FCEIA - Universidad Nacional de Rosario
Optimización Continua - LCC 2025
Profesor: Alejandro G. Marchetti

Guía de Ejercicios No. 6

1) La función de Rosenbrock representa un desafío para muchos algoritmos de optimización, debido a que presenta un mínimo dentro de un valle muy curvado relativamente llano. Encuentre un mínimo de la función de Rosenbrock:

$$f(\mathbf{x}) = (1 - x_1)^2 + 100 \left(x_2 - x_1^2\right)^2, \quad \text{con} \;\; \mathbf{x} \in \mathbb{R}^2.$$

Utilizar la función `minimize` de Scipy:

a) Leer el tutorial de `minimize` en `https://docs.scipy.org/doc/scipy-1.13.0/tutorial`.

b) Seleccionar el método BFGS.

c) Fijar el máximo número de iteraciones en 200, y la tolerancia de la función objetivo, `ftol`, en 10^{-8}. ¿Cómo se define esta tolerancia?.

d) Utilizar el punto inicial $\mathbf{x}_0 = [3 \;\; 2{,}5]^\mathsf{T}$.

e) Resolver (i) suministrando, y (ii) sin suministrar el gradiente de la función objetivo al resolvedor. En cada caso, contar el número de iteraciones y el número de evaluaciones de la función objetivo.

f) Verificar que el algoritmo terminó correctamente.

g) Presentar la solución óptima obtenida y el valor óptimo de la función objetivo.

h) Presentar el script se Python utilizado, con explicaciones comentadas de cada paso.

i) Utilizando `matplotlib`, y la ayuda de `ChatGPT`, crear un gráfico de contorno de la función de Rosenbrock en el dominio $x_1 \in [-2, 2]$, $x_2 \in [-1, 3]$ (emplear una grilla fina de 400×400). Incluir los siguientes niveles de contorno: $[0{,}1, 1, 10, 100, 500, 1000, 2000, 3000, 4000]$. Incluir en el gráfico el punto óptimo obtenido como un punto azul.

2) Considerar el siguiente problema de optimización:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^2} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 + \log(x_1 x_2) \\ \text{s.t.} \quad & 1 - x_1 x_2 \leq 0 \\ & 0{,}1 \leq x_1, \; x_2 \leq 10. \end{aligned}$$

Utilizar la función `minimize` de Scipy:

a) Leer el tutorial de `minimize` en
`https://docs.scipy.org/doc/scipy-1.13.0/tutorial`.

b) Seleccionar el método `SLSQP`. El método SLSQP implementa un método SQP mejorado, con actualización de la matriz hesiana de la función Lagrangiana en cada iteración empleando la fórmula BFGS, y búsqueda lineal usando una función de mérito.

c) Fijar la tolerancia de terminación del punto solución, `xtol`, la tolerancia de la función objetivo, `ftol`, y la tolerancia de las restricciones, `tol`, en 10^{-7}. ¿Cómo se define cada una de estas tolerancias?

d) Utilizar el punto inicial $\mathbf{x}_0 = [2 \ \ 1]^\mathsf{T}$.

e) Proveer a `minimize` las expresiones analíticas de los gradientes de la función objetivo y las restricciones.

f) Verificar que el algoritmo terminó correctamente.

g) Presentar la solución óptima obtenida y el valor óptimo de la función objetivo.

h) ¿El algoritmo SQP convergió a la solución óptima sin quitar la región factible? ¿Cómo lo podría verificar?

i) Presentar el script de Python utilizado, con explicaciones comentadas de cada paso.

3) Resolver el problema de optimización del ejercicio 2, empleando el solver IPOPT disponible en Pyomo. Las instrucciones de instalación de Pyomo y IPOPT se pueden encontrar en el documento `Pyomo-Workshop-December-2023.pdf`. IPOPT implementa un método de punto interior, basado en el uso de funciones barrera.

a) Crear un modelo concreto en Pyomo. Definir las variables, la función objetivo, y las restricciones.

b) Fijar la tolerancia de terminación del punto solución y la tolerancia de la función objetivo en 10^{-7}.

c) Resolver utilizando el solver IPOPT, tomando como punto inicial $\mathbf{x}_0 = [2 \ \ 1]^\mathsf{T}$.

d) Solicitar que IPOPT devuelva el valor de los multiplicadores de Lagrange en la solución óptima.

e) Resolver nuevamente, tomando como punto inicial un punto infactible. Comentar los resultados.

f) Presentar el script de Python utilizado, con explicaciones comentadas de cada paso.

4) Considerar el siguiente problema NLP:

$$\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^3} \quad & f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 - 6x1 - 2x_2 - 12x_3 \\
\text{s.t.} \quad & g_1(\mathbf{x}) = 2x_1^2 + x_2^2 - 15 \leq 0 \\
& g_2(\mathbf{x}) = -x_1 + 2x_2 + x_3 - 3 \leq 0 \\
& x_1,\ x_2,\ x_3 \geq 0.
\end{aligned}$$

a) Hallar una solución óptima $\mathbf{x}^\star$ del problema utilizando un resolvedor adecuado:

- Fijar las tolerancias de terminación del punto solución y de la función objetivo, y la tolerancia de las restricciones en 10^{-7}.
- Utilizar el punto inicial $\mathbf{x}_0 = [1,\ 1,\ 1]^\mathsf{T}$.
- Resolver el NLP aproximando primero los gradientes de la función objetivo y las restricciones mediante diferencias finitas. Luego resolver el problema suministrando al resolvedor las expresiones analíticas del gradiente de la función objetivo y de las restricciones.
- Verificar que el algoritmo terminó correctamente.
- Presentar el código de Python utilizado, la solución óptima obtenida y el valor óptimo de la función objetivo.

b) Repetir la optimización numérica comenzando desde un nuevo punto inicial: $\mathbf{x}_0 = [4,\ 0,\ 0]^\mathsf{T}$. ¿La solución óptima obtenida es la misma que en el caso anterior? Justificar.

c) Obtener los valores de los multiplicadores de Lagrange $\boldsymbol{\nu}^\star$ asociados a la solución óptima $\mathbf{x}^\star$, así como también los valores de los gradientes de la función objetivo y de las restricciones. Verificar que la solución óptima es (i) un punto regular de las restricciones, y (ii) un punto KKT.

d) Considerar el problema perturbado:

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^3} \quad & f(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_2^2 - 6x1 - 2x_2 - 12x_3 \\
\text{s.t.} \quad & g_1(\mathbf{x},\theta) = 2x_1^2 + x_2^2 \leq \theta \\
& g_2(\mathbf{x}) = -x_1 + 2x_2 + x_3 - 3 \leq 0 \\
& x_1,\ x_2,\ x_3 \geq 0,
\end{aligned}
$$

donde θ es el parámetro de perturbación. Resolver el problema perturbado para N valores igualmente espaciados de θ en el intervalo $[0,\ 30]$ (ej., usar una resolución de 0.2 para θ). Denotar el valor óptimo de la función objetivo como $\xi^\star(\theta)$ y el valor óptimo del multiplicador de Lagrange asociado a la restricción perturbada como $\omega^\star(\theta)$.

- Utilizando `matplotlib`, graficar $\xi^\star(\theta)$ versus θ, y estimar la pendiente de la curva en $\theta = 15$. ¿Qué representa dicho valor de la pendiente?
- Utilizando `matplotlib`, graficar $\omega^\star(\theta)$ versus θ. Comentar el gráfico y, en particular, explicar el comportamiento observado en $\theta = 0$.