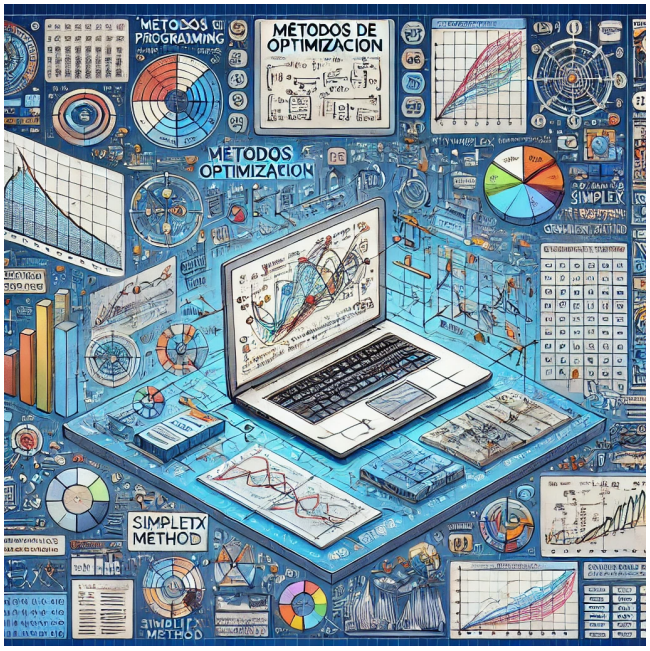
UNIVERSIDAD NACIONAL DEL ALTIPLANO

INGENIERÍA ESTADÍSTICA E INFORMÁTICA

MÉTODOS DE OPTIMIZACIÓN

Actividad N° 5

Desarrolle el ejercicio designado por el docent, genere el codigo acoplado a una interfaz grafica, adicionalmente un video usando manim

[h] 

Andree Alessandro Chili Lima
[229071]
https://github.com/antartida15l
https://youtu.be/8T-CvQt-jZ4?si=RCF1Q6bu_zmAxFX0

11/10/2024

PROBLEMA 16

Una empresa está optimizando su infraestructura distribuida para ejecutar tareas de análisis de datos. La infraestructura está compuesta por dos tipos de máquinas: Tipo A y Tipo B.

- Las máquinas de Tipo A procesan 200 GB de datos por hora y tienen un costo operativo de 50 por hora.
- Las máquinas de Tipo B procesan 300 GB de datos por hora y tienen un costo operativo de 70 por hora.
- La empresa tiene un presupuesto operativo diario de $15,000$ y necesita procesar al menos 25,000 GB de datos al día.
- Cada máquina Tipo A solo puede operar durante un máximo de 10 horas diarias, mientras que las de Tipo B pueden operar hasta 8 horas diarias.

Formule un modelo de programación lineal que minimice los costos de operación de la infraestructura, cumpliendo con las restricciones de procesamiento de datos y presupuesto.

Objetivo

Minimizar el costo total operativo diario de las máquinas de Tipo A y Tipo B:

$$\text{Minimizar } Z = 50x_A + 70x_B$$

Restricciones

1. $200x_A + 300x_B \geq 25,000$ (Capacidad mínima de procesamiento de datos)
2. $50x_A + 70x_B \leq 15,000$ (Presupuesto operativo diario)
3. $x_A \leq 10$ (Límite de operación de máquinas Tipo A)
4. $x_B \leq 8$ (Límite de operación de máquinas Tipo B)
5. $x_A \geq 0, \quad x_B \geq 0$ (Restricciones de no negatividad)

Modelo completo

$$\text{Minimizar } Z = 50x_A + 70x_B$$

Sujeto a:

$$\begin{aligned} 200x_A + 300x_B &\geq 25,000 \\ 50x_A + 70x_B &\leq 15,000 \\ x_A &\leq 10 \\ x_B &\leq 8 \\ x_A, x_B &\geq 0 \end{aligned}$$

```python
import streamlit as st
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import linprog
st.set_page_config(page_title="Optimización de Infraestructura Distribuida", layout="centered")

def optimizar_costo():
    c = [50, 70]
    A_ub = [
        [-200, -300],
        [50, 70]
    ]
    b_ub = [-25000, 15000]
    x_bounds = (0, 10)
    y_bounds = (0, 8)
    res = linprog(c, A_ub=A_ub, b_ub=b_ub, bounds=[x_bounds, y_bounds], method='highs')

    if res.success:
        x_A, x_B = res.x
        costo_total = res.fun
        return {
            "x_A": x_A,
            "x_B": x_B,
            "costo_total": costo_total,
            "mensaje": "Optimización exitosa."
        }
    else:
        return {
            "mensaje": "No se pudo encontrar una solución óptima. La solución es inviable."
        }

st.title("Optimización de Infraestructura Distribuida")
st.write(""" FINESI MAS NA""")
```

```python
st.title("Optimización de Infraestructura Distribuida")
st.write(""" FINESI MAS NA""")
if st.button("Resolver metodo de Optimización"):
    resultado = optimizar_costo()
    if "x_A" in resultado:
        st.success(f"**Solución Óptima:**")
        st.write(f"- Horas para Máquinas Tipo A: **{resultado['x_A']:.2f} horas**")
        st.write(f"- Horas para Máquinas Tipo B: **{resultado['x_B']:.2f} horas**")
        st.info(f"**Costo Operativo Total:** ${resultado['costo_total']:.2f} por día")
    else:
        st.error(resultado["mensaje"])

st.markdown("---")
st.subheader("")
st.write("Ajusta las horas de operación de las máquinas para ver cómo se comporta el modelo y verifica si se cumplen las restricciones.")
col1, col2 = st.columns(2)
with col1:
    x_A_input = st.slider("Horas de operación para Máquinas Tipo A", min_value=0.0, max_value=10.0, value=5.0, step=0.1)
with col2:
    x_B_input = st.slider("Horas de operación para Máquinas Tipo B", min_value=0.0, max_value=8.0, value=5.0, step=0.1)
procesamiento_total = 200 * x_A_input + 300 * x_B_input
costo_total_input = 50 * x_A_input + 70 * x_B_input
st.write(f"**Procesamiento Total:** {procesamiento_total:.2f} GB")
st.write(f"**Costo Total:** ${costo_total_input:.2f}")
if procesamiento_total < 25000:
    st.warning("⚠️ El procesamiento es insuficiente. Se deben procesar al menos **25,000 GB**.")
else:
    st.success("✅ El procesamiento cumple con el requisito mínimo.")

if costo_total_input > 15000:
    st.warning("⚠️ El costo supera el presupuesto operativo de **$15,000**.")
```

PROBLEMA16.py > optimizar_costo

```python
if costo_total_input > 15000:
    st.warning("⚠️ El costo supera el presupuesto operativo de **$15,000**.")
else:
    st.success("✅ El costo está dentro del presupuesto operativo.")

if x_A_input > 10:
    st.error("❌ Las máquinas de Tipo A no pueden operar más de **10 horas al día**.")
if x_B_input > 8:
    st.error("❌ Las máquinas de Tipo B no pueden operar más de **8 horas al día**.")

st.markdown("---")

st.subheader("Visualización de Restricciones y Soluciones")
fig, ax = plt.subplots(figsize=(10, 6))
x_vals = np.linspace(0, 10, 400)
y_procesamiento = (25000 - 200 * x_vals) / 300
y_costo = (15000 - 50 * x_vals) / 70
y_max_A = 8
y_max_B = 8
ax.plot(x_vals, y_procesamiento, label='Procesamiento mínimo (≥ 25,000 GB)', color='blue')
ax.plot(x_vals, y_costo, label='Presupuesto máximo ($15,000)', color='red')
ax.axhline(8, label='Máximo de 8 horas para Tipo B', color='green', linestyle='--')
ax.axvline(10, label='Máximo de 10 horas para Tipo A', color='orange', linestyle='--')
y_min = np.maximum(y_procesamiento, 0)
y_max = np.minimum(y_costo, 8)
ax.fill_between(x_vals, y_min, y_max, where=(y_max >= y_min), color='yellow', alpha=0.3, label='Región Factible')
resultado_optimo = optimizar_costo()
if "x_A" in resultado_optimo:
    ax.plot(resultado_optimo["x_A"], resultado_optimo["x_B"], 'ko', label='Solución Óptima')
ax.plot(x_A_input, x_B_input, 'mo', label='Punto Seleccionado')
ax.set_xlim(0, 11)
ax.set_ylim(0, 9)
```

PROBLEMA16.py > optimizar_costo

```python
ax.set_xlabel("Horas de Máquinas Tipo A")
ax.set_ylabel("Horas de Máquinas Tipo B")
ax.set_title("Visualización de Restricciones y Soluciones")
ax.legend()
st.pyplot(fig)
st.subheader("Resumen de Restricciones")

restricciones = {
    "Restricción": [
        "Procesamiento mínimo (GB)",
        "Presupuesto máximo ($)",
        "Máximo horas Tipo A (horas)",
        "Máximo horas Tipo B (horas)"
    ],
    "Condición": [
        "200x_A + 300x_B ≥ 25,000",
        "50x_A + 70x_B ≤ 15,000",
        "x_A ≤ 10",
        "x_B ≤ 8"
    ],
    "Estado Actual": [
        f"{procesamiento_total:.2f} ≥ 25,000 GB" if procesamiento_total >= 25000 else f"{procesamiento_total:.2f} < 25,000 GB",
        f"${costo_total_input:.2f} ≤ $15,000" if costo_total_input <= 15000 else f"${costo_total_input:.2f} > $15,000",
        f"{x_A_input} ≤ 10" if x_A_input <=10 else f"{x_A_input} > 10",
        f"{x_B_input} ≤ 8" if x_B_input <=8 else f"{x_B_input} > 8"
    ]
}

import pandas as pd
df_restricciones = pd.DataFrame(restricciones)
st.table(df_restricciones)
st.markdown("---")

st.subheader("Conclusión")
if "x_A" in resultado_optimo:
    st.success("La optimización fue exitosa. Se puede proceder con los valores obtenidos.")
else:
    st.error("No se pudo encontrar una solución óptima. La solución es inviable.")

```

Optimizacion de Infraestructura Distribuida

FINESI MAS NA

Resolver metodo de Optimización

No se pudo encontrar una solución óptima. La solución es inviable.

Ajusta las horas de operación de las máquinas para ver cómo se comporta el modelo y verifica si se cumplen las restricciones.

Horas de operación para Máquinas Tipo A
5.00
0.00 10.00

Horas de operación para Máquinas Tipo B
5.00
0.00 8.00

Procesamiento Total: 2500.00 GB

Costo Total: $600.00

⚠ El procesamiento es insuficiente. Se deben procesar al menos **25,000 GB**.

✅ El costo está dentro del presupuesto operativo.

Visualización de Restricciones y Soluciones