Practica 1: Introduccio a l'Optimització: Condicions d'optimalitat local i global

Dra. Sundus Zafar

19 de setembre de 2024

1 Introducció

En aquesta sessió de practica, explorarem les condicions d'optimalitat local i global per a problemes d'optimització mitjançant Python. Utilitzarem biblioteques com `numpy` i `scipy` per resoldre diversos problemes d'optimització.

Exemple 1

Considera la funció objectiu quadràtica:

$$f(x) = x^2 + 4x + 4$$

1. Troba el mínim global de la funció.
2. Utilitza Python per verificar els resultats.

Solució

El mínim global es pot trobar analíticament i verifiquem amb Python.

```
import numpy as np
from scipy.optimize import minimize

# Definim la funció objectiu
def func(x):
return x**2 + 4*x + 4

# Trobar el mínim global
result = minimize(func, x0=0)
print("Mínim global en x =", result.x)
print("Valor de la funció en el mínim =", result.fun)
```

Problema 1

Considera la funció objectiu:

$$f(x) = (x - 2)^2$$

amb la restricció:

$$x \geq 1$$

1. Troba el mínim local de la funció subjecte a la restricció.
2. Utilitza Python per comprovar els resultats.

Exemple 2

Considera la funció objectiu:

$$f(x) = x^3 - 3x$$

1. Troba els punts crítics i determina si són mínims locals, màxims locals o punts de sella.

Solució

Utilitzem derivades per trobar els punts crítics i la matriu Hessiana per analitzar la seva naturalesa.

```
[language=python, caption={Codi Python per al Problema 3}]
from sympy import symbols, diff, hessian, solve

# Definim la variable i la funció
x = symbols('x')
f = x**3 - 3*x

# Derivades de primer i segon ordre
df = diff(f, x)
d2f = diff(df, x)

# Trobar punts crítics
critical_points = solve(df, x)

# Calcular la matriu Hessiana
H = hessian(f, x)

# Analitzar punts crítics
for point in critical_points:
hessian_value = H.subs(x, point)
```

```
print(f"Punt crític: {point}")
print(f"Valor de la Hessiana: {hessian_value}")
if hessian_value > 0:
print("És un mínim local.")
elif hessian_value < 0:
print("És un màxim local.")
else:
print("És un punt de sella.")
```

2 Conjunts Convexos

Exemple 3:

Donat un conjunt de punts en el pla $\mathbf{R}^2$, determinar si el conjunt format per aquests punts és convex.

Solució:

Utilitzarem Python per verificar la convexitat d'un conjunt de punts. Utilitzem la biblioteca 'scipy.spatial.ConvexHull' per trobar la convex hull (envoltant convexa) dels punts donats per després dibuixa el conjunt convex.

```
[language=Python, caption=Verificació de la Convexitat d'un Conjunt de Punts]
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial import ConvexHull

# Definir els punts
punts = np.array([[0, 0], [1, 0], [0, 1], [0.5, 0.5]])

# Crear el conjunt convex
hull = ConvexHull(punts)

# Dibuixar els punts i el conjunt convex
plt.plot(punts[:, 0], punts[:, 1], 'o')
for simplex in hull.simplices:
plt.plot(punts[simplex, 0], punts[simplex, 1], 'k-')
plt.fill(punts[hull.vertices, 0], punts[hull.vertices, 1], 'lightgrey', alpha=0.5)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Conjunt Convex')
plt.show()
```

3 Funcions Convexes

Problema 3:

Comprovar la convexitat de la funció $f(x) = x^2$ utilitzant Python.

4 Problemes Convexos

Exemple 4:

Minimitzar la funció $f(x_1, x_2) = x_1^2 + x_2^2$ subjecta a la restricció $x_1 + x_2 \leq 1$.

Solució:

Utilitzarem el mòdul ‘scipy.optimize‘ per resoldre el problema d’optimització convexa.

```
from scipy.optimize import minimize

# Definir la funció objectiu
def func_objectiu(x):
return x[0]**2 + x[1]**2

# Definir la restricció
def restriccio(x):
return 1 - (x[0] + x[1])

# Definir els límits i les restriccions
restriccions = {'type': 'ineq', 'fun': restriccio}
x0 = [0.5, 0.5]  # Punt inicial

# Resoldre el problema d'optimització
sol = minimize(func_objectiu, x0, constraints=restriccions)

# Mostrar el resultat
print('Solució òptima:', sol.x)
print('Valor de la funció objectiu:', sol.fun)
```