

Informe Taller 1

INFO200 - Investigación Operativa

Académica: Tania Letelier {tletelier@inf.uach.cl}
Instituto de Informática, Universidad Austral de Chile
Alumnos: - Renato Atencio {renato.atencio@alumnos.uach.cl}
- Felipe Córdova {felipe.cordova@alumnos.uach.cl}
- Sebastián Montecinos {sebastian.montecinos02@alumnos.uach.cl}

Octubre 07, 2023

1 Introducción

La resolución de problemas de programación lineal (PL) es fundamental en el campo de la optimización, y se aplica en una amplia variedad de situaciones cotidianas, desde la planificación de la producción hasta la gestión de recursos y la toma de decisiones en empresas y organizaciones. En este taller, nos enfocaremos en la aplicación de las técnicas de programación lineal para resolver problemas prácticos y visualizar sus soluciones utilizando distintas bibliotecas en el lenguaje de programación Python. El taller se divide en dos partes clave: la representación gráfica de las restricciones y la región factible, y la identificación del punto óptimo de la función objetivo.

2 Enunciado

Se requiere determinar la cantidad óptima de unidades de infantería y de fuerza aérea que deben asignarse a una armada para maximizar su poder de ataque, donde las unidades de infantería aportan 1 de poder de ataque y las unidades aéreas de 4. La armada dispone de un total de 73000 galones de gasolina y 15000 kg de acero. La creación de estas unidades tiene requisitos de recursos: la infantería necesita 20 galones de gasolina y 30 kg de acero por unidad, la fuerza aérea requiere 100 galones de gasolina y 10 kg de acero por unidad. Además, se establecen estándares militares los cuales indican que requiere un mínimo de 100 unidades para cada una de las ramas militares. Asimismo, se establecen restricciones adicionales: la cantidad de unidades de fuerza aérea no puede superar las 700 unidades.

2.1 Formulación Matemática

Función Objetivo:

$$\text{Máx poder} = x + 4y$$

Sujeto a las siguientes restricciones:

$$20x + 100y \leq 73000 \quad (1)$$

$$30x + 10y \leq 15000 \quad (2)$$

$$y \leq 700 \quad (3)$$

$$x, y \geq 100 \quad (4)$$

3 Desarrollo

Para optimizar esta función, seguimos los siguientes pasos:

- Encontrar los puntos de intersección entre las restricciones consideradas como igualdades. Para lograr esto, llamamos a la función `encontrar_interseccion` en nuestro código y luego almacenamos los puntos en un arreglo llamado `puntos_interseccion`.
- Determinar qué puntos cumplen con todas las restricciones en nuestro problema, evaluándolos en cada una de las restricciones. Los puntos que cumplan con todas las restricciones se almacenan en el arreglo `puntosPolinomio`.
- Eliminar los puntos repetidos.
- Una vez que tenemos los puntos, podemos evaluarlos en la función objetivo para obtener el valor máximo. También podemos utilizar la librería Pulp para obtener el valor máximo.

Para graficar los resultados, utilizamos las librerías NumPy y Matplotlib. Primero, graficamos las restricciones junto con las intersecciones de la zona factible. Luego, graficamos solo la zona factible junto con la solución óptima.

4 Resultados

Grafico con las restricciones. Las líneas en color rojo representan las restricciones de menor igual, y las azules son las restricciones de mayor o igual.

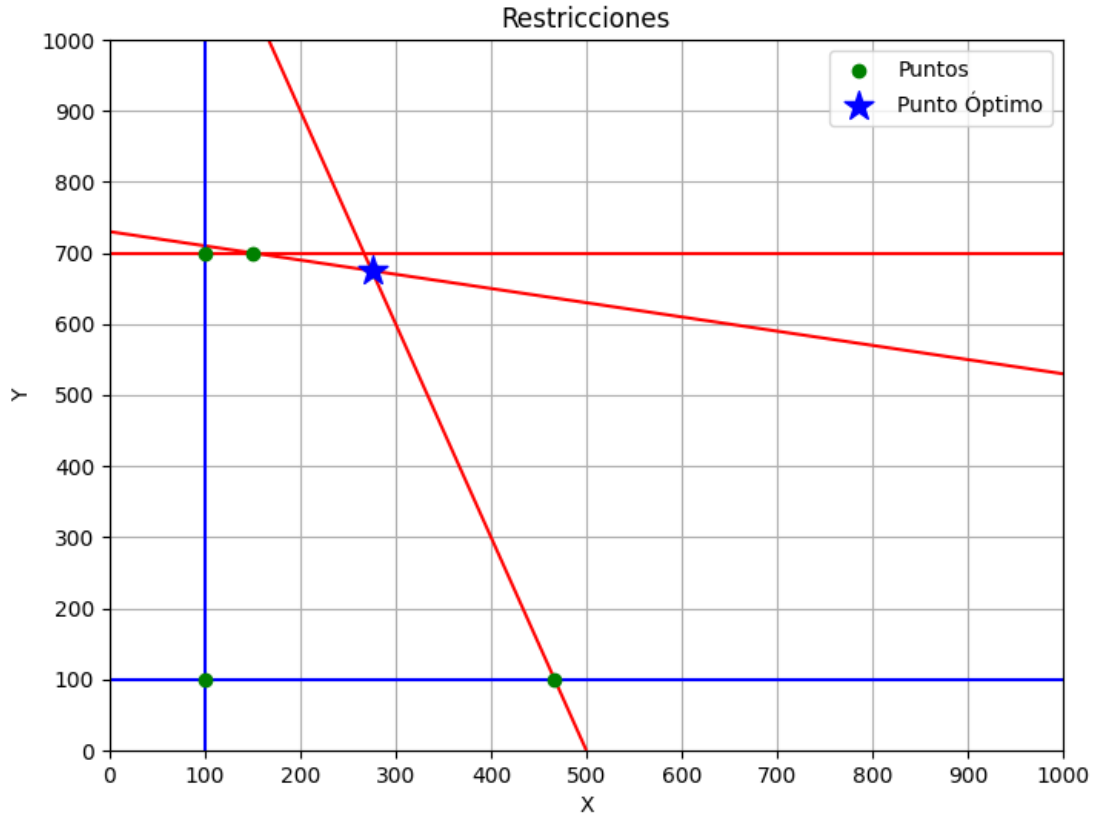
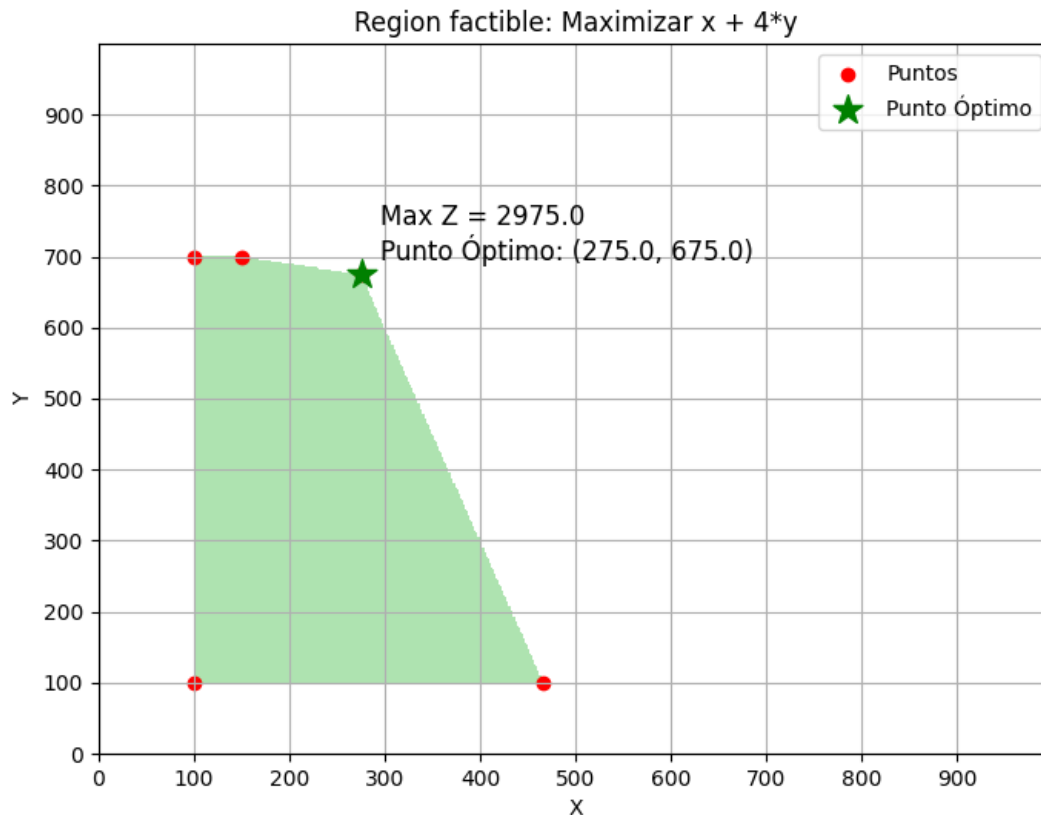


Grafico con la zona factible y la solución óptima.



5 Conclusión

En este trabajo, hemos explorado la resolución de un problema cotidiano mediante el método gráfico de programación lineal, utilizando una implementación en Python con las bibliotecas PuLP, NumPy, Shapely, Matplotlib y openpyxl. A lo largo del proceso, hemos formulado y abordado un problema de optimización que refleja decisiones comunes en la vida diaria, modelando una función objetivo y restricciones adecuadas. Nuestra solución permite al usuario interactuar con el programa mediante un archivo excel, ajustando coeficientes y restricciones para explorar diferentes escenarios. A través de visualizaciones claras y representativas generadas con Matplotlib, hemos ilustrado la región factible y los puntos óptimos, facilitando así la comprensión y la toma de decisiones en situaciones prácticas.

En resumen, este taller destaca la aplicabilidad de la programación lineal en la resolución de problemas del mundo real y demuestra cómo una implementación en Python con herramientas de visualización efectivas puede mejorar la comprensión y la toma de decisiones. La combinación de la modelización matemática, la interactividad y la representación gráfica de resultados ofrece una herramienta poderosa para abordar una variedad de situaciones de optimización en diversos contextos.