Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: Fred Torres Cruz
Autor : Luz Magaly Turpo Mamani
Link Github : https://github.com/luz897/ACTIVIDAD-08

Trabajo Encargado - N° 08

Integer Linear Programming

Ejercicio 01

Resuelve el siguiente problema de forma incremental utilizando el método de ramificación y acotamiento de Dakin:

$$\text{Maximizar } P(x_1, x_2, x_3) = 4x_1 + 3x_2 + 3x_3$$

Sujeto a:

$$4x_1 + 2x_2 + x_3 \leq 10 \tag{8.19}$$
$$3x_1 + 4x_2 + 2x_3 \leq 14 \tag{8.20}$$
$$2x_1 + x_2 + 3x_3 \leq 7 \tag{8.21}$$

donde x_1, x_2, x_3 son enteros no negativos.

Dibuja un árbol de decisión con tus respuestas a los subproblemas, como en el Ejemplo 8.2.4. Además, para cada iteración de la ruta particular que sigues, indica claramente el problema de programación lineal (PL) que estás resolviendo. Puedes usar Solver (o el programa de tu preferencia) para resolver cada uno de estos problemas individuales de PL.

Código en Python

```
import streamlit as st
import pulp
import matplotlib.pyplot as plt
import numpy as np

st.title("Ejercicio 8.1 - Metodo de Dakin's Branch and Bound ")

# Definir el problema de maximizacion
prob = pulp.LpProblem("Maximizar_P", pulp.LpMaximize)

# Definir variables
x1 = pulp.LpVariable('x1', lowBound=0, cat='Integer')
x2 = pulp.LpVariable('x2', lowBound=0, cat='Integer')
x3 = pulp.LpVariable('x3', lowBound=0, cat='Integer')
```

```python

# Funcion objetivo
prob += 4 * x1 + 3 * x2 + 3 * x3

# Restricciones
prob += 4 * x1 + 2 * x2 + x3 <= 10
prob += 3 * x1 + 4 * x2 + 2 * x3 <= 14
prob += 2 * x1 + x2 + 3 * x3 <= 7

# Resolver el problema
prob.solve()

# Mostrar la solucion en Streamlit
st.write("Estado:", pulp.LpStatus[prob.status])
st.write("x1 =", x1.varValue)
st.write("x2 =", x2.varValue)
st.write("x3 =", x3.varValue)
st.write("Valor optimo (P) =", pulp.value(prob.objective))

# Grafica de la region factible
x_vals = np.linspace(0, 5, 400)
y_vals1 = (10 - 4 * x_vals) / 2
y_vals2 = (14 - 3 * x_vals) / 4
y_vals3 = (7 - 2 * x_vals) / 1

plt.figure(figsize=(10, 6))
plt.plot(x_vals, y_vals1, label="4x1 + 2x2 + x3 <= 10")
plt.plot(x_vals, y_vals2, label="3x1 + 4x2 + 2x3 <= 14")
plt.plot(x_vals, y_vals3, label="2x1 + x2 + 3x3 <= 7")
plt.xlim((0, 5))
plt.ylim((0, 5))
plt.xlabel('x1')
plt.ylabel('x2')
plt.legend()
plt.title('Region factible')
plt.fill_between(x_vals, np.minimum(np.minimum(y_vals1, y_vals2), y_vals3),
    alpha=0.3)
st.pyplot(plt)
```

Resultados

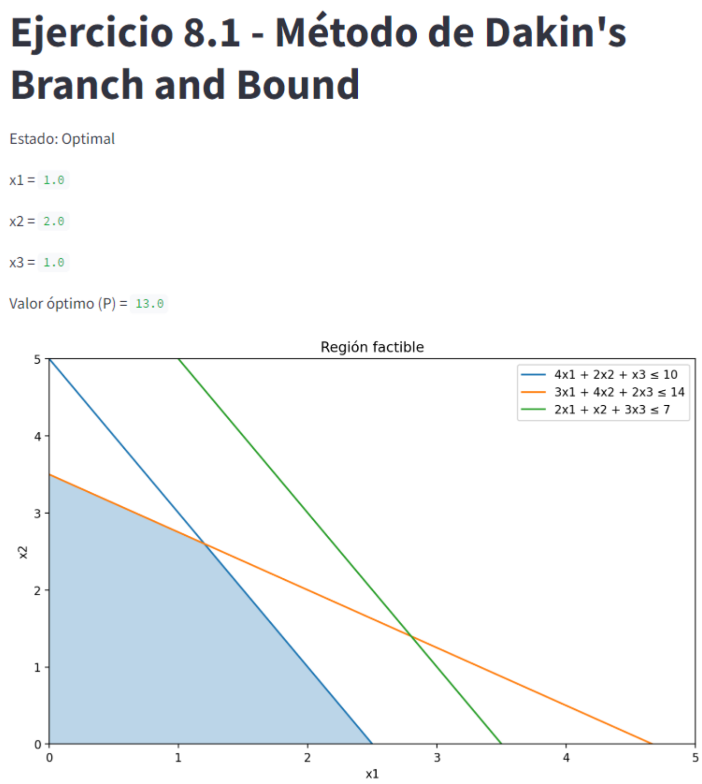


Figura 1: Gráfico

Ejercicio 2

Resuelve el siguiente problema de forma incremental y a mano utilizando planos de corte (Cut-Planes):

$$\text{Minimizar } C(x, y) = x - y$$

Sujeto a:

$$3x + 4y \leq 6 \tag{8.22}$$
$$x - y \leq 1 \tag{8.23}$$

donde x e y son enteros no negativos.

1. Explica el problema de Programación Lineal modificado (es decir, lo que tenemos después de introducir variables de holgura, excedente y artificial, etc.).
2. Grafica la región factible.
3. Proporciona el tableau inicial del método simplex.

4. Comienza a introducir de forma iterativa cortes de Gomory hasta que se obtenga una solución entera, donde para cada iteración:

   a) Muestra claramente el trabajo que justifica la introducción de un determinado plano de corte (es decir, la nueva restricción).

   b) Escribe el nuevo problema de PL (el que se deriva de la iteración anterior más la nueva restricción) y el nuevo tableau inicial del método simplex.

   c) Proporciona un diagrama que muestre la región factible para las variables de decisión.

   d) Usa algún recurso computacional para encontrar el tableau final del método simplex para la PL de esta iteración y proporciona el tableau final (una captura de pantalla es aceptable).

Código en Python

```python
import streamlit as st
import pulp
import matplotlib.pyplot as plt
import numpy as np

st.title("Ejercicio 8.3 - Metodo de Cortes de Gomory")

# Definir el problema de minimizacion
prob = pulp.LpProblem("Minimizar_C", pulp.LpMinimize)

# Definir variables
x = pulp.LpVariable('x', lowBound=0, cat='Integer')
y = pulp.LpVariable('y', lowBound=0, cat='Integer')

# Funcion objetivo
prob += x - y

# Restricciones
prob += 3 * x + 4 * y <= 6
prob += x - y <= 1

# Resolver el problema
prob.solve()

# Mostrar la solucion en Streamlit
st.write("Estado:", pulp.LpStatus[prob.status])
st.write("x =", x.varValue)
st.write("y =", y.varValue)
st.write("Valor optimo (C) =", pulp.value(prob.objective))

# Grafica de la region factible
x_vals = np.linspace(0, 3, 400)
y_vals1 = (6 - 3 * x_vals) / 4
y_vals2 = x_vals - 1

```

```
plt.figure(figsize=(10, 6))
plt.plot(x_vals, y_vals1, label="3x + 4y <= 6")
plt.plot(x_vals, y_vals2, label="x - y <= 1")
plt.xlim((0, 3))
plt.ylim((0, 3))
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.title('Region factible')
plt.fill_between(x_vals, np.minimum(y_vals1, y_vals2), alpha=0.3)
st.pyplot(plt)
```

Resultados

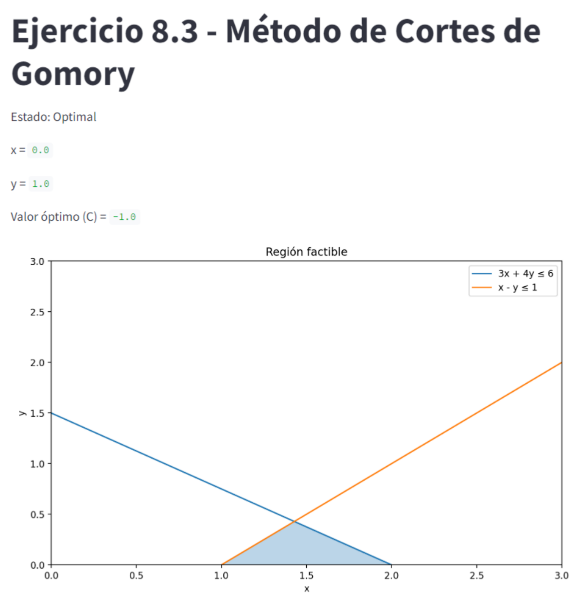
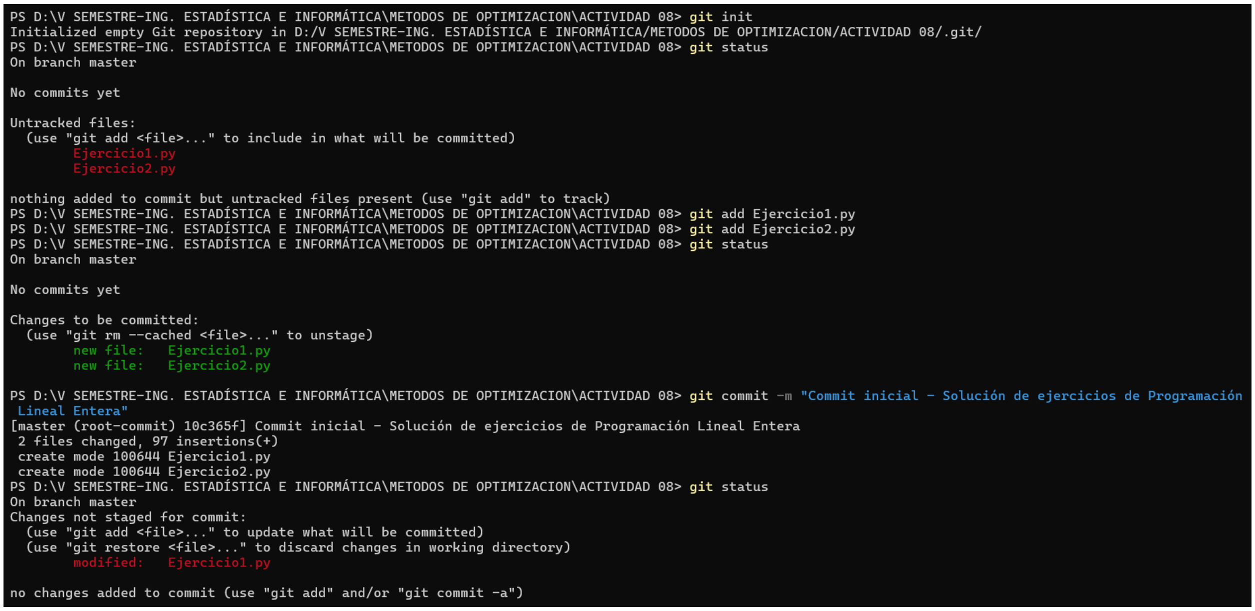


Figura 2: Gráfico

Programa en GitHub



```
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git init
Initialized empty Git repository in D:/V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA/METODOS DE OPTIMIZACION/ACTIVIDAD 08/.git/
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Ejercicio1.py
        Ejercicio2.py

nothing added to commit but untracked files present (use "git add" to track)
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git add Ejercicio1.py
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git add Ejercicio2.py
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Ejercicio1.py
        new file:   Ejercicio2.py

PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git commit -m "Commit inicial - Solución de ejercicios de Programación
 Lineal Entera"
[master (root-commit) 10c365f] Commit inicial - Solución de ejercicios de Programación Lineal Entera
 2 files changed, 97 insertions(+)
 create mode 100644 Ejercicio1.py
 create mode 100644 Ejercicio2.py
PS D:\V SEMESTRE-ING. ESTADÍSTICA E INFORMÁTICA\METODOS DE OPTIMIZACION\ACTIVIDAD 08> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Ejercicio1.py

no changes added to commit (use "git add" and/or "git commit -a")
```

Figura 3: GitHub