Tarea 11 Optimización

Roberto Vásquez Martínez
Profesor: Joaquín Peña Acevedo

22/Mayo/2022

1 Ejercicio 1 (3 puntos)

Usando alguna librería de Python para resolver problemas de programación lineal, escriba y resuelva el problema de la Tarea 10:

$$\begin{array}{ll} \max & x_1 + x_2 \\ & 50x_1 + 24x_2 \leq 2400 \\ & 30x_1 + 33x_2 \leq 2100 \\ & x_1 \geq 45 \\ & x_2 \geq 5 \end{array}$$

1. Cambie el problema para que todas las desigualdes sean de la forma

$$\mathbf{Ax} \leq \mathbf{b}.$$

2. Construya los vectores $\mathbf{b}, \mathbf{c}$ y la matriz $\mathbf{A}$ y resuelva el problema con la librería.

3. Imprima un mensaje que indique si se encontró la solución, y en ese caso imprima :

- la solución $\mathbf{x}$,
- el valor de la función objetivo,
- las variables de holgura,

4. Calcule los errores

$$E_x = \sum_{x_i<0} |x_i|.$$

$$E_{b-Ax} = \sum_{(b-Ax)_i<0} |(b-Ax)_i|$$

Es decir, se suman las componentes de $\mathbf{x}$ que no cumplen la condición $\mathbf{x} \geq \mathbf{0}$ y las componentes que no cumplen con $\mathbf{Ax} \leq \mathbf{b}$.

5. Defina la tolerancia $\tau = \sqrt{\epsilon_m}$, donde ϵ_m es el épsilon de la máquina. Si $E_x < \tau$ imprima un mensaje de que se cumple la condición de no negatividad, y si $E_{b-Ax} < \tau$ imprima un mensaje de que se cumplen las restricciones de desigualdad.

1.1 Solución:

En primer lugar, hallaremos $\mathbf{A}, \mathbf{b}$ de forma que las restricciones son de la forma $\mathbf{Ax} \leq \mathbf{b}$. Notemos que las restricciones son equivalentes a la siguiente lista de desigualdades

$$
\begin{aligned}
&50x_1 + 24x_2 \leq 2400 \\
&30x_1 + 33x_2 \leq 2100 \\
&-x_1 \leq 45 \\
&-x_2 \leq 5
\end{aligned}
$$

Si $\mathbf{c} = (-1, -1)^T$, $\mathbf{b} = (2400, 2100, -45, -5)^T$ y

$$
\mathbf{A} = \begin{pmatrix} 50 & 24 \\ 30 & 33 \\ -1 & 0 \\ 0 & -1 \end{pmatrix},
$$

entonces el problema de optimización que queremos resolver es

$$
\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} \quad \text{sujeto a} \quad \mathbf{Ax} \leq \mathbf{b},
$$

que es lo que queríamos obtener el numeral 1.

Para el numeral 2 y 3 construimos los vectores y matrices correspondiente para posteriormente aplicar la librería `linprog` de `scipy` obtener la solución así como los demás datos requeridos. Esto lo hacemos en la siguiente celda de código.

[1]:
```
from scipy.optimize import linprog
import scipy


# Coeficientes de la funcion objetivo
obj = [-1.0, -1.0]

# Coeficientes del lado izquierdo de las desigualdades del tipo "menor o igual a"
lhs_ineq = [[50.0, 24.0],
            [30.0, 33.0],
            [-1.0, 0.0],
            [0.0,-1.0]]

# Coeficientes del vector del lado derecho de las desigualdades del tipo "menor␣
↪o igual a"
rhs_ineq = [2400, 2100, -45,-5]


# Cotas de las variables
bnd = [(0, scipy.inf),  # cotas para x1
       (0, scipy.inf)]  # cotas para x2
```

```
opt_ineq = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq, bounds=bnd,
              method="simplex")

print('\nResultado del proceso:', opt_ineq.message)
if opt_ineq.success:
    print('Valor de la función objetivo:', opt_ineq.fun)
    print('Solución:\n', opt_ineq.x)
    print('\nVariables de holgura:\n', opt_ineq.slack)
```

```
Resultado del proceso: Optimization terminated successfully.
Valor de la función objetivo: -51.25
Solución:
 [45.    6.25]

Variables de holgura:
 [  0.    543.75   0.     1.25]
```

Y esta solución es la misma que la encontrada en el Ejercicio 2 de la Tarea 10 usando la forma estándar y hallando los puntos básicos factibles.

Finalmente, importamos el modulo `lib_t11` donde implementamos la funciones `positive_cond` y `restriction_cond`, las cuales verifican bajo la tolerancia seleccionada si se cumplen la condición de no negatividad de las variables y las reestricciones de desigualdad, respectivamente.

Resolvemos el numeral 4 y 5 con estas funciones en la siguiente celda.

[2]:
```
import numpy as np
from lib_t11 import *

tol=np.finfo(float).eps**(1/2)

# Condicion de no negatividad
positive_cond(tol,opt_ineq.x)

# Restricciones de desigualdad
restriction_cond(tol,opt_ineq.x,lhs_ineq,rhs_ineq)
```

```
Se cumplen la condicion de no negatividad
Se cumple las restricciones de desigualdad
```

2 Ejercicio 2 (3 puntos)

1. Escriba el problema anterior en su forma estándar.
2. Construya los vectores **b**, **c** y la matriz **A** y resuelva este problema con la librería.
3. Imprima un mensaje que indique si se encontró la solución, y en ese caso imprima la solución, el valor de la función objetivo, las variables de holgura y el error

$$\|\mathbf{Ax} - \mathbf{b}\|.$$

4. Calcule el error E_x como en el Ejercicio 1 y si $E_x < \tau$ imprima un mensaje de que se cumple la condición de no negatividad.

2.1 Solución:

En la Tarea 10 se nos proporciona la forma estándar considerando $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)^T$, aquí las variables del problema son x_1 y x_2, las restantes son las variables de holgura.

En este caso, consideramos $\mathbf{c} = (-1, -1, 0, 0, 0, 0)^T$, $\mathbf{b} = (2400, 2100, 45, 5)$ y

$$\mathbf{A} = \begin{pmatrix} 50 & 24 & 1 & 0 & 0 & 0 \\ 30 & 33 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

Por lo tanto, el problema en forma estándar es

$$\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \quad \text{sujeto a} \quad \mathbf{Ax} = \mathbf{b} \quad \text{y} \quad \mathbf{x} \geq \mathbf{0}$$

En forma estándar, el problema de programación lineal lo resolvemos utilizando los argumentos `A_eq` y `b_eq` de la función `linprog` de la librería `scipy`

[3]:
```python
c = np.array([-1, -1, 0, 0, 0, 0 ])
b = np.array([2400, 2100, 45, 5])
A = np.array([[50, 24, 1, 0, 0, 0],
              [30, 33, 0, 1, 0, 0],
              [ 1,  0, 0, 0,-1, 0],
              [ 0,  1, 0, 0, 0,-1] ])

# Cotas de las variables
bnd = [(0, scipy.inf),  # cotas para x1
       (0, scipy.inf),  # cotas para x2
       (0, scipy.inf),  # cotas para x3
       (0, scipy.inf),  # cotas para x4
       (0, scipy.inf),  # cotas para x5
       (0, scipy.inf)]  # cotas para x6

opt_eq = linprog(c=c, A_eq=A, b_eq=b, bounds=bnd,
              method="simplex")

print('\nResultado del proceso:', opt_eq.message)
if opt_eq.success:
    print('Valor de la función objetivo:', opt_eq.fun)
    print('\nSolución:\n', opt_eq.x)
    print('\n|Ax-b|: ', np.linalg.norm(opt_eq.con))
```

```
Resultado del proceso: Optimization terminated successfully.
Valor de la función objetivo: -51.25

Solución:
 [ 45.     6.25   0.   543.75   0.     1.25]

|Ax-b|:  3.552713678800501e-15
```

Las variables de holgura, como previamente dijimos, son las coordenadas del vector solución correspondiente a las variables x_3, x_4, x_5, x_6.

Finalmente, checamos la condición de no negatividad

[4]:
```
positive_cond(tol,opt_eq.x)
```

```
Se cumplen la condicion de no negatividad
```

3 Ejercicio 3 (4 puntos)

1. Escriba el problema dual del Ejercicio 2.
2. Resuelva el problema dual con la librería. Esto debería devolver el vector λ que son los multiplicadores de Lagrange de la restricciones de igualdad del problema primal.
3. Imprima un mensaje que indique si se encontró la solución, y de ser así, imprima λ, el valor de la función objetivo y las variables de holgura.
4. Usando el valor $\mathbf{x}$ del Ejercicio 2, imprima el error relativo

$$\frac{|\mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \lambda|}{|\mathbf{c}^\top \mathbf{x}|}.$$

4. Defina el vector $\mathbf{s}$ como las variables de holgura.
5. Programe una función que reciba los vectores $\mathbf{b}, \mathbf{c}, \mathbf{x}, \lambda, \mathbf{s}$, la matriz $\mathbf{A}$ y una tolerancia τ, y verique que se cumplen las condiciones KKT:

$$\begin{aligned}
\mathbf{A}^\top \lambda + \mathbf{s} &= \mathbf{c}, && (1)\\
\mathbf{A}\mathbf{x} &= \mathbf{b}, && (2)\\
\mathbf{x} &\geq \mathbf{0}, && (3)\\
\mathbf{s} &\geq \mathbf{0}, && (4)\\
x_i s_i &= 0, \qquad i = 1, 2, ..., n. && (5)
\end{aligned}$$

Calcule los errores E_x y E_s como en el Ejercicio 1, para saber que tanto se violan las restricciones $\mathbf{x} \geq \mathbf{0}$ y $\mathbf{s} \geq \mathbf{0}$.

La función debe imprimir - El error $\|\mathbf{A}^\top \lambda + \mathbf{s} - \mathbf{c}\|$. - El error $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$. - Si $E_x < \tau$, imprima que se cumple las restricciones de no negatividad de $\mathbf{x}$. - Si $E_s < \tau$, imprima que se cumple las restricciones de no negatividad de $\mathbf{s}$. - Calcule el valor de la suma $\sum_i |x_i s_i|$ y si es menor que τ, imprima un mensaje que indique que se cumple la condición de complementariedad.

6. Use la función anterior en el problema para reportar los resultados.

Nota: En el problema dual las variables en λ no tienen restricciones de cota. Si usa, por ejemplo, la función `linprog` para resolver el problema, ponga explícitamente que las cotas de las variables son $-\infty$ e ∞ para que la función no use las cotas que tiene fijas de manera predeterminada.

3.1 Solución:

El problema dual del problema primal en forma estándar descrito en el Ejercicio 2 es

$$\min_{\lambda} -\mathbf{b}^T\lambda \quad \text{sujeto a} \quad \mathbf{A}^T\lambda \leq \mathbf{c},$$

con $\mathbf{A}$, $\mathbf{b}$ y $\mathbf{c}$ como en el Ejercicio 2.

Utilizamos la función `linprog` con los argumentos correspondientes las restricciones de desigualdad del tipo *menor o igual*. La solución al problema dual es la siguiente.

[5]:
```
# Coeficientes de la funcion objetivo
c_dual = -b

# Coeficientes del lado izquierdo de las desigualdades del tipo "menor o igual a"
A_dual = A.T

# Coeficientes del vector del lado derecho de las desigualdades del tipo "menor␣
↪o igual a"
b_dual = c

# Cotas de las variables
bnd = [(-scipy.inf, scipy.inf),  # cotas para lamb1
       (-scipy.inf, scipy.inf),  # cotas para lamb2
       (-scipy.inf, scipy.inf),  # cotas para lamb3
       (-scipy.inf, scipy.inf)]  # cotas para lamb4

opt_dual = linprog(c=c_dual, A_ub=A_dual, b_ub=b_dual, bounds=bnd,
              method="simplex")

print('\nResultado del proceso:', opt_dual.message)
if opt_dual.success:
    print('Valor de la función objetivo:', opt_dual.fun)
    print('Solución:\n', opt_dual.x)
    print('\nVariables de holgura:\n', opt_dual.slack)
```

```
Resultado del proceso: Optimization terminated successfully.
Valor de la función objetivo: 51.24999999999999
Solución:
 [-0.04166667  0.          1.08333333  0.        ]

Variables de holgura:
```

```
[-4.44089210e-16  0.00000000e+00  4.16666667e-02  0.00000000e+00
  1.08333333e+00  0.00000000e+00]
```

Que coincide con la vectores de multiplicadores de Lagrange λ y $\mathbf{s}$, salvo por error numérico, que se obtuvieron en la Tarea 10 a partir de la solución del problema primal y las condiciones KKT.

Por otro lado, por la relación existente entre el problema primal y su dual tienen el mismo valor para la función objetivo (considerando la función objetivo del problema dual correspondiente a una maximización).

A continuación, mostramos el error relativo entre los valores óptimos hayados para el para el problema primal y el dual.

[6]:
```
lamb=opt_dual.x  # Multiplicadores de Lagrange condiciones de igualdad.
s=opt_dual.slack # Variables de holgura

rel_err=np.abs(opt_eq.fun+opt_dual.fun)/np.abs(opt_eq.fun) # Error relativo

print('Error relativo entre las funciones objetivos del problema prima y dual es:
 ↪ ')
print(rel_err)
```

```
Error relativo entre las funciones objetivos del problema prima y dual es:
1.38642485026361e-16
```

Que es un valor pequeño y de hecho es más pequeño que la tolerancia $\tau = \sqrt{\epsilon_m}$, por lo que podríamos afirmar que el valor óptimo de la función objetivo para el problema primal y dual es el mismo.

Finalmente, prograremos la función que verificará las condiciones KKT con la solución del problema obtenida y las variables duales óptimas. Esta función quedará definida en el módulo `lib_t11` y llevará por nombre `KKT_cond`.

Usamos esta última función para obtener los resultados solicitados

[7]:
```
KKT_cond(tol,b,c,opt_eq.x,lamb,s,A)
```

```
Condicion 1: |AT*lamb+s-c| = 0.0
Condicion 2: |Ax-b| =  3.552713678800501e-15
SI se cumple la condicion de no negatividad de x
SI se cumple la condicion de no negatividad de s
SI se cumple la condicion de complentariedad
```

De los resultados anterior concluimos que los valores hallados para el problema primal como el dual son óptimos de sus respectivos contextos.