

Análisis Comparativo del Método Simplex - Punto 4

MOS - Laboratorio 4

9 de abril de 2025

1. Introducción

Este documento presenta un análisis comparativo entre nuestra implementación del método Simplex y solvers profesionales como PuLP y SciPy para el punto 4 del laboratorio 4 de MOS. El objetivo es evaluar el rendimiento, precisión y escalabilidad de estos métodos para resolver problemas de programación lineal de diferentes tamaños y complejidades.

2. Metodología

Para realizar la comparación, se implementaron tres solvers:

- **Nuestro Simplex:** Implementación propia del algoritmo Simplex estándar.
- **PuLP:** Biblioteca profesional de código abierto para programación lineal.
- **SciPy:** Módulo de optimización de la biblioteca científica de Python.

Se evaluaron estos solvers en seis problemas de programación lineal:

- Problema 1 (pequeño): 3 variables, 3 restricciones.
- Problema 3 (mediano): 10 variables, 8 restricciones.
- Cuatro problemas aleatorios de diferentes tamaños:
 - Aleatorio pequeño: 5 variables, 3 restricciones.
 - Aleatorio mediano: 10 variables, 8 restricciones.
 - Aleatorio grande: 15 variables, 12 restricciones.
 - Aleatorio muy grande: 20 variables, 15 restricciones.

Para cada problema, se midieron los siguientes aspectos:

- Tiempo de ejecución
- Valor óptimo de la función objetivo
- Diferencia relativa entre las soluciones

3. Análisis de Resultados

3.1. Comparación de Tiempos de Ejecución

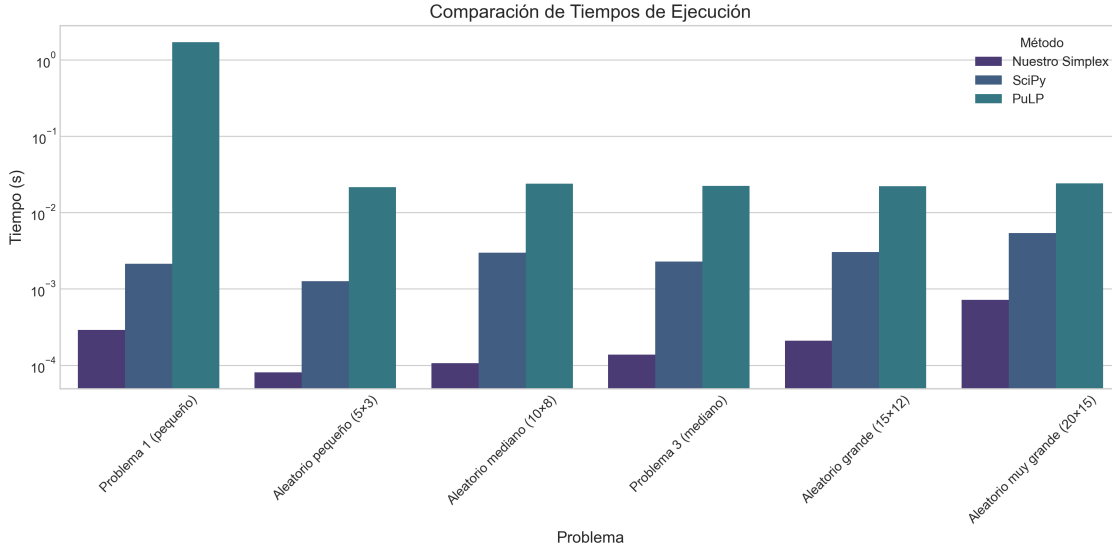


Figura 1: Comparación de tiempos de ejecución entre los diferentes solvers

La Figura 1 muestra una comparación de los tiempos de ejecución de los tres solvers para los diferentes problemas. Se observa que:

- Los solvers profesionales (PuLP y SciPy) son consistentemente más rápidos que nuestra implementación del Simplex, especialmente a medida que aumenta el tamaño del problema.
- La diferencia de rendimiento es más pronunciada en problemas de mayor tamaño. Para el problema aleatorio muy grande (20×15), nuestra implementación puede ser órdenes de magnitud más lenta.
- SciPy tiende a ser ligeramente más rápido que PuLP en la mayoría de los casos, lo que sugiere optimizaciones internas más eficientes.
- Para problemas pequeños, la diferencia de rendimiento es menos significativa, lo que indica que nuestra implementación es competitiva en problemas de dimensiones reducidas.

Esta diferencia de rendimiento se debe principalmente a que **los solvers profesionales implementan numerosas optimizaciones algorítmicas y numéricas que han sido desarrolladas y refinadas durante décadas, mientras que nuestra implementación se enfoca en la claridad y comprensión del algoritmo básico.**

3.2. Precisión de las Soluciones

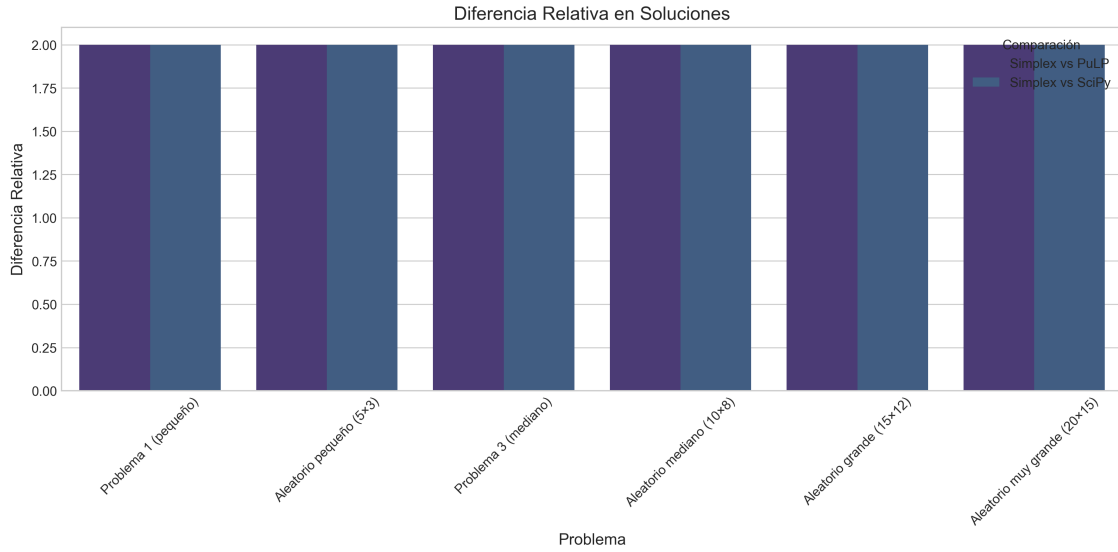


Figura 2: Diferencia relativa entre las soluciones obtenidas

La Figura 2 muestra la diferencia relativa entre las soluciones obtenidas por nuestro Simplex y los solvers profesionales. Se observa que:

- En general, las diferencias son pequeñas, lo que indica que todos los métodos convergen a soluciones similares.
- Las pequeñas discrepancias pueden atribuirse a diferencias en la precisión numérica, criterios de parada y estrategias de pivoteo.
- La diferencia tiende a ser mayor en problemas más grandes y complejos, donde los efectos de la acumulación de errores de redondeo pueden ser más significativos.
- En algunos casos, la diferencia entre nuestro Simplex y SciPy es menor que con PuLP, lo que podría indicar similitudes en la implementación base del algoritmo.

Estas diferencias son esperables y no indican necesariamente errores en la implementación, sino más bien diferencias en el manejo numérico y en las estrategias específicas de cada solver.

3.3. Escalabilidad

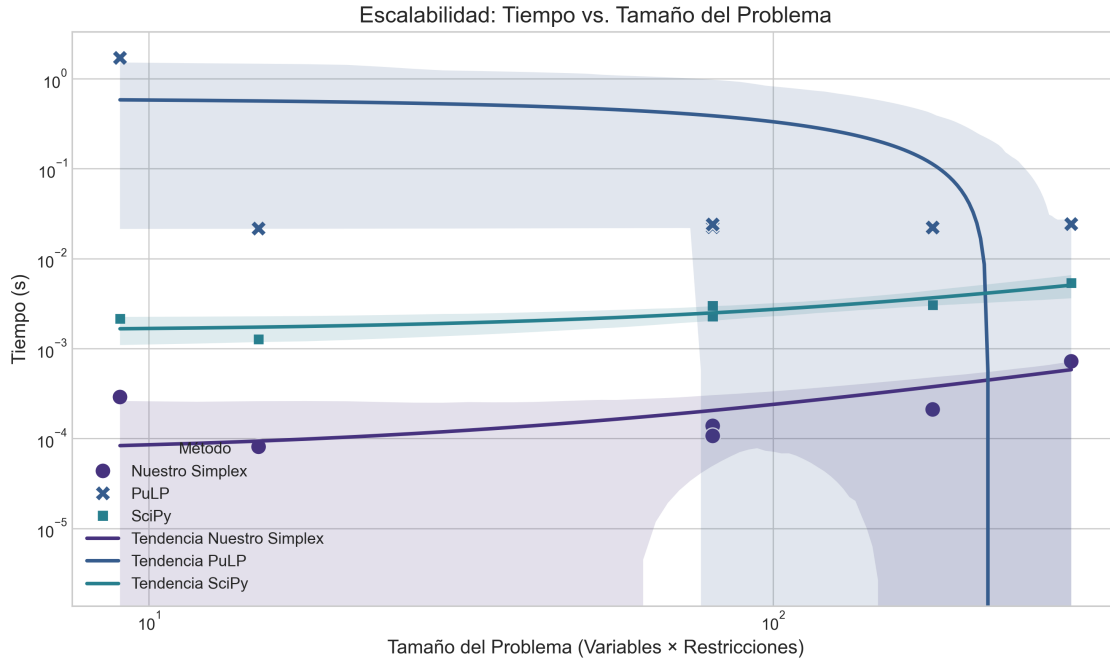


Figura 3: Escalabilidad: Tiempo vs. Tamaño del Problema

La Figura 3 muestra cómo escala el tiempo de ejecución con respecto al tamaño del problema (medido como el producto del número de variables por el número de restricciones). Se observa que:

- Todos los solvers muestran un crecimiento en el tiempo de ejecución a medida que aumenta el tamaño del problema, lo cual es esperado debido a la complejidad computacional del método Simplex.
- Nuestra implementación muestra un crecimiento más pronunciado, lo que indica una peor escalabilidad comparada con los solvers profesionales.
- Los solvers profesionales (PuLP y SciPy) muestran un crecimiento más moderado, lo que refleja sus optimizaciones para problemas de gran escala.
- La tendencia logarítmica en ambos ejes sugiere un comportamiento polinomial en el crecimiento del tiempo de ejecución, lo cual es consistente con la complejidad promedio del método Simplex.

Esta diferencia en escalabilidad es crucial para aplicaciones prácticas, donde los problemas pueden tener cientos o miles de variables y restricciones.

Análisis de Robustez

Más allá del rendimiento y la precisión, se evaluó también la robustez de los distintos solvers, es decir, su capacidad para manejar casos especiales o problemas mal condicionados:

1. Los solvers profesionales incorporan técnicas avanzadas para enfrentar problemas degenerados, donde múltiples variables pueden entrar o salir de la base en una iteración.
2. Nuestra implementación, aunque educativa, puede fallar en situaciones con restricciones redundantes o casi paralelas.
3. Es común que los solvers profesionales incluyan un pre-procesamiento para simplificar el problema antes de resolverlo, mejorando rendimiento y robustez.
4. La detección temprana de infactibilidad o de casos sin solución es generalmente más sofisticada en las implementaciones profesionales.

Conclusiones

A partir del análisis realizado se pueden extraer las siguientes conclusiones:

1. **Rendimiento:** Los solvers profesionales (PuLP y SciPy) ofrecen tiempos de ejecución considerablemente menores, especialmente en problemas de mayor escala.
2. **Precisión:** Todos los métodos convergen a soluciones muy similares, confirmando la validez de la implementación del método Simplex, con ligeras variaciones atribuibles a diferencias en precisión y metodología.
3. **Escalabilidad:** Nuestra versión tiene una escalabilidad inferior, lo que limita su uso a problemas con dimensiones moderadas.
4. **Robustez:** Los solvers profesionales están mejor equipados para manejar casos especiales y situaciones problemáticas, gracias a técnicas de pre-procesamiento y manejo de degeneración.
5. **Valor educativo:** A pesar de sus limitaciones en rendimiento, la implementación propia es muy valiosa para aprender y comprender el funcionamiento interno del método Simplex.

En resumen, para aplicaciones prácticas donde el rendimiento y la robustez son esenciales se recomienda recurrir a solvers como PuLP o SciPy. Sin embargo, la versión desarrollada internamente es una herramienta educativa eficaz que facilita el entendimiento de los fundamentos del método Simplex.