



M1
Rutas óptimas
y optimización
multiobjetiva
con Python

Temario

MOO: Optimización Multiobjetiva

Definición y características

Tipos

Ejemplos de resolución

Python al rescate:

Spicy...

Pulp

Pyomo



MOO: Optimización multiobjetivo

#StucomÉsDemà



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com



MOO

Fuente: <https://relopezbriega.github.io/blog/2017/01/18/problemas-de-optimizacion-con-python/>



MOO: ¿Qué és y para qué sirve ?

FUNCION OBJETIVO (1, N)

+

CONDICION (0,N)

+

VARIABLES (1,N)



MOO: ¿Qué és y para qué sirve ?

FUNCION OBJETIVO (1, N)
+
CONDICION (0,N)
+
VARIABLES (1,N)



MOO: ¿Qué és y para qué sirve ?

La **solución al problema de optimización** va a estar dada por **conjunto de valores permitidos** para las variables independientes **de acuerdo con las limitaciones del problema**, en los que la **función objetivo asume un valor óptimo.**



MOO: Clasificación

- Optimización continua versus optimización discreta
- Optimización sin restricciones versus optimización con restricciones
- Ninguna, una o varias funciones objetivo
- Optimización determinista versus optimización estocástica



MOO: Resolución

Objetivo: Modelización a un sistema matemático/algebráico

$$\begin{aligned} & \min \quad f_0(x) \\ & \text{sujeto a } f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned}$$



MOO: Resolución

Para resolver un problema de programación lineal, debemos seguir los siguientes pasos:

- Elegir las incógnitas o variables de decisión.
- Escribir la función objetivo en función de los datos del problema.
- Escribir las restricciones en forma de sistema de inecuaciones.
- Averiguar el conjunto de soluciones factibles representando gráficamente las restricciones.





MOO: Resolución

- Calcular las coordenadas de los vértices del recinto de soluciones factibles (son pocos).
- Calcular el valor de la función objetivo en cada uno de los vértices para ver en cuál de ellos presenta el valor máximo o mínimo según nos pida el problema (hay que tener en cuenta aquí la posible no existencia de solución).



MOO: ¿Qué és y para qué sirve ?

Fuente: <https://relopezbriega.github.io/blog/2017/01/18/problemas-de-optimizacion-con-python/>

#StucomÉsDemà



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com

AENOR
GESTIÓN
DE LA CALIDAD
ISO 9001

Librerías de Python

#StucomÉsDemà



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com

AENOR
GESTIÓN
DE LA CALIDAD
ISO 9001



<https://www.scipy.org/>

#StucomÉsDemà



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com





https://pulpproject.org/pulp_python/





<https://www.pyomo.org/>

#StucomÉsDemà



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com





PULP vs PYOMO

Pulp y Pyomo son dos bibliotecas populares en Python para la programación matemática pero tienen diferencias en su enfoque y características.

1. Enfoque y Diseño

Pulp: Es más sencilla y está diseñada para ser fácil de usar. Se centra en la creación de modelos de programación lineal y entera de manera rápida y directa.

Pyomo: Es más compleja y versátil, diseñada para modelar problemas de optimización amplios, incluyendo programación lineal, no lineal, entera y combinatoria.





PULP vs PYOMO

2. Sintaxis y Modelado

Pulp: Utiliza una sintaxis más simple y directa, lo que facilita la construcción de modelos básicos. Ideal para quienes buscan una solución rápida.

Pyomo: Proporciona una sintaxis más rica y permite un modelado más detallado y estructurado, lo que es útil para problemas más complejos.





PULP vs PYOMO

3. Funciones y Capacidades

Pulp: Ideal para problemas sencillos y de tamaño moderado. Sus capacidades son adecuadas para la mayoría de las necesidades básicas de optimización.

Pyomo: Ofrece más funciones avanzadas, como la posibilidad de definir parámetros dinámicos, y compatible con una variedad más amplia de solvers, lo que permite resolver problemas más grandes y complejos





PULP vs PYOMO

4. Interacción con Solvers

Pulp: Tiene integración con varios solvers, pero generalmente se utiliza con solvers más simples.

Pyomo: Soporta una gama más amplia de solvers, incluidos los que manejan programación no lineal y problemas más complejos





PULP vs PYOMO

5. Comunidad y Documentación

Ambas bibliotecas tienen comunidades activas y buena documentación, aunque Pyomo tiende a tener una base de usuarios más amplia debido a su mayor versatilidad.



CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com

AENOR
GESTIÓN
DE LA CALIDAD
ISO 9001



PULP vs PYOMO

Conclusión

Si buscas una solución rápida y sencilla para problemas básicos de optimización, Pulp es una buena opción. Si necesitas un enfoque más robusto y flexible para problemas complejos, Pyomo podría ser la mejor elección.





Gracias