiene que hay que pivotar lo siguiente:
```

```
[col, valor contenido]
```

```
[]
```

```
Ahora si, el tablero ya se encuentra en forma canonica
```

```
[[ 1. -2.  0.  0.  0.]
 [ 1.  2.  1.  0.  5.]
 [ 2.  1.  0.  1. 15.]]
```

```
No se cumple el criterio de optimalidad
```

```
No se cumple el criterio de optimalidad
```

```
El orden del vector de VB:
```

```
[2, 1, 3]
```

```
Fila que sale: 1
```

```
Entra x_0 y Sale x_2
```

```
Se actualiza vector de VB:
```

```
[0, 1, 3]
```

```
[0, 1, 3]
```

```
Tablero operado
```

```
[[ 0. -4. -1.  0. -5.]
 [ 1.  2.  1.  0.  5.]
 [ 0. -3. -2.  1.  5.]]
```

```
Cumple el criterio de optimalidad
```

```
C:\Users\Win\Desktop\Investigacion-Operaciones>
```

Se obtiene el tablero final:

```
[[ 0. -4. -1.  0. -5.]  
 [ 1.  2.  1.  0.  5.]  
 [ 0. -3. -2.  1.  5.]]
```

Donde la primera fila corresponde a la Función Objetivo, la ultima columna a los valores independientes vector b y lo no mencionado los coeficientes de las restricciones.

Se concluye con el vector VB [0,1,3] Donde se observa que:

$x^* = (x, y) = (5, 0)$
por lo que
 $z^* = -15$

Librerías empleadas:

Uso de matrices: <https://numpy.org/>

Coloreado de consola: <https://pypi.org/project/colorama/> <https://pypi.org/project/termcolor/>