

Desarrollo de un Software para la Resolución de Problemas de Programación Lineal mediante el Método Simplex

Juan Carlos Barrera Guevara, Diego Alejandro Machado Tovar
Facultad de Ciencias Básicas e Ingeniería, Universidad de los Llanos
 Villavicencio, Colombia
jc.bguevara@unillanos.edu.co
damachado@unillanos.edu.co

I. INTRODUCCIÓN

La investigación de operaciones constituye una disciplina matemática orientada al análisis y la optimización de sistemas complejos mediante el uso de modelos formales y técnicas cuantitativas. Dentro de este campo, la programación lineal se ha consolidado como una de las herramientas más influyentes, al permitir la representación de problemas en los que es necesario maximizar o minimizar una función objetivo lineal sujeta a un conjunto de restricciones. Aunque existen métodos algorítmicos alternativos y de mayor alcance, el método simplex conserva un papel fundamental en el ámbito académico y práctico, debido a su simplicidad conceptual, su carácter iterativo y su capacidad para ilustrar los principios básicos de la programación lineal.

La importancia del método simplex radica en su valor didáctico, pues a través de la construcción de tableaux permite representar la evolución de la solución, facilitando la comprensión de conceptos como factibilidad, región factible, solución básica factible y optimalidad. Asimismo, constituye una base sólida para la introducción de técnicas más avanzadas como el método de dos fases, el método del gran M y algoritmos de optimización entera y no lineal.

En este contexto, se desarrolló un software orientado a la resolución de problemas de programación lineal mediante el método simplex. El objetivo de este prototipo fue integrar los fundamentos teóricos de la optimización lineal con una implementación computacional en Python que no solo permita resolver problemas sino también visualizar las iteraciones intermedias del algoritmo, lo que aporta tanto al ámbito docente como al práctico. El sistema se distingue por ofrecer dos modos de operación: uno manual, donde el usuario introduce directamente los coeficientes de la función objetivo y las restricciones, y otro basado en inteligencia artificial, en el cual el problema puede

plantearse en lenguaje natural para ser automáticamente transformado en un modelo matemático compatible con el método simplex.

I. REFERENTE TEÓRICO

A. Programación Lineal

La programación lineal es una técnica matemática empleada para optimizar la utilización de recursos limitados en un sistema. Su propósito consiste en determinar los valores de un conjunto de variables de decisión que maximizan o minimizan una función objetivo lineal, bajo un conjunto de restricciones también lineales. Estos modelos se utilizan ampliamente en problemas de asignación de recursos, planeación de la producción, transporte y gestión operativa, entre otros.

De manera formal, un modelo de programación lineal puede representarse como:

$$\text{Maximizar o minimizar } Z = \sum_{j=1}^n c_j x_j$$

Sujeto a

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 0, 1, 2, \dots, m$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 0, 1, 2, \dots, m$$

$$x_j \geq 0, \quad j = 0, 1, 2, \dots, n$$

En este esquema, x_j corresponde a las variables de decisión, c_j a los coeficientes de la función objetivo, a_{ij} a los coeficientes de las restricciones y b_i la disponibilidad de recursos.

B. Metodo Simplex

El método simplex, propuesto por Dantzig en 1947, es un algoritmo iterativo diseñado para resolver este tipo de problemas. Su funcionamiento se basa en recorrer los vértices de la región factible definida por las restricciones, partiendo de una solución básica inicial y avanzando hacia vértices adyacentes que mejoran el valor de la función objetivo, hasta alcanzar la solución óptima. Cada iteración se organiza en un tableau, que constituye una

representación matricial donde se almacenan tanto los coeficientes del sistema como los valores de las variables básicas y no básicas. El proceso consiste en identificar la variable entrante (columna pivote) a partir del coeficiente más negativo en la fila objetivo y la variable saliente (fila pivote) mediante la prueba de razón mínima. A continuación, se realiza la operación de pivoteo que actualiza el tableau y define una nueva base. Este procedimiento se repite hasta que no existan más coeficientes negativos en la fila objetivo, condición que garantiza la optimalidad de la solución.

El método simplex, pese a ser robusto, puede enfrentar situaciones especiales. Entre ellas se encuentran la degeneración, cuando múltiples soluciones básicas corresponden al mismo valor de la función objetivo, y los problemas no acotados, en los que el algoritmo detecta que la función objetivo puede crecer indefinidamente sin alcanzar un valor óptimo finito. Asimismo, existe la posibilidad de problemas inviables, en los que el conjunto de restricciones es inconsistente y no existe ninguna solución factible. Estas situaciones constituyen un reto adicional en la implementación computacional, pero su detección refuerza la utilidad del algoritmo tanto en el aprendizaje como en la práctica profesional.

II. PROCEDIMIENTO

El software se diseñó bajo una arquitectura modular que divide claramente el núcleo algorítmico, los componentes de visualización, la interfaz gráfica y el módulo de interpretación de lenguaje natural. El archivo **simplex.py** contiene la implementación principal del algoritmo, incluyendo la construcción del tableau inicial, la selección de columnas y filas pivote, las operaciones de pivoteo y la verificación de condiciones de parada. Este módulo fue diseñado para recibir como parámetros el vector de coeficientes de la función objetivo, la matriz de restricciones y el vector de recursos, retornando tanto las iteraciones como la solución final.

El archivo **display.py** se encarga de la visualización de los tableaux. Para ello, se emplea la librería **pandas**, que permite representar de manera clara y tabular las iteraciones, rotulando adecuadamente las columnas correspondientes a las variables de decisión, las variables de holgura y los valores del lado derecho de las restricciones. Este componente facilita la interpretación pedagógica del algoritmo y su aplicación práctica.

La interfaz gráfica, implementada en el archivo **app.py** mediante la librería **Tkinter**, constituye el punto de interacción con el usuario. Esta interfaz incluye campos de entrada para la captura de coeficientes y restricciones en el modo manual, así como un área de texto para el ingreso de problemas en lenguaje natural en el modo IA. La interfaz

dispone de botones que permiten ejecutar el algoritmo en cualquiera de los modos y un cuadro de salida que muestra tanto las iteraciones intermedias como la solución óptima final.

Finalmente, el módulo **parser_ai.py** integra un sistema de procesamiento de lenguaje natural. En su versión completa, este módulo se conecta a un modelo de lenguaje externo (Groq/LLaMA 3.1), capaz de interpretar la formulación textual de un problema y convertirla en una estructura matemática compatible con el algoritmo simplex. En caso de no disponer de conexión a internet o credenciales de API, el sistema recurre a un parser alternativo basado en expresiones regulares, que reconoce patrones básicos en el texto y construye el modelo correspondiente.

El flujo de trabajo general comienza con la captura del problema en la interfaz gráfica, ya sea en modo manual o IA. Posteriormente, el sistema construye las estructuras de datos correspondientes, inicializa el tableau, ejecuta iteraciones de pivoteo y genera las tablas intermedias hasta alcanzar la condición de optimalidad. Finalmente, el sistema muestra los valores óptimos de las variables de decisión, el valor máximo de la función objetivo y, en caso de presentarse, mensajes diagnósticos que indican la inviabilidad o no acotación del problema.

III. RESULTADOS

Con el fin de validar el correcto funcionamiento del software, se realizaron pruebas sobre distintos problemas de programación lineal aplicando el método simplex. Los resultados obtenidos incluyen la visualización de las iteraciones intermedias en forma de tableaux, así como la identificación de la solución óptima de la función objetivo. La salida generada por el sistema permitió comprobar que el comportamiento del programa es consistente con la teoría de programación lineal y confirma su utilidad como herramienta de apoyo tanto en la enseñanza como en la resolución práctica de problemas de optimización.

Iteración 2

VB entra	VB sale	Elemento pivote					
x1	s2	fila 2, columna x1					

Tabla antes del pivote

	VB	Z	x1	x2	s1	s2	s3	RHS
0	x2	0	0.6667	1	0.3333	0	0	40
1	x2	0	7.3333	0	-1.3333	1	0	80
2	s3	0	-0.3333	0	-0.6667	0	1	20
3	Z	1	-3.3333	0	18.3333	0	0	2200

Resultado después del pivote

Solucion optima (metodo Simplex)

x1	x2
34.2857	17.1429
Z	2314.2857

Figura 1. Interfaz del software simplex con resultados de iteraciones y solución óptima

IV. CONCLUSIONES

El desarrollo del software permitió materializar una herramienta computacional que integra fundamentos teóricos de la programación lineal con una implementación robusta del método simplex. La arquitectura modular, que separa cálculo, visualización, interfaz gráfica y procesamiento de lenguaje natural, asegura claridad, escalabilidad y facilidad de mantenimiento. El sistema demostró ser capaz de resolver problemas de programación lineal de manera confiable, mostrando tanto la secuencia de iteraciones como la solución final, lo que refuerza su valor pedagógico.

La incorporación de un módulo de inteligencia artificial representa un aporte innovador al permitir que el usuario formule problemas en lenguaje natural, disminuyendo la barrera de entrada para quienes no están familiarizados con la notación matemática formal. Aunque este componente se encuentra en fase experimental, su desempeño sugiere un camino prometedor hacia la construcción de sistemas híbridos entre optimización matemática e interfaces de usuario más accesibles.

En conclusión, el proyecto constituye un prototipo que vincula la teoría con la práctica, reforzando el aprendizaje del método simplex e introduciendo un enfoque moderno basado en interfaces gráficas y procesamiento automático del lenguaje. Este desarrollo abre la posibilidad de futuras ampliaciones, como la exportación de resultados a distintos formatos, la incorporación de gráficos de regiones factibles y la integración con librerías de optimización más avanzadas.

BIBLIOGRAFIA

- [1] H. A. Taha, Investigación de operaciones, 9.^a ed. México: Pearson Educación, 2017.
- [2] F. S. Hillier y G. J. Lieberman, Introducción a la investigación de operaciones, 10.^a ed. México: McGraw-Hill, 2021.
- [3] S. Boyd y L. Vandenberghe, Convex Optimization. Cambridge, Reino Unido: Cambridge University Press, 2004.
- [4] W. McKinney, Python for Data Analysis, 3rd ed. Sebastopol, CA: O'Reilly Media, 2022.
- [5] M. Summerfield, Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. Upper Saddle River, NJ: Prentice Hall, 2007.
- [6] J. D. Hunter, "Matplotlib: A 2D graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.