LABORATORIO DE OPTIMIZACIÓN

UNIDAD ACADÉMICA:	INGENIERÍA DE SISTEMAS
CURSO:	OPTIMIZACIÓN
PRÁCTICA Nº 03:	Método Simplex

1. OBJETIVO

- *Diseñar y Desarrollar un programa de computadora en Python, que resuelva modelos matemáticos de Programación Lineal, utilizando el Método Simplex, visualizando de forma didáctica los resultados de las iteraciones de la solución.*

2. CONSULTA PREVIA

Conceptos básicos de Programación Lineal, métodos de optimización, Método Simplex para resolver problemas de Programación Lineal, Herramientas informáticas para la programación en Python.

3. FUNDAMENTO TEÓRICO

La Programación Lineal es una técnica matemática utilizada para la optimización, es decir, la maximización o minimización de una función objetivo lineal sujeta a un conjunto de restricciones también lineales. Esta metodología es ampliamente aplicada en la investigación operativa y la economía para resolver problemas en los que los recursos son limitados y deben ser asignados de manera eficiente.

Componentes de un Modelo de Programación Lineal:

- ***Variables de decisión:*** *Representan las cantidades que el modelo intenta determinar. Estas variables son los "factores controlables" del problema, como la cantidad de productos a fabricar, la cantidad de horas de trabajo a asignar, etc.*
- ***Función objetivo:*** *Es la función lineal que se desea maximizar o minimizar. Por ejemplo, maximizar las ganancias o minimizar los costos. Se expresa en función de las variables de decisión.*
- ***Restricciones:*** *Son las limitaciones o condiciones que deben cumplir las variables de decisión. Estas restricciones suelen representar recursos limitados como tiempo, dinero, materiales, etc. Se expresan como ecuaciones o inecuaciones lineales.*
- ***Región factible:*** *Es el conjunto de todas las posibles soluciones que satisfacen las restricciones. Cuando el problema tiene más de dos variables, la región factible se representa en un espacio de mayor dimensión. En el caso de tres variables, la región factible es un poliedro en un espacio tridimensional. Para más de tres variables, la región factible existe en un espacio n-dimensional y no puede visualizarse fácilmente.*

Método Simplex en Programación Lineal: *Es un algoritmo iterativo utilizado para resolver problemas de programación lineal con múltiples variables de decisión. Se basa en el desplazamiento entre los vértices de la región factible en un espacio multidimensional hasta encontrar la solución óptima. Es eficiente para problemas de gran escala en los que no puede utilizarse el método gráfico debido al número limitado de variables que se pueden representar en un plano cartesiano bidimensional.*

Pasos del Método Simplex:

- ***Paso 1: Formular el problema en una forma estándar:***
  Se expresan todas las restricciones como ecuaciones de igualdad agregando variables de holgura, exceso o artificiales según sea necesario. También se debe representar la función objetivo en forma de maximización.
- ***Paso 2: Construir la tabla inicial (tablero simplex):***
  Se organiza la información en una tabla simplex, que incluye los coeficientes de las restricciones y la función objetivo en un orden especifico (posteriormente se ilustra con un ejemplo).
- ***Paso 3: Identificar la variable entrante y la variable saliente:***
  Se elige la variable con el coeficiente más negativo en la fila de la función objetivo (para maximización) o más positivo (para minimización), indicando la dirección de mejora (criterio de optimalidad del problema). Luego de esto se determina la variable de salida que dejará la base dividiendo los valores de la ultima columna de la tabla simplex entre los coeficientes positivos de la variable entrante y seleccionando el menor cociente (criterio de razón mínima).
- ***Paso 4: Identificar el elemento pivote:***
  Se selecciona el elemento situado entre la intersección de la fila de la variable saliente y la columna de la variable entrante. Este valor se utiliza para normalizar la tabla y actualizar la base.
- ***Paso 5: Actualizar la tabla simplex (pivoteo):***
  Se normaliza la fila pivote (haciendo que el elemento pivote sea 1) y se realizan operaciones matriciales entre filas para convertir los demás valores de la columna a cero, garantizando que la nueva variable de la base tenga una representación correcta.
- ***Paso 6: Repetir el proceso:***
  Se repiten los pasos 3 a 5 hasta que ya no haya más coeficientes negativos en la fila de la función objetivo (para maximización), indicando que se ha alcanzado la solución óptima.
- ***Paso 7: Interpretar los resultados:***
  Se extraen los valores óptimos de las variables de decisión a partir de la tabla simplex, y se interpreta el significado de las variables de holgura o artificiales según el contexto del problema.

LABORATORIO DE OPTIMIZACIÓN

Resolución paso a paso de un problema de PL utilizando el Método Simplex:

Una empresa produce cuatro tipos de productos, x1, x2, x3 y x4. Cada producto requiere diferentes cantidades de recursos limitados. La empresa desea determinar cuántas unidades de cada producto debe producir para maximizar sus ganancias, respetando las restricciones de disponibilidad de recursos.

La ganancia total para maximizar está dada por:

$$maximizar\ z = 2x_1 - 4x_2 + 5x_3 - 6x_4$$

Sujeto a las restricciones de recursos:

$$x_1 + 4x_2 - 2x_3 + 8x_4 \leq 2$$

$$-x_1 + 2x_2 + 3x_3 + 4x_4 \leq 1$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Encuentre la cantidad óptima de cada producto para maximizar la ganancia.

Procedimiento:

1. Formular el problema en una forma estándar

Dado que las restricciones son de la forma ≤, se agregan las variables de holgura s_1 *y* s_2 *para convertir las inecuaciones en ecuaciones:*

$$x_1 + 4x_2 - 2x_3 + 8x_4 + s_1 = 2$$

$$-x_1 + 2x_2 + 3x_3 + 4x_4 + s_2 = 1$$

Ahora, la función objetivo en su forma estándar es:

$$z - 2x_1 + 4x_2 - 5x_3 + 6x_4 = 0$$

2. Construir la tabla inicial (tablero simplex):

VB	Z	x1	x2	x3	x4	S1	S2	CR
S1	0	1	4	-2	8	1	0	2
S2	0	-1	2	3	4	0	1	1
Z	1	-2	4	-5	6	0	0	0

Se construye la tabla simplex con cada una de las ecuaciones definidas en el paso 1, la columna ***CR*** *hace referencia al coeficiente de restricción de cada ecuación. La columna* ***VB*** *hace referencia a las variables base establecidas para el problema, en esta columna se establecen las variables de holgura.*

LABORATORIO DE OPTIMIZACIÓN

3. Identificar la variable entrante y la variable saliente:

VB	Z	x1	x2	x3	x4	S1	S2	CR
S1	0	1	4	-2	8	1	0	2
S2	0	-1	2	3 (EP)	4	0	1	1
Z	1	-2	4	-5	6	0	0	0

Para identificar la variable entrante se identifica el coeficiente más negativo en la fila de z, en este caso x_3 tiene el coeficiente más negativo (-5), por lo que entra a la base. Posteriormente se calculan las razones CR/x_3 para obtener la variable saliente.

$$\frac{2}{-2}\ (No\ se\ considera\ porque\ es\ negativa) \qquad \frac{1}{3} = 0.33$$

Por tanto, la variable s_2 sale de la base.

4. Identificar el elemento pivote:

El elemento pivote es 3 (EP), ubicado en la intersección de la fila s_2 y la columna x_3.

5. Normalizar la fila pivote y actualizar la tabla

En este paso de método simplex, se procede a la normalización de la fila pivote dividiéndola entre el valor del elemento pivote identificado inicialmente (3). Esto permite que el elemento pivote se convierta en 1, facilitando las siguientes operaciones. Posteriormente, se aplican operaciones fila para hacer ceros en el resto de la columna x_3, esto se logra restando múltiplos adecuados de la fila pivote a las demás filas de la tabla.

Una vez realizado el proceso mencionado anteriormente, se obtiene una nueva tabla simplex con variables entrante, saliente y elemento pivote actualizados.

VB	Z	x1	x2	x3	x4	S1	S2	CR
S1	0	1/3 (EP)	16/3	0	32/3	1	2/3	8/3
x3	0	-1/3	2/3	1	4/3	0	1/3	1/3
Z	1	-11/3	22/3	0	38/3	0	5/3	5/3

El proceso se repite iterativamente hasta que se cumple el criterio de optimalidad, es decir, hasta que no quedan coeficientes negativos en la fila de la función objetivo (en caso de maximización). En este punto, se ha alcanzado la solución óptima, lo que indica que se ha encontrado la mejor combinación de valores de decisión que maximiza la función objetivo dentro de las restricciones del problema.

VB	Z	x1	x2	x3	x4	S1	S2	CR
x1	0	1	16	0	32	3	2	8
x3	0	0	6	1	12	1	1	3
Z	1	0	66	0	130	11	9	31

Finalmente, se obtiene una tabla simplex sin elementos negativos en la fila z, donde después de resolver el modelo de PL la solución óptima es:

$$x_1 = 8, x_3 = 3, x_2 = 0, x_4 = 0\,; \quad z = 8$$

*Esto significa que, para obtener la máxima ganancia, la empresa debe producir **8 unidades del producto $\boldsymbol{x_1}$** y **3 unidades del producto $\boldsymbol{x_3}$.** En cambio, no es rentable producir los productos x_2 y x_4, por lo que su cantidad óptima es 0. El valor óptimo de la función objetivo es $z_{max} = 31$, esto indica que, si la empresa sigue esta estrategia de producción, obtendrá una ganancia máxima de 31 unidades monetarias bajo las restricciones de recursos establecidas.*

Herramientas Matemáticas para la Implementación en Python:

Numpy: *Es una biblioteca fundamental en Python para realizar cálculos numéricos. Proporciona soporte para grandes matrices y vectores multidimensionales, así como una colección de funciones matemáticas para operar con estos datos de manera eficiente. En el contexto de programación lineal, numpy facilita la manipulación de matrices que representan las restricciones y la función objetivo.*

Matplotlib: *Útil para visualizar gráficamente restricciones y regiones factibles en problemas con dos variables.*

Scipy: *Es una biblioteca que extiende las capacidades de numpy y proporciona herramientas y algoritmos de alto nivel para la optimización, integración, interpolación, entre otros. En programación lineal, scipy.optimize.linprog puede resolver problemas de PL mediante métodos más avanzados como el Simplex, cuando el problema no se puede resolver gráficamente debido a la presencia de más de dos variables.*

Pandas: *Organiza y muestra las tablas iterativas del Método Simplex de manera estructurada y comprensible.*

SymPy: *Ayuda con la manipulación simbólica de ecuaciones para verificar soluciones y realizar operaciones algebraicas.*

Tkinter / PyQt / Streamlit: *Facilitan la creación de una interfaz gráfica para mostrar dinámicamente las iteraciones del Método Simplex.*

Google Colab:

También llamado Google Colaboratory, es un entorno de desarrollo interactivo que permite a los usuarios escribir y ejecutar código Python en un navegador. Es especialmente popular en la comunidad científica y de aprendizaje automático debido a sus potentes características, facilidad de uso, y la capacidad de compartir proyectos.

Características principales:

Entorno basado en la nube*:*

Google Colab se ejecuta en la nube, lo que significa que no necesitas instalar nada en tu computadora. Puedes acceder y trabajar en tus cuadernos desde cualquier lugar con conexión a Internet.

Soporte para Python y bibliotecas científicas*:*

Colab es compatible con Python y viene preinstalado con muchas bibliotecas populares como NumPy, Pandas, Matplotlib, TensorFlow, y muchas otras, lo que facilita el desarrollo de proyectos de análisis de datos, ciencia de datos, y aprendizaje automático.

GPU y TPU gratuitas*:*

Google Colab permite utilizar unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPU) de forma gratuita, lo que acelera considerablemente las tareas computacionalmente intensivas, como el entrenamiento de modelos de machine learning.

Interactividad*:*

Puedes escribir y ejecutar bloques de código de forma interactiva, lo que es útil para la experimentación y visualización de datos. Además, puedes incluir texto, imágenes, y enlaces, lo que hace que los cuadernos sean ideales para la documentación y la enseñanza.

Colaboración en tiempo real*:*

Al igual que Google Docs, Google Colab permite la colaboración en tiempo real. Puedes compartir tu cuaderno con otros usuarios, quienes pueden ver y editar el código simultáneamente.

Integración con Google Drive*:*

Colab se integra fácilmente con Google Drive, lo que permite guardar y gestionar tus cuadernos en tu cuenta de Drive. Además, puedes importar y exportar archivos entre tu Drive y Colab.

Facilidad de uso*:*

Es muy accesible para principiantes y expertos por igual. Su interfaz es simple e intuitiva, lo que permite a los usuarios concentrarse en el desarrollo del código sin preocuparse por la configuración del entorno.

LABORATORIO DE OPTIMIZACIÓN

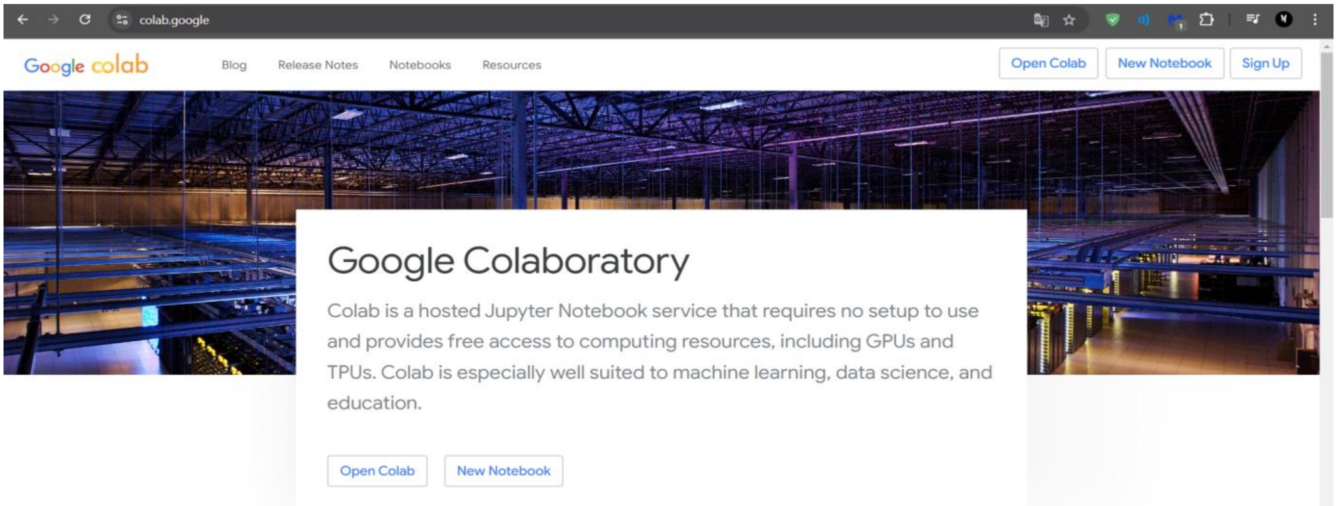

Figura 1. *Google colab*

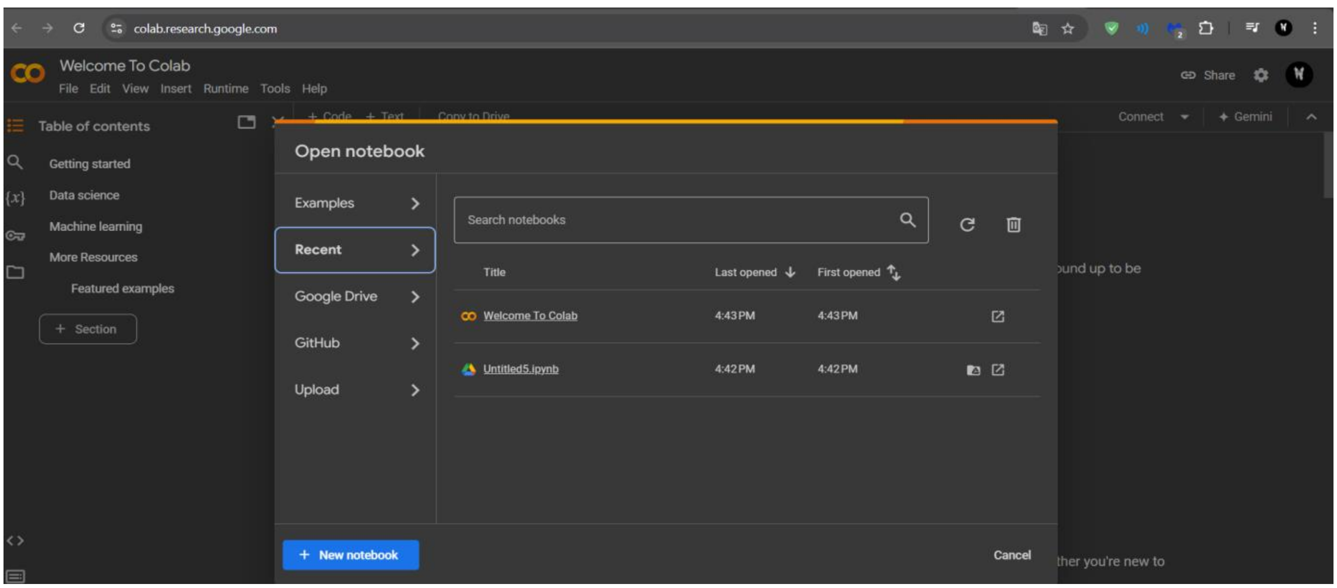

Figura 2. *Nuevo libro en Google colab*

Figura 3. *Layers de código*

Figura 4. *Ejecución de código*

4. EQUIPOS, MATERIALES Y REACTIVOS

Equipos	Materiales	Sustancias y/o Reactivos
Computador	Software Visual Studio code, Python3, Google Colab, bibliotecas (pandas, matplotlib, numpy)	

5. PROCEDIMIENTO O METODOLOGÍA

En esta práctica de laboratorio, los estudiantes diseñarán y desarrollarán un programa en Python que permita construir y resolver un modelo matemático de programación lineal utilizando el método simplex. Se trabajará en grupos de dos estudiantes, empleando Visual Studio Code para el desarrollo del código y Google Colab para la ejecución y visualización de resultados.

Paso 1. Configuración del Entorno de Desarrollo

- *Configurar Visual Studio Code con las extensiones necesarias para Python.*
- *Crear un nuevo proyecto en Visual Studio Code y organizar las carpetas para el código y los gráficos.*

Paso 2. Desarrollo del Modelo Matemático

- *Identificar y formular las funciones objetivo y las restricciones del problema de optimización para posteriormente expresarlas en la forma estándar del método simplex adicionando las variables de holgura necesarias para convertir las inecuaciones en ecuaciones.*

- *Implementar las ecuaciones matemáticas y restricciones en el código Python, asegurando que el programa pueda manejar* ***modelos matemáticos de al menos 2 restricciones*** *y* ***al menos 2 variables.***

Paso 3. Implementación del Método Simplex

El programa debe implementar el método simplex y visualizar en la interfaz gráfica el proceso iterativo de la resolución del problema. Esto implica:

- ***Mostrar las tablas del método simplex en cada iteración,*** *resaltando claramente:*

- *La **columna pivote**, que indica la variable entrante.*
- *La **fila pivote**, que indica la variable saliente.*
- *El **elemento pivote**, que se usa para normalizar la tabla.*

- ***Presentar el resultado final,*** *indicando los valores óptimos de las variables de decisión y el valor de la función objetivo para facilitar la interpretación de los datos obtenidos.*

Paso 4. Ejecución y Pruebas en Google Colab

- *Subir el código a Google Colab para ejecutar y visualizar las tablas utilizadas en el método.*

- *Probar el programa con diferentes conjuntos de datos para validar la correcta implementación del método simplex, asegurándose de que cumpla con las especificaciones de manejo de restricciones, visualización de las tablas y determinación de la solución óptima.*

Paso 5. Documentación del Proceso

Elaborar un informe técnico siguiendo el formato Working Paper de IEEE, que documente todo el proceso de desarrollo, resultados y conclusiones. El informe debe contener los siguientes elementos mínimos:

- *Título de la práctica*
- *Autores*
- *Introducción*
- *Referente teórico*
- *Procedimiento*
- *Código fuente*
- *Resultados*
- *Conclusiones*
- *Bibliografía*

6. RESULTADOS

Como evidencia del desarrollo de esta práctica. Subir un archivo comprimido al aula virtual antes de la fecha límite establecida que contenga el código fuente del programa y un informe técnico del proceso realizado este archivo será la evidencia del desarrollo de la práctica y debe reflejar tanto la implementación técnica como la comprensión teórica del método simplex en la programación lineal.

7. BIBLIOGRAFÍA

*[1]. Murty, K. G. (1983). **Linear Programming**. John Wiley & Sons.*
*[2]. Chvatal, V. (1983). **Linear Programming**. W. H. Freeman and Company.*
*[3].Papadimitriou, C. H., & Steiglitz, K. (1998). **Combinatorial Optimization: Algorithms and Complexity**. Dover Publications.*
*[4].Matplotlib Documentation. (2024). **Matplotlib: Visualization with Python**. Recuperado de https://matplotlib.org/*
*[5]. NumPy Documentation. (2024). **NumPy: The fundamental package for scientific computing with Python**. Recuperado de https://numpy.org/*
*[6].Google Colab. (2024). **Google Collaboratory**. Recuperado de https://colab.research.google.com/*