

# Proyecto de Modelos de Optimización

## Integrantes:

- Leonardo Amaro
- Alfredo Montero
- Antuán Montes de Oca
- Francisco Vicente Suárez

## Introducción

En este proyecto, presentamos el análisis de problemas de optimización no lineal. Para ello, hemos seleccionado ciertos métodos basándonos en fundamentos sólidos. Nuestra elección se justifica en la eficacia de estos métodos para resolver problemas complejos de optimización.

## Particularidades comunes de los problemas:

- Problemas de Optimización no lineales.
- Las restricciones son lineales.
- Las restricciones son únicamente con respecto al valor máximo y/o mínimo de cada variable.

## Métodos Seleccionados:

### Sequential Least Squares Programming:

El método Sequential Least Squares Programming (SLSQP) es un método de optimización no lineal versátil y eficiente que se puede aplicar a una amplia gama de problemas, tanto con restricciones como sin restricciones. Es una buena opción para problemas que requieren una combinación de eficiencia y versatilidad. Entre sus bondades se encuentran su versatilidad, eficiencia y facilidad de implementación. Sin embargo, también tiene algunas limitaciones, como no ser tan robusto como otros métodos y requerir la evaluación de la función objetivo y sus derivadas en cada paso del algoritmo de optimización. Algunos casos de uso típicos del SLSQP incluyen la optimización de funciones no lineales con restricciones, la resolución de problemas de programación lineal con restricciones no lineales y la optimización de problemas de ingeniería.

### L-BFGS-B :

L-BFGS-B es un método de optimización no lineal que funciona bien en problemas con restricciones simples (del tipo  $l_i \leq x_i \leq u_i$ , siendo  $l_i$  y  $u_i$  límites inferiores y superiores respectivamente). Es una versión modificada del método BFGS que permite incluir restricciones en las variables. Las principales ventajas de este método son que es rápido, eficiente y robusto. El método funciona construyendo una matriz hessiana inversa aproximada, que se utiliza para calcular la dirección de descenso de la función objetivo. La matriz hessiana inversa aproximada se construye utilizando información de la función objetivo y de los puntos de búsqueda anteriores. El proceso se repite hasta que la función objetivo converge a un mínimo. L-BFGS-B es adecuado para problemas no lineales porque no requiere información sobre las derivadas de la función objetivo. También es robusto y eficiente, lo que significa que puede funcionar bien en una amplia gama de problemas.

Se utiliza para realizar una comparación con respecto a SLSQP (dado que nuestros problemas cumplen con sus hipótesis) dado que en este se busca aproximaciones que en teoría deben de disminuir la cantidad de pasos así como los requerimientos de cómputo

### Nelder-Mead:

El método Nelder-Mead es un método de optimización no lineal sin restricciones que utiliza una búsqueda por coordenadas para encontrar el mínimo de una función. Las principales ventajas de este método son que es simple de implementar y puede converger rápidamente a una solución óptima

local. El método funciona construyendo un simplex, que es una figura geométrica con  $N+1$  vértices. El simplex se mueve a través del espacio de búsqueda mediante cuatro operaciones básicas: reflexión, expansión, contracción y shrinking. La reflexión mueve el simplex a través del punto más alejado de la función objetivo. La expansión mueve el simplex aún más lejos de este punto. La contracción mueve el simplex hacia el punto más cercano a la función objetivo. El shrinking contrae uniformemente el simplex. El proceso se repite hasta que el simplex converge a un punto que se considera una solución óptima local. Este método es adecuado para problemas no lineales porque no requiere información sobre las derivadas de la función objetivo; lo cual corresponde con el tipo de problemas que estamos analizando. También es robusto y eficiente, lo que significa que puede funcionar bien en una amplia gama de problemas.

Este es utilizado en este caso para hacer la comparación al realizar una transformación de cada problema a uno sin restricciones.

## Ejercicio 21

### Box-Betts Quadratic Sum Function

Continua, Diferenciable, No Separable, No Escalable, Multimodal

$$f(x) = \sum_{i=1}^n g(x)^2$$

$$g(x) = e^{-0.1*(i+1)x_1} - e^{-0.1*(i+1)x_2} - e^{((-0.1*(i+1))^* - e^{-(i+1)})x_3}$$

sujeto a:

$$0.9 \leq x_1 \leq 1.2, 9 \leq x_2 \leq 11.2, 0.9 \leq x_3 \leq 1.2$$

Vector Esperado :

$$x^* = (1, 10, 1)$$

**Análisis de los resultados:**

#### SLSQP:

Encontró un mínimo que satisface el conjunto de restricciones y numéricamente es el mismo que el esperado (0), las razones por las que difieren los valores de las variables con los esperados pueden ser por errores numéricos de la aritmética al trabajar con valores tan cercanos a cero dado que para valores cercanos a (1, 10, 1) converge rápidamente a la solución óptima. Se mantiene el mismo criterio para todo valor de N. El número de iteraciones se mantuvo entorno a los 13 siendo 14 para  $N = 10$  esta disminución se debe a que por la misma naturaleza de los valores de  $g(x)$  hayan valores que no aporten a la suma por los errores numéricos.

#### L-BFGS-B:

Encontró un mínimo que satisface el conjunto de restricciones y numéricamente es el mismo que el esperado (0), se visualizan los mismos errores numéricos explicados anteriormente, en todo momento la convergencia fue más rápida en el orden de 3-4 pasos y 25% tiempo con respecto al anterior dado que este no se basa en soluciones "exactas" sino que utiliza aproximaciones de la matriz Hessiana. También es mucho más eficiente a nivel de memoria.

#### Nelder-Mead:

Encuentra una soluciones óptimas en varios puntos, los valores de las variables no satisfacen las restricciones. Esto se debe a que no trabajan sobre el conjunto de restricciones. Tiene un muy

elevado número de pasos “cientos” y recursos computacionales, lo cual evidencia que para este tipo de problemas no es conveniente utilizarlo.

## Ejercicio 24

### Brent Function

Continua, Diferenciable, No Separable, No Escalable, Unimodal

$$f(x) = (x_1 + 10)^2 - (x_2 + 10)^2 + e^{-x_1^2 - x_2^2}$$

sujeto a:

$$-10 \leq x_i \leq 10$$

Vector esperado

$$x^* = (-10, -10)$$

Análisis de los resultados:

**SLSQP:**

Encontró un mínimo que satisface el conjunto de restricciones y numéricamente es el mismo que el esperado (-10), el número de iteraciones es de 2 dado que es una función con derivadas sin mayores complicaciones.

**L-BFGS-B:**

En este se encuentran igualmente el óptimo en este caso más rápido como se esperaba por la misma naturaleza del algoritmo al aproximar las derivadas, las cuales no presentan grandes dificultades ni problemas numéricos excepto los ya mencionados.

**Nelder-Mead:**

Logró encontrar el óptimo global, dado que esta es la ruta de descenso posible, el número de iteraciones sigue siendo muy alta 74.

## Ejercicio 25

### Brown Function

Continua, Diferenciable, No Separable, Escalable, Unimodal

$$f(x) = \sum_{i=1}^n (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$$

sujeto a:

$$-1 \leq x_i \leq 4$$

Vector esperado:

$$x^* = (0, 0, \dots, 0)$$

Análisis de los resultados:

**SLSQP:**

Tiene problemas para encontrar el óptimo en todos los casos por ejemplo para  $N = 10$  no devuelve correctamente los valores de los  $x_i$  dado que por ejemplo en  $x_1$  el valor asignado es 3 el cual está muy “alejado” de cero y estos errores se amplifican mientras mayor sea el  $N$  de la sumatoria estos

errores son debido principalmente a los errores del cálculo computacional y la misma función la cual dado su forma exponencial presenta grandes facilidades a los errores en el cálculo de derivadas. En cuanto al número de iteraciones se ve una disminución desde 24 hasta 1 para  $N > 50$  debido a los errores mencionados anteriormente lo cual lleva a que a partir de estos  $N$  no sea fiable los resultados que nos entregan el método.

#### **L-BFGS-B:**

Este algoritmo aunque en un inicio tiene como promedio de valores más fiables que el método anterior, los valores de las variables siguen sin poder cumplir los requisitos de estar en una vecindad de 0, aunque los valores óptimos de la función objetivo eran más pequeños que los anteriores los errores de cálculo están presente desde el inicio, volviendo a recalcar lo no común que es la derivada de esta función que al ser ejecutada tal operación en métodos numéricos aumentan las probabilidades de error, para valores de  $N$  entorno a 270 empieza dar valores de la función objetivo muy alejados del esperado, los errores acumulados por el alto nivel de variables provoca que no sea recomendado en este caso.

#### **Nelder-Mead:**

Aunque no contempla las restricciones este método, para valores de  $N$  entorno a los 10 entrega resultados aceptables numéricamente, aunque esto es solo si se parte desde vecindades del valor buscado, lo que en la práctica no sería viable dado que en caso de no conocer una región inicial no son fiables los resultados. El número de pasos es muy elevado en comparación con los anteriores.

### **Ejercicio 34**

#### **Chung Reynolds Function**

Continua, Diferenciable, Parcialmente Separable, Escalable, Unimodal

$$f(x) = \left( \sum_{i=1}^n (x_i^2) \right)^2$$

**sujeto a:**

$$-100 \leq x_i \leq 100$$

**Vector Esperado :**

$$x^* = (0, 0, \dots, 0)$$

**Análisis de los resultados:**

#### **SLSQP:**

Encontró un mínimo que satisface el conjunto de restricciones y numéricamente es el mismo que el esperado (0), aunque por problemas de la aritmética para valores  $N \geq 100$  se empieza a ver que se aleja del óptimo, siendo así que en con  $N=100$  y  $N=500$  los valores se alejan del óptimo. En cuanto a iteraciones para  $N \leq 100$  se mantiene sobre los 53 - 100 pasos, para valores cercanos a 500 puede verse como desde la primera iteración nos devuelve un resultado, siendo este erróneo por errores aritméticos en el cálculo matricial.

#### **L-BFGS-B:**

En este se encuentran igualmente los óptimos y ocurren los mismos errores aunque en este puede verse los errores a partir de  $N = 100$ . El número de iteraciones se mantiene por debajo del anterior método siendo mayor su tendencia mientras mayor es el  $N$ .

**Nelder-Mead:**

Número de iteraciones en el orden de las miles, no terminó después de 10 minutos de completar para  $N = 500$  Resultados alejados del óptimo desde  $N = 25$ .