

Universitat Politècnica de Catalunya

FACULTAT D'INFORMÀTICA DE BARCELONA

PRÁCTICA 1 - IMPLEMENTACIÓN DEL SIMPLEX

OPTIMIZACIÓN

Grado en Inteligencia Artificial

Daniel Álvarez 23857151X

Albert Roca 48106974J

31/03/2025

Contents

1	Introducción	2
2	Descripción de la implementación usada	3
2.1	Parte 1: Funciones auxiliares	3
2.2	Parte 2: Inicializaciones de las Fases + Función principal del SIMPLEX	5
2.2.1	Inicialización de la fase 1:	5
2.2.2	Inicialización de la Fase 2:	6
2.2.3	Función principal: resolver_simplex()	7
2.2.4	Ejecutar fase 2:	8
2.3	if __name__ == "__main__"	8
3	Soluciones obtenidas para los problemas asignados:	10
3.1	Cjt. Datos 5	10
3.1.1	PL 1	10
3.1.2	PL 2	11
3.1.3	PL 3 - PROBLEMA NO FACTIBLE	12
3.1.4	PL 4 - PROBLEMA NO ACOTADO	13
3.2	Cjt. Datos 48	15
3.2.1	PL 1	15
3.2.2	PL 2	16
3.2.3	PL 3 - PROBLEMA NO FACTIBLE	17
3.2.4	PL 4 - PROBLEMA NO ACOTADO	17
4	Conclusiones	19

1 Introducción

Para la primera práctica de este curso hemos podido desarrollar e implementar de forma satisfactoria una versión completa del método Simplex en dos fases, con el objetivo de resolver problemas de programación lineal mediante una herramienta propia. A lo largo del proyecto, hemos diseñado una estructura modular que permite leer problemas desde archivo, gestionar la presencia de variables artificiales y ejecutar ambas fases del algoritmo de forma diferenciada.

Durante el proceso de implementación, nos han surgido distintos retos tanto conceptuales como técnicos. Uno de los principales ha sido asegurar la correcta unión entre la fase 1 y la fase 2, especialmente en lo referente al tratamiento de la base inicial y la eliminación de variables artificiales. Otro desafío relevante ha sido evitar bucles infinitos en casos degenerados, así como garantizar que la matriz base sea invertible en cada iteración.

En este informe detallaremos la metodología seguida, la estructura del código desarrollado, los resultados obtenidos al resolver distintos problemas, y un análisis de los errores más frecuentes. Este trabajo no solo permite afianzar la comprensión del algoritmo simplex, sino también aplicar dicha comprensión a la resolución computacional de problemas reales de optimización lineal.

2 Descripción de la implementación usada

El código se ha estructurado en un solo archivo, diferenciado de tres partes para mejorar la organización y la legibilidad. La primera parte, marcada en el archivo, consta de todas las funciones auxiliares que ejecutan cálculos específicos necesarios para el funcionamiento del algoritmo, como el valor de la función objetivo (z), los costes reducidos, la dirección básica factible, el valor de theta, y las actualizaciones de las variables en cada iteración.

Por otro lado, la segunda parte principal se encarga de coordinar el flujo general del algoritmo Simplex, incluyendo la lectura de los datos de entrada, la inicialización de las fases y la gestión del ciclo iterativo que lleva a la solución óptima. Esta separación facilita la reutilización del código, simplifica el proceso de depuración y permite una implementación más modular y mantenible.

Finalmente, la última parte se encarga de la ejecución del código llamando a otro archivo de lectura de datos que hemos creado, el cual devuelve la matriz A , el vector b y el vector c según el número de alumno y problema que el usuario ponga en la terminal.

2.1 Parte 1: Funciones auxiliares

El objetivo principal de este archivo es ejecutar las funciones de cálculos de distintos parámetros como hemos comentado antes ya sean como theta, la dirección básica factible y más. A continuación, explicaremos qué es lo que calcula cada función o lo que ejecuta.

calculo_z(C_b, X_b, C_n, X_n)

Calcula el valor de la función objetivo z en el estado actual con la siguiente fórmula:

$$z = C_b \cdot X_b + C_n \cdot X_n$$

donde C_b y C_n son los coeficientes asociados a las variables básicas y no básicas respectivamente, mientras que X_b y X_n son los valores actuales de variables mencionadas. Esta función se llama en distintos pasos del archivo **main.py**

calcular_costes_reducidos(C_n, C_b, B_inv, A_n)

Calcula el vector de costes reducidos:

$$r = C_n - C_b B^{-1} A_n$$

Este vector indica si la solución actual se puede mejorar. Si todos los valores de r son mayores o iguales que cero, entonces la solución se considera óptima, ya que no existe ninguna dirección que reduzca el valor de la función objetivo.

Los parámetros B^{-1} y A_n representan la inversa de la matriz formada por las columnas de las variables básicas y la matriz formada las variables no básicas. Estas matrices se obtienen durante la inicialización del algoritmo o se actualizan en cada iteración.

variable_entrada(costes_reducidos, vars_no_basicas)

Selecciona qué variable no básica entra en la base. Se selecciona la de menor índice usando la regla de Bland entre las que tienen un coste reducido negativo. Se realiza recorriendo los costes y se identificando los índices negativos.

direccion_basica_factible(B_inv, A_n, var_entrada, vars_no_basicas)

Calcula la dirección básica factible de a la variable de entrada:

$$d = -B^{-1}a_q$$

donde a_q es la columna de la matriz A_n que hace referencia a la variable que entra. Esta dirección se emplea para actualizar la solución básica factible.

theta(X_b, D_b, vars_basicas)

Esta función tiene como objetivo calcular el valor de la θ , este parámetro determina cuánto podemos avanzar en la dirección factible antes de que alguna variable básica llegue a cero, provocando su salida de la base.

De entrada, recorremos todos los elementos del vector de dirección D_b , que nos indica hacia dónde se moverían las variables básicas si incrementamos el valor de la variable entrante. Solo nos interesan los coeficientes de D_b que sean negativos. Por cada componente negativa $d_i < 0$, calculamos la división de $\theta_i = -x_b[i]/d_i$.

Continuando, almacenamos estos candidatos en una lista. Y una vez construida esta lista de posibles variables de salida, identificamos el menor valor de θ . Este valor nos asegura que seguimos en el espacio factible. Para resolver el caso de empate, aplicamos la regla de Bland: seleccionamos entre las variables empatadas aquella con el menor índice en la lista **vars_basicas**.

Si ningún componente de D_b es negativo, la función devuelve **None** y ∞ , indicando que no hay ninguna variable que pueda salir de la base y que el problema es no acotado en esa dirección.

actualizacion(...)

Esta función se encarga de actualizar, como bien dice su nombre, todos los parámetros después de terminar una iteración.

De entrada, se identifican las posiciones de la variable que sale de la base (**var_salida**) y la que entra (**var_entrada**) en las listas **vars_basicas** y **vars_no_basicas**. Luego, estas listas se actualizan intercambiando los valores correspondientes.

Seguidamente, se recalcula el vector de valores de las variables básicas. Para hacerlo, hemos creado una copia del vector anterior y modificado cada componente utilizando la dirección básica factible (**direccion**) y el valor de (**theta**). El valor en la posición de la variable saliente se reemplaza por **theta**, mientras que el resto calcula con la fórmula $x_i \leftarrow x_i + \theta \cdot d_i$, donde d_i es el componente correspondiente de la dirección.

Siguiendo con la función, actualizamos la matriz inversa de la base (**B_inv**) usando una matriz elemental E . Esta matriz se construye de forma que la nueva base se obtiene aplicando una transformación elemental sobre la base anterior. Así entonces, hemos modificado la columna correspondiente a la variable que sale

para incorporar la nueva dirección, siguiendo las fórmulas de actualización.

Después reemplazamos en la matriz **A_n**, la columna de la variable de salida por la variable de entrada extraída de la matriz original ampliada. También actualizamos los vectores de costes, reemplazando en cada uno el coste de la variable que sale por el de la que entra.

Finalmente, actualizamos el valor de la función objetivo z . Este nuevo valor se obtiene sumando al anterior el incremento producido por la variable que entra, dado por el producto $\theta \cdot r_q$, donde r_q es el coste reducido correspondiente. Una vez hecho este cálculo, comprobamos que la z haya disminuido. Así entonces, la función devuelve todas las estructuras actualizadas.

2.2 Parte 2: Inicializaciones de las Fases + Función principal del SIMPLEX

2.2.1 Inicialización de la fase 1:

La función **init_fase_1** tiene como propósito preparar todos los elementos necesarios para iniciar la fase 1 del algoritmo. Esta fase se utiliza cuando el problema original no tiene una solución básica inicial factible evidente, por lo que se introducen variables artificiales que permiten construir la solución.

Primeramente, se extraen las dimensiones de la matriz de restricciones A , de la que se pueden determinar el número de restricciones m y el número de variables originales **n_original**. También se calcula el número total de variables tras la ampliación con las artificiales, **n_total**, a partir de la matriz **A_artificial**, que ha sido construida concatenando una matriz identidad a la derecha de A .

Segundamente, se identifican los índices correspondientes a las variables artificiales, que son aquellas que ocupan las últimas columnas de **A_artificial**, y se crea una lista a la que le asignamos el nombre de las variables básicas iniciales. Entonces, las variables originales del problema se consideran como no básicas al inicio del algoritmo.

A continuación, se define un vector de costes auxiliar **c_auxiliar**, este funciona de la siguiente manera: se asigna un cero a todas las variables originales y un uno a todas las variables artificiales. Este se utiliza como función objetivo temporal durante la fase 1, y nuestro objetivo es minimizar la suma de las variables artificiales.

Continuando con la función, se construye la matriz de base B , que se toma como la matriz identidad de tamaño $m \times m$, ya que este es el hecho de seleccionar inicialmente las columnas de las variables artificiales como básicas. Se calcula su inversa B^{-1} , que como es la identidad se hace simplemente una copia de la matriz B . También se almacena una copia de la matriz A original, que representa las variables no básicas.

Los vectores de costes básicos y no básicos se extraen a partir del vector auxiliar: C_B contiene los costes de las variables artificiales y C_N los de las variables originales. Con la base definida, se calcula la solución básica inicial $x_B = B^{-1}b$, mientras que las variables no básicas se inicializan con valor cero.

Finalmente, se calcula el valor inicial de la función objetivo auxiliar z llamando a **calculo_z**, una función auxiliar que se explica más adelante. El valor de la z se utiliza para comprobar si la solución inicial ya es factible; en caso contrario, será necesario realizar iteraciones para reducir el valor de z a cero.

La función devuelve los vectores de índices de variables básicas, artificiales y no básicas; el vector de costes auxiliar; la base B y su inversa; la matriz A , los vectores de costes básicos y no básicos; las soluciones básicas y no básicas iniciales; y finalmente el valor inicial de z .

2.2.2 Inicialización de la Fase 2:

La función **init_fase2** es como la de la fase 1; se encarga de preparar las variables necesarias para comenzar la fase 2 del método del Simplex, una vez completada la fase anterior y eliminadas las variables artificiales.

Primeramente, se eliminan de la lista de variables no básicas aquellas que también pertenecen al conjunto de variables artificiales. Esto es necesario para garantizar que estas variables ya no participen en la resolución del problema durante la fase 2. Para eliminarlas, se recorre un bucle **while** que recorre la lista de las variables no básicas y elimina los elementos que estén presentes en la lista de las variables básicas.

Segundamente, se crea la matriz de variables no básicas, extrayendo las columnas de la matriz A correspondientes a los índices en las variables no básicas. Para realizarlo, se han convertido estos índices a un array de NumPy, y se utiliza la función **np.take** para extraer dichas columnas.

A continuación, se calcula \mathbf{X}_b multiplicando la inversa de la matriz base B^{-1} por el vector de los términos independientes de las restricciones \mathbf{b} , es decir:

$$X_b = B^{-1}b$$

Las variables no básicas \mathbf{X}_n se inicializan como un vector columna de ceros, ya que por definición su valor es cero en una SBF.

Seguidamente, se extraen los coeficientes de la función objetivo asociados a las variables básicas y no básicas. Los coeficientes de las básicas, C_b , y de las no básicas, C_n , se obtienen aplicando **np.take** sobre el vector de costes original c , utilizando los índices de las correspondientes variables ajustados a base cero.

Finalmente, se calcula el valor de la z invocando la función **calculo_z**, que como hemos comentado antes la explicaremos más adelante.

La función devuelve los siguientes elementos: C_b , C_n , X_b , X_n , A_n , y el valor actual de la función objetivo z .

2.2.3 Función principal: `resolver_simplex()`

Definición de la función

La función `resolver_simplex` es la función principal del método simplex. Esta coordina las distintas subfunciones necesarias para resolver el problema paso a paso. Recibe como parámetros la matriz de restricciones, el vector de términos independientes, las variables básicas y no básicas, la función objetivo y otras estructuras intermedias que podemos haber generado previamente.

Fase 1:

Si se indica que el algoritmo se encuentra en la fase 1, el código imprime un mensaje de inicio. A continuación, calculamos el número de restricciones del problema a partir del número de filas de la matriz **A**. Construimos una matriz identidad del mismo tamaño y generamos la matriz aumentada **A_ext**, que concatena la matriz de restricciones con la identidad, lo cual permite introducir variables artificiales. Llamamos entonces a la función `init_fase1`, que inicializa las variables necesarias para comenzar el algoritmo: listas de variables básicas y no básicas, la matriz inversa de la base (**B_inv**), los coeficientes de la función auxiliar (**c_aux**), las matrices extendidas, los costes, el valor inicial de la función objetivo (**z**) y otros elementos. Se inicia el contador de iteraciones en 1 y calculamos los costes reducidos con la función `calcular_costes_reducidos`.

Fase 2:

Si se indica que el algoritmo se encuentra en la fase 2, el código muestra un mensaje correspondiente. En esta fase ya no trabajamos con la función objetivo auxiliar utilizada en la fase 1, sino con la función objetivo real del problema. Por ello, reconfiguramos las estructuras para ajustarlas a la nueva función objetivo. Llamamos a la función `init_fase2`, que prepara todas las variables necesarias: los coeficientes de la función objetivo correspondientes a las variables básicas (**Cb**) y no básicas (**Cn**), los valores actuales de las variables básicas (**Xb**), la matriz de columnas no básicas (**A_n**) y el valor inicial de la función objetivo (**z**). Además, copiamos la matriz inversa de la base (**B_inv**), que utilizaremos en las iteraciones para actualizar los cálculos. Establecemos el número de iteración a partir del parámetro `iter_inicial` y calculamos los costes reducidos en relación con la función objetivo. Estos costes nos indican si la solución actual es mejorable y en qué dirección debemos avanzar.

Bucle principal:

La parte principal del algoritmo consiste en un bucle que ejecutamos hasta alcanzar la optimalidad o hasta que detectamos que el algoritmo no puede continuar. Si todos los costes reducidos son mayores o iguales que cero, interpretamos que hemos llegado a una solución óptima. En la fase 1, si el valor de **z** sigue siendo positivo, concluimos que no existe una solución factible y devolvemos **None**. En caso contrario, determinamos la variable que entra en la base utilizando la función `variable_entrada`. Si no hay ninguna variable candidata, consideramos que ya hemos alcanzado una solución óptima. A continuación, calculamos la dirección básica factible mediante la función `direccion_basica_factible`, e identificamos qué variable debe salir de la base y cuál es el valor de **theta**, utilizando la función `theta`. Si no encontramos una variable de salida válida o el paso es infinito, concluimos que el problema no está acotado y devolvemos **None**.

Actualización de variables:

Agrupamos todos los elementos necesarios para la actualización y llamamos a la función `actualizacion`. Esta función nos devuelve las nuevas variables básicas y no básicas, los nuevos valores básicos, el nuevo

valor de la función objetivo y las matrices actualizadas. Tras cada iteración, imprimimos un resumen con la variable que entra, la que sale, el valor de `theta` y el valor actualizado de `z`. Finalmente, volvemos a calcular los costes reducidos para la siguiente iteración.

Finalización:

Si estamos en la fase 1, imprimimos un mensaje indicando que la fase ha finalizado, junto con el número total de iteraciones, y devolvemos todas las estructuras necesarias para iniciar la fase 2. Si estamos en la fase 2, mostramos un resumen final con el valor óptimo encontrado, las variables básicas finales, sus valores y los costes reducidos. La función devuelve toda esta información para que podamos analizar la solución final del problema.

2.2.4 Ejecutar fase 2:

Esta función es la encargada de continuar con la resolución del problema una vez que hemos finalizado con éxito la fase 1 del método del simplex. A esta función le pasamos tres argumentos: el resultado obtenido al finalizar la fase 1, la función objetivo original del problema, y el vector de términos independientes b . Lo primero que hacemos dentro de esta función es desempaquetar el resultado de la fase 1. Este resultado contiene toda la información necesaria para seguir con la fase 2 sin volver a repetir cálculos: el número de iteraciones realizadas, las variables básicas y no básicas al finalizar la fase anterior, los costes asociados, la matriz A_n , la solución básica X_b , el valor de z , la inversa de la base y la matriz A ampliada con las variables artificiales. Una vez tenemos todos estos elementos listos, simplemente llamamos a la función `resolver_simplex` con `fase=2` y le pasamos los datos correctos como argumentos. De esta forma, podemos reutilizar la misma lógica del algoritmo que usamos en la fase 1, pero ahora partiendo de una base factible ya obtenida y usando la función objetivo real del problema.

2.3 if `__name__ == "__main__"`

Esta parte del código implementado representa el punto de entrada de la ejecución del algoritmo. El objetivo es guiar al usuario que esté usando el programa para que vea cómo resuelve el algoritmo creado los problemas que le pase el usuario.

En primer lugar, el programa pregunta al usuario dos datos en concreto, el número de alumno y el número de problema. Si el usuario introduce un valor no numérico, el programa devuelve error y finaliza la ejecución.

Seguidamente, con los datos que le haya pasado el usuario al programa, este llama a la función principal del archivo de lectura de datos que se ha creado para la práctica. La función de este archivo busca entre todos los problemas que hay el ID del alumno con su problema respectivo. Esta devuelve la función objetivo, la matriz de restricciones y el vector de términos independientes. En caso de que alguno de estos valores no se haya cargado correctamente, el programa termina mostrando un mensaje de error.

Queremos recalcar que para que el código pueda funcionar correctamente se traspone el vector b para asegurar la compatibilidad en las operaciones matriciales posteriores. Una vez preparados todos los datos, el algoritmo entra en la primera fase del método del simplex, invocando la función `resolver_simplex()` con la fase 1, la matriz A i el vector b . Si la fase 1 devuelve un resultado distinto de `None`, que nos indica que se ha encontrado una solución factible, lo que se realiza es ejecutar la fase 2 con la función `ejecutar_fase2()`,

pasando como parámetros los resultados de la fase 1, junto con la función objetivo y el vector b . Por otra parte, la fase 1 determina que el problema no es factible, se imprime un mensaje nos que indica la finalización del programa.

3 Soluciones obtenidas para los problemas asignados:

3.1 Cjt. Datos 5

3.1.1 PL 1

```
-----  
----- INICIANDO FASE 1 -----  
Z inicial: 2779.000  
Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]  
Iteración 1: v_entrada = 1, v_salida = 24, = 1.050, z = 2774.800  
Iteración 2: v_entrada = 2, v_salida = 28, = 0.580, z = 2534.334  
Iteración 3: v_entrada = 3, v_salida = 29, = 2.782, z = 1880.793  
Iteración 4: v_entrada = 4, v_salida = 27, = 0.152, z = 1750.962  
Iteración 5: v_entrada = 6, v_salida = 23, = 1.512, z = 1238.437  
Iteración 6: v_entrada = 7, v_salida = 21, = 2.238, z = 566.397  
Iteración 7: v_entrada = 10, v_salida = 25, = 0.052, z = 562.773  
Iteración 8: v_entrada = 8, v_salida = 6, = 0.689, z = 450.696  
Iteración 9: v_entrada = 11, v_salida = 26, = 0.000, z = 450.552  
Iteración 10: v_entrada = 6, v_salida = 10, = 1.325, z = 270.446  
Iteración 11: v_entrada = 9, v_salida = 8, = 2.092, z = 207.963  
Iteración 12: v_entrada = 12, v_salida = 30, = 0.540, z = 100.084  
Iteración 13: v_entrada = 5, v_salida = 6, = 0.486, z = 51.192  
Iteración 14: v_entrada = 8, v_salida = 11, = 0.952, z = 30.040  
Iteración 15: v_entrada = 13, v_salida = 22, = 0.173, z = 0.000  
Fase 1 completada tras 15 iteraciones.  
----- INICIANDO FASE 2 -----  
Iteración 16: v_entrada = 6, v_salida = 9, = 1.344, z = -202.574  
Iteración 17: v_entrada = 10, v_salida = 8, = 1.526, z = -391.304  
Iteración 18: v_entrada = 11, v_salida = 12, = 0.111, z = -399.095  
Iteración 19: v_entrada = 15, v_salida = 11, = 61.752, z = -470.691  
Iteración 20: v_entrada = 8, v_salida = 7, = 0.221, z = -522.241  
Iteración 21: v_entrada = 9, v_salida = 8, = 0.168, z = -530.675  
Iteración 22: v_entrada = 11, v_salida = 5, = 0.407, z = -545.542  
Iteración 23: v_entrada = 17, v_salida = 10, = 20.189, z = -577.637  
Iteración 24: v_entrada = 5, v_salida = 11, = 0.107, z = -638.269  
Iteración 25: v_entrada = 16, v_salida = 5, = 221.149, z = -665.782  
Iteración 26: v_entrada = 7, v_salida = 3, = 0.008, z = -666.097  
Iteración 27: v_entrada = 18, v_salida = 15, = 42.115, z = -670.537  
Iteración 28: v_entrada = 3, v_salida = 9, = 0.789, z = -673.199  
Iteración 29: v_entrada = 19, v_salida = 16, = 56.092, z = -715.113  
Iteración 30: v_entrada = 8, v_salida = 7, = 0.117, z = -721.167  
Iteración 31: v_entrada = 9, v_salida = 8, = 0.148, z = -729.894  
Fase 2 completada tras 31 iteraciones.  
-----
```

```

----- RESULTADOS -----
z*: -729.893842
Vb*: [3, 13, 6, 1, 19, 17, 4, 2, 9, 18]
Valores básicos: [ 1.28957655  2.0040141  2.06433283  2.31286924  76.75136029
 138.88728824  1.81734658  3.81167449  0.14825813 128.44597736]
r*: [7.98948200e+01 7.43294993e+01 1.06433382e+02 1.42559891e+01
 2.05625135e+02 1.37729761e+02 9.76151941e+01 5.48393647e-01
 1.77075646e-01 2.84591704e-01]

```

3.1.2 PL 2

```

----- INICIANDO FASE 1 -----
Z inicial: 2083.000
Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
Iteración 1: v_entrada = 2, v_salida = 29, = 0.568, z = 1865.054
Iteración 2: v_entrada = 1, v_salida = 21, = 0.400, z = 1571.848
Iteración 3: v_entrada = 3, v_salida = 23, = 0.336, z = 1420.418
Iteración 4: v_entrada = 4, v_salida = 3, = 0.343, z = 1354.159
Iteración 5: v_entrada = 6, v_salida = 26, = 0.566, z = 1204.891
Iteración 6: v_entrada = 5, v_salida = 30, = 0.301, z = 1186.357
Iteración 7: v_entrada = 3, v_salida = 2, = 0.322, z = 1174.339
Iteración 8: v_entrada = 7, v_salida = 24, = 0.133, z = 1149.975
Iteración 9: v_entrada = 8, v_salida = 25, = 1.653, z = 862.315
Iteración 10: v_entrada = 2, v_salida = 28, = 1.289, z = 707.087
Iteración 11: v_entrada = 9, v_salida = 27, = 1.281, z = 474.438
Iteración 12: v_entrada = 10, v_salida = 2, = 0.115, z = 461.870
Iteración 13: v_entrada = 11, v_salida = 7, = 2.638, z = 143.839
Iteración 14: v_entrada = 12, v_salida = 4, = 0.247, z = 44.983
Iteración 15: v_entrada = 7, v_salida = 22, = 0.595, z = -0.000
Fase 1 completada tras 15 iteraciones.

```

```

----- INICIANDO FASE 2 -----
Iteración 16: v_entrada = 4, v_salida = 9, = 0.110, z = 73.863
Iteración 17: v_entrada = 13, v_salida = 1, = 0.621, z = 9.996
Iteración 18: v_entrada = 14, v_salida = 11, = 0.359, z = -27.454
Iteración 19: v_entrada = 17, v_salida = 4, = 106.380, z = -52.408
Iteración 20: v_entrada = 18, v_salida = 13, = 7.464, z = -58.534
Iteración 21: v_entrada = 4, v_salida = 17, = 0.285, z = -122.278
Iteración 22: v_entrada = 9, v_salida = 8, = 1.132, z = -285.386
Iteración 23: v_entrada = 2, v_salida = 4, = 0.351, z = -296.456
Iteración 24: v_entrada = 11, v_salida = 2, = 0.202, z = -313.326
Iteración 25: v_entrada = 13, v_salida = 10, = 0.204, z = -358.907
Iteración 26: v_entrada = 20, v_salida = 13, = 85.484, z = -431.074
Iteración 27: v_entrada = 4, v_salida = 7, = 0.136, z = -461.437

```

Iteración 28: v_entrada = 2, v_salida = 11, = 0.622, z = -468.945
 Iteración 29: v_entrada = 10, v_salida = 2, = 0.604, z = -483.263
 Iteración 30: v_entrada = 16, v_salida = 3, = 76.728, z = -545.822
 Iteración 31: v_entrada = 11, v_salida = 4, = 0.641, z = -568.146
 Fase 2 completada tras 31 iteraciones.

----- RESULTADOS -----

z*: -568.146104
 Vb*: [18, 11, 12, 14, 9, 6, 10, 20, 16, 5]
 Valores básicos: [7.82355719e+02 6.41062295e-01 3.55307214e+00 1.06869745e+00
 2.30912641e+00 2.18186248e+00 7.81621750e-01 3.95316752e+02
 1.37627148e+02 2.90689844e+00]
 r*: [1.21588625e+02 1.18553727e+02 2.87820447e+02 4.11790900e+01
 7.13642188e-01 1.07643982e+02 8.09314822e-02 4.59723069e+01
 7.21364842e-01 1.50871988e+02]

3.1.3 PL 3 - PROBLEMA NO FACTIBLE

----- INICIANDO FASE 1 -----

Z inicial: 1638.000
 Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
 Iteración 1: v_entrada = 2, v_salida = 28, = 0.754, z = 1417.942
 Iteración 2: v_entrada = 3, v_salida = 30, = 0.636, z = 1189.692
 Iteración 3: v_entrada = 4, v_salida = 29, = 0.056, z = 1172.102
 Iteración 4: v_entrada = 5, v_salida = 2, = 0.770, z = 1139.498
 Iteración 5: v_entrada = 8, v_salida = 21, = 0.323, z = 1133.262
 Iteración 6: v_entrada = 7, v_salida = 4, = 0.003, z = 1133.097
 Iteración 7: v_entrada = 9, v_salida = 5, = 0.852, z = 998.625
 Iteración 8: v_entrada = 1, v_salida = 7, = 0.084, z = 998.028
 Iteración 9: v_entrada = 4, v_salida = 8, = 0.021, z = 997.868
 Iteración 10: v_entrada = 10, v_salida = 4, = 0.036, z = 993.776
 Iteración 11: v_entrada = 7, v_salida = 1, = 0.098, z = 990.265
 Iteración 12: v_entrada = 13, v_salida = 7, = 0.091, z = 985.498
 Iteración 13: v_entrada = 14, v_salida = 13, = 0.077, z = 983.700
 Iteración 14: v_entrada = 15, v_salida = 25, = 281.911, z = 701.789
 Iteración 15: v_entrada = 13, v_salida = 14, = 0.091, z = 700.827
 Iteración 16: v_entrada = 16, v_salida = 26, = 323.488, z = 377.339
 Iteración 17: v_entrada = 1, v_salida = 13, = 0.071, z = 364.217
 Iteración 18: v_entrada = 14, v_salida = 1, = 0.077, z = 360.384
 Iteración 19: v_entrada = 11, v_salida = 10, = 0.105, z = 359.009
 Iteración 20: v_entrada = 17, v_salida = 27, = 46.745, z = 312.265
 Iteración 21: v_entrada = 4, v_salida = 11, = 0.045, z = 304.008
 Iteración 22: v_entrada = 18, v_salida = 4, = 6.715, z = 303.758

Iteración 23: $v_{\text{entrada}} = 19$, $v_{\text{salida}} = 18$, $= 4.980$, $z = 303.627$

No se ha encontrado una solución factible.

Fase 1 ha determinado que el problema no es factible. THE END.

3.1.4 PL 4 - PROBLEMA NO ACOTADO

----- INICIANDO FASE 1 -----

Z inicial: 6763.000

Variables básicas iniciales: [25, 26, 27, 28, 29, 30, 31, 32, 33, 34]

Iteración 1: $v_{\text{entrada}} = 1$, $v_{\text{salida}} = 27$, $= 6.198$, $z = 3608.260$

Iteración 2: $v_{\text{entrada}} = 2$, $v_{\text{salida}} = 25$, $= 4.184$, $z = 1572.947$

Iteración 3: $v_{\text{entrada}} = 3$, $v_{\text{salida}} = 33$, $= 1.737$, $z = 1332.528$

Iteración 4: $v_{\text{entrada}} = 7$, $v_{\text{salida}} = 28$, $= 1.437$, $z = 1169.457$

Iteración 5: $v_{\text{entrada}} = 5$, $v_{\text{salida}} = 3$, $= 1.586$, $z = 1135.947$

Iteración 6: $v_{\text{entrada}} = 8$, $v_{\text{salida}} = 32$, $= 0.892$, $z = 654.375$

Iteración 7: $v_{\text{entrada}} = 6$, $v_{\text{salida}} = 34$, $= 0.888$, $z = 407.310$

Iteración 8: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 5$, $= 0.180$, $z = 406.412$

Iteración 9: $v_{\text{entrada}} = 9$, $v_{\text{salida}} = 4$, $= 0.113$, $z = 394.568$

Iteración 10: $v_{\text{entrada}} = 10$, $v_{\text{salida}} = 31$, $= 0.060$, $z = 384.651$

Iteración 11: $v_{\text{entrada}} = 11$, $v_{\text{salida}} = 2$, $= 0.281$, $z = 139.538$

Iteración 12: $v_{\text{entrada}} = 3$, $v_{\text{salida}} = 29$, $= 0.418$, $z = 119.193$

Iteración 13: $v_{\text{entrada}} = 5$, $v_{\text{salida}} = 3$, $= 0.550$, $z = 85.717$

Iteración 14: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 30$, $= 0.757$, $z = 31.235$

Iteración 15: $v_{\text{entrada}} = 12$, $v_{\text{salida}} = 11$, $= 1.275$, $z = 11.083$

Iteración 16: $v_{\text{entrada}} = 13$, $v_{\text{salida}} = 8$, $= 1.935$, $z = 10.587$

Iteración 17: $v_{\text{entrada}} = 14$, $v_{\text{salida}} = 26$, $= 0.162$, $z = -0.000$

Fase 1 completada tras 17 iteraciones.

----- INICIANDO FASE 2 -----

Iteración 18: $v_{\text{entrada}} = 8$, $v_{\text{salida}} = 13$, $= 1.749$, $z = -510.669$

Iteración 19: $v_{\text{entrada}} = 11$, $v_{\text{salida}} = 12$, $= 1.038$, $z = -597.875$

Iteración 20: $v_{\text{entrada}} = 13$, $v_{\text{salida}} = 6$, $= 2.554$, $z = -733.926$

Iteración 21: $v_{\text{entrada}} = 12$, $v_{\text{salida}} = 1$, $= 0.451$, $z = -740.020$

Iteración 22: $v_{\text{entrada}} = 15$, $v_{\text{salida}} = 8$, $= 17.318$, $z = -751.943$

Iteración 23: $v_{\text{entrada}} = 1$, $v_{\text{salida}} = 9$, $= 0.317$, $z = -779.899$

Iteración 24: $v_{\text{entrada}} = 8$, $v_{\text{salida}} = 12$, $= 0.970$, $z = -792.686$

Iteración 25: $v_{\text{entrada}} = 16$, $v_{\text{salida}} = 8$, $= 43.532$, $z = -800.786$

Iteración 26: $v_{\text{entrada}} = 19$, $v_{\text{salida}} = 7$, $= 52.554$, $z = -882.234$

Iteración 27: $v_{\text{entrada}} = 6$, $v_{\text{salida}} = 11$, $= 2.282$, $z = -982.669$

Iteración 28: $v_{\text{entrada}} = 17$, $v_{\text{salida}} = 13$, $= 13.821$, $z = -983.130$

Iteración 29: $v_{\text{entrada}} = 18$, $v_{\text{salida}} = 17$, $= 12.888$, $z = -997.892$

Iteración 30: $v_{\text{entrada}} = 11$, $v_{\text{salida}} = 14$, $= 1.225$, $z = -1088.007$

Iteración 31: $v_{\text{entrada}} = 20$, $v_{\text{salida}} = 16$, $= 109.477$, $z = -1160.760$

Iteración 32: $v_{\text{entrada}} = 9$, $v_{\text{salida}} = 6$, $= 0.022$, $z = -1162.425$

Iteración 33: $v_{\text{entrada}} = 14$, $v_{\text{salida}} = 9$, $= 0.007$, $z = -1162.508$
Iteración 34: $v_{\text{entrada}} = 21$, $v_{\text{salida}} = 14$, $= 0.468$, $z = -1162.966$
Iteración 35: $v_{\text{entrada}} = 16$, $v_{\text{salida}} = 20$, $= 130.250$, $z = -1313.690$
Iteración 36: $v_{\text{entrada}} = 9$, $v_{\text{salida}} = 11$, $= 3.948$, $z = -1648.431$
Iteración 37: $v_{\text{entrada}} = 3$, $v_{\text{salida}} = 4$, $= 1.183$, $z = -1700.219$
Iteración 38: $v_{\text{entrada}} = 20$, $v_{\text{salida}} = 3$, $= 30.538$, $z = -1707.829$
Iteración 39: $v_{\text{entrada}} = 22$, $v_{\text{salida}} = 1$, $= 84.118$, $z = -1982.999$
Iteración 40: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 5$, $= 5.376$, $z = -2184.690$
Iteración 41: $v_{\text{entrada}} = 17$, $v_{\text{salida}} = 4$, $= 154.281$, $z = -2360.800$
Iteración 42: $v_{\text{entrada}} = 23$, $v_{\text{salida}} = 10$, $= 10485353683789312000.000$, $z = -12163010273195599872.000$
Problema no acotado.

3.2 Cjt. Datos 48

3.2.1 PL 1

```
-----
----- INICIANDO FASE 1 -----
Z inicial: 2191.000
Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
Iteración 1: v_entrada = 1, v_salida = 23, = 0.257, z = 2110.257
Iteración 2: v_entrada = 3, v_salida = 26, = 0.182, z = 2107.953
Iteración 3: v_entrada = 4, v_salida = 29, = 0.238, z = 2036.957
Iteración 4: v_entrada = 2, v_salida = 27, = 2.612, z = 1785.755
Iteración 5: v_entrada = 5, v_salida = 30, = 0.879, z = 1632.252
Iteración 6: v_entrada = 7, v_salida = 4, = 0.351, z = 1441.684
Iteración 7: v_entrada = 8, v_salida = 22, = 0.702, z = 1388.770
Iteración 8: v_entrada = 10, v_salida = 2, = 2.405, z = 829.133
Iteración 9: v_entrada = 4, v_salida = 5, = 0.232, z = 767.906
Iteración 10: v_entrada = 11, v_salida = 28, = 0.046, z = 734.219
Iteración 11: v_entrada = 2, v_salida = 1, = 0.793, z = 688.856
Iteración 12: v_entrada = 6, v_salida = 2, = 0.959, z = 584.035
Iteración 13: v_entrada = 12, v_salida = 21, = 0.268, z = 332.342
Iteración 14: v_entrada = 1, v_salida = 4, = 0.276, z = 322.377
Iteración 15: v_entrada = 2, v_salida = 25, = 0.408, z = 199.318
Iteración 16: v_entrada = 13, v_salida = 1, = 0.935, z = 189.474
Iteración 17: v_entrada = 5, v_salida = 10, = 0.960, z = 188.921
Iteración 18: v_entrada = 9, v_salida = 12, = 0.471, z = 188.451
Iteración 19: v_entrada = 14, v_salida = 24, = 0.699, z = 0.000
Fase 1 completada tras 19 iteraciones.
----- INICIANDO FASE 2 -----
Iteración 20: v_entrada = 1, v_salida = 5, = 0.633, z = -497.188
Iteración 21: v_entrada = 4, v_salida = 11, = 0.242, z = -523.229
Iteración 22: v_entrada = 5, v_salida = 8, = 0.155, z = -571.066
Iteración 23: v_entrada = 10, v_salida = 5, = 0.255, z = -599.574
Iteración 24: v_entrada = 11, v_salida = 1, = 0.360, z = -647.086
Iteración 25: v_entrada = 12, v_salida = 13, = 0.781, z = -767.780
Iteración 26: v_entrada = 16, v_salida = 6, = 192.960, z = -815.277
Iteración 27: v_entrada = 8, v_salida = 2, = 0.015, z = -815.474
Iteración 28: v_entrada = 17, v_salida = 8, = 2.703, z = -816.138
Iteración 29: v_entrada = 18, v_salida = 16, = 273.768, z = -880.019
Fase 2 completada tras 29 iteraciones.
-----
----- RESULTADOS -----
z*: -880.019266
Vb*: [9, 10, 18, 14, 17, 3, 11, 4, 7, 12]
```


Valores básicos: [2.60577195 3.43429796 273.76791858 0.57911628 148.92446464
1.08503138 1.10698349 1.74568786 1.75368328 1.06891412]
r*: [5.46356125e+01 9.00554716e-02 8.89117657e+01 1.36142176e+02
7.81794074e-01 1.22294647e+02 5.63032571e+01 3.19309313e-01
1.98040923e-02 5.20844758e-01]

3.2.2 PL 2

----- INICIANDO FASE 1 -----
Z inicial: 2662.000
Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
Iteración 1: v_entrada = 1, v_salida = 21, = 0.671, z = 2498.934
Iteración 2: v_entrada = 2, v_salida = 29, = 0.635, z = 2405.629
Iteración 3: v_entrada = 3, v_salida = 24, = 2.830, z = 1526.238
Iteración 4: v_entrada = 4, v_salida = 30, = 0.040, z = 1518.344
Iteración 5: v_entrada = 5, v_salida = 4, = 0.022, z = 1515.306
Iteración 6: v_entrada = 6, v_salida = 5, = 0.045, z = 1508.220
Iteración 7: v_entrada = 7, v_salida = 22, = 0.434, z = 1131.391
Iteración 8: v_entrada = 4, v_salida = 28, = 0.184, z = 970.098
Iteración 9: v_entrada = 5, v_salida = 1, = 0.842, z = 233.627
Iteración 10: v_entrada = 9, v_salida = 23, = 0.026, z = 210.320
Iteración 11: v_entrada = 1, v_salida = 27, = 0.274, z = 130.834
Iteración 12: v_entrada = 8, v_salida = 25, = 0.087, z = 94.581
Iteración 13: v_entrada = 11, v_salida = 8, = 0.045, z = 75.410
Iteración 14: v_entrada = 10, v_salida = 26, = 0.646, z = -0.000
Fase 1 completada tras 14 iteraciones.

----- INICIANDO FASE 2 -----
Iteración 15: v_entrada = 8, v_salida = 5, = 0.734, z = 196.211
Iteración 16: v_entrada = 12, v_salida = 4, = 0.053, z = 185.383
Iteración 17: v_entrada = 14, v_salida = 7, = 0.062, z = 166.821
Iteración 18: v_entrada = 4, v_salida = 12, = 0.218, z = 165.699
Iteración 19: v_entrada = 5, v_salida = 3, = 0.424, z = 62.226
Iteración 20: v_entrada = 12, v_salida = 4, = 1.562, z = -111.012
Iteración 21: v_entrada = 7, v_salida = 10, = 0.749, z = -127.775
Iteración 22: v_entrada = 16, v_salida = 2, = 20.110, z = -168.589
Iteración 23: v_entrada = 3, v_salida = 7, = 0.051, z = -212.120
Iteración 24: v_entrada = 17, v_salida = 6, = 116.701, z = -238.514
Iteración 25: v_entrada = 15, v_salida = 9, = 453.710, z = -266.632
Fase 2 completada tras 25 iteraciones.

----- RESULTADOS -----
z*: -266.631836
Vb*: [8, 14, 15, 5, 11, 3, 1, 12, 16, 17]

Valores básicos: [2.15152202e+00 1.63352822e+00 4.53709904e+02 9.74151034e-01
7.44810125e-02 1.80668608e-01 1.91756726e+00 1.23933173e+00
1.80071753e+02 2.97186329e+02]
r*: [1.11027075e+02 6.13436525e+01 1.33242852e+02 8.96052476e+01
1.45917738e+01 9.06577636e+01 3.12128627e+01 1.31830645e-01
6.77179101e-01 1.41598841e-01]

3.2.3 PL 3 - PROBLEMA NO FACTIBLE

```

----- INICIANDO FASE 1 -----
Z inicial: 1224.000
Variables básicas iniciales: [21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
Iteración 1: v_entrada = 1, v_salida = 29,  = 0.153, z = 1189.831
Iteración 2: v_entrada = 3, v_salida = 1,  = 0.107, z = 1188.750
Iteración 3: v_entrada = 6, v_salida = 3,  = 0.100, z = 1176.000
Iteración 4: v_entrada = 8, v_salida = 22,  = 0.273, z = 1120.907
Iteración 5: v_entrada = 1, v_salida = 6,  = 0.030, z = 1087.720
Iteración 6: v_entrada = 14, v_salida = 30,  = 0.424, z = 1052.962
Iteración 7: v_entrada = 4, v_salida = 1,  = 0.006, z = 1052.662
Iteración 8: v_entrada = 11, v_salida = 4,  = 0.004, z = 1052.447
Iteración 9: v_entrada = 16, v_salida = 26,  = 285.825, z = 766.622
Iteración 10: v_entrada = 1, v_salida = 11,  = 0.005, z = 766.569
Iteración 11: v_entrada = 17, v_salida = 27,  = 194.135, z = 572.433
Iteración 12: v_entrada = 12, v_salida = 1,  = 0.026, z = 570.547
Iteración 13: v_entrada = 18, v_salida = 28,  = 35.449, z = 535.098
Iteración 14: v_entrada = 20, v_salida = 14,  = 16.488, z = 496.712
Iteración 15: v_entrada = 22, v_salida = 8,  = 13.000, z = 484.500
No se ha encontrado una solución factible.
Fase 1 ha determinado que el problema no es factible. THE END.

```

3.2.4 PL 4 - PROBLEMA NO ACOTADO

```

----- INICIANDO FASE 1 -----
Z inicial: 6945.000
Variables básicas iniciales: [25, 26, 27, 28, 29, 30, 31, 32, 33, 34]
Iteración 1: v_entrada = 1, v_salida = 32,  = 6.626, z = 3956.505
Iteración 2: v_entrada = 2, v_salida = 33,  = 7.001, z = 1492.461
Iteración 3: v_entrada = 3, v_salida = 27,  = 1.537, z = 1067.975
Iteración 4: v_entrada = 5, v_salida = 31,  = 1.390, z = 771.220
Iteración 5: v_entrada = 4, v_salida = 30,  = 1.388, z = 661.750
Iteración 6: v_entrada = 6, v_salida = 4,  = 1.806, z = 384.709
Iteración 7: v_entrada = 7, v_salida = 34,  = 0.504, z = 357.354
Iteración 8: v_entrada = 9, v_salida = 29,  = 0.570, z = 322.543

```

Iteración 9: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 1$, $\theta = 0.242$, $z = 264.392$
 Iteración 10: $v_{\text{entrada}} = 10$, $v_{\text{salida}} = 28$, $\theta = 0.421$, $z = 128.203$
 Iteración 11: $v_{\text{entrada}} = 1$, $v_{\text{salida}} = 26$, $\theta = 0.443$, $z = 121.181$
 Iteración 12: $v_{\text{entrada}} = 8$, $v_{\text{salida}} = 1$, $\theta = 0.262$, $z = 68.701$
 Iteración 13: $v_{\text{entrada}} = 11$, $v_{\text{salida}} = 25$, $\theta = 0.487$, $z = 0.000$

Fase 1 completada tras 13 iteraciones.

----- INICIANDO FASE 2 -----

Iteración 14: $v_{\text{entrada}} = 12$, $v_{\text{salida}} = 10$, $\theta = 2.377$, $z = -943.215$
 Iteración 15: $v_{\text{entrada}} = 13$, $v_{\text{salida}} = 4$, $\theta = 0.140$, $z = -946.135$
 Iteración 16: $v_{\text{entrada}} = 15$, $v_{\text{salida}} = 11$, $\theta = 76.100$, $z = -1073.440$
 Iteración 17: $v_{\text{entrada}} = 1$, $v_{\text{salida}} = 2$, $\theta = 0.787$, $z = -1112.020$
 Iteración 18: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 1$, $\theta = 0.066$, $z = -1132.677$
 Iteración 19: $v_{\text{entrada}} = 16$, $v_{\text{salida}} = 4$, $\theta = 6.781$, $z = -1133.431$
 Iteración 20: $v_{\text{entrada}} = 17$, $v_{\text{salida}} = 6$, $\theta = 9.156$, $z = -1147.226$
 Iteración 21: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 16$, $\theta = 0.153$, $z = -1148.584$
 Iteración 22: $v_{\text{entrada}} = 18$, $v_{\text{salida}} = 8$, $\theta = 124.506$, $z = -1265.273$
 Iteración 23: $v_{\text{entrada}} = 16$, $v_{\text{salida}} = 4$, $\theta = 190.277$, $z = -1304.410$
 Iteración 24: $v_{\text{entrada}} = 19$, $v_{\text{salida}} = 7$, $\theta = 333.320$, $z = -1516.129$
 Iteración 25: $v_{\text{entrada}} = 20$, $v_{\text{salida}} = 5$, $\theta = 325.246$, $z = -1646.517$
 Iteración 26: $v_{\text{entrada}} = 4$, $v_{\text{salida}} = 12$, $\theta = 0.262$, $z = -1699.902$
 Iteración 27: $v_{\text{entrada}} = 7$, $v_{\text{salida}} = 4$, $\theta = 0.440$, $z = -1704.756$
 Iteración 28: $v_{\text{entrada}} = 21$, $v_{\text{salida}} = 7$, $\theta = 27.204$, $z = -1732.649$
 Iteración 29: $v_{\text{entrada}} = 22$, $v_{\text{salida}} = 21$, $\theta = 176.662$, $z = -1868.531$
 Iteración 30: $v_{\text{entrada}} = 5$, $v_{\text{salida}} = 13$, $\theta = 1.611$, $z = -2482.392$
 Iteración 31: $v_{\text{entrada}} = 23$, $v_{\text{salida}} = 5$, $\theta = 123.838$, $z = -2801.316$
 Iteración 32: $v_{\text{entrada}} = 21$, $v_{\text{salida}} = 15$, $\theta = 3509.556$, $z = -7420.635$
 Iteración 33: $v_{\text{entrada}} = 12$, $v_{\text{salida}} = 9$, $\theta = 15.714$, $z = -11633.810$
 Iteración 34: $v_{\text{entrada}} = 15$, $v_{\text{salida}} = 3$, $\theta = 25700.000$, $z = -41250.000$

Problema no acotado.

4 Conclusiones

A lo largo de esta práctica hemos implementado de forma completa el algoritmo del método del simplex, dividiendo su ejecución en dos fases claramente diferenciadas. Los resultados obtenidos muestran que el algoritmo es capaz de identificar correctamente tanto soluciones óptimas como situaciones de inviabilidad o no acotación del problema. Para poder confirmar esto, le pedimos a un par de compañeros que probaran nuestro código con sus soluciones, y el resultado en todos los casos fue satisfactorio.

Más allá de los resultados numéricos concretos, esta práctica nos ha permitido consolidar nuestra comprensión del funcionamiento interno del método del simplex. Hemos trabajado con estructuras clave como las matrices base, los vectores de costes reducidos y el cálculo iterativo de la solución. Además, se ha puesto en práctica el uso de técnicas de programación estructurada y modular, facilitando tanto la legibilidad como la depuración del código.

En definitiva, este trabajo no solo ha servido para entender un algoritmo fundamental en la optimización lineal, sino también para desarrollar habilidades prácticas en la implementación de métodos matemáticos complejos desde cero. La experiencia que obtenemos en este trabajo nos resulta valiosa no solo en el contexto académico, sino también como base para problemas más avanzados de programación matemática y optimización en aplicaciones reales.