Sprawozdanie - Metody numeryczne i optymailzacja

Jakub Andryszczak 259519,
Jakub Żak 244255,
Maciej Cierpisz 249163

Spis treści

1 Zadanie nr. 1

2 Zadanie nr. 2

3 Zadanie nr. 3

4 Zadanie nr. 4

Rozwiąż następujące zadanie programowania kwadratowego:

$$\begin{aligned}
\min \frac{1}{2}x^T Q x + c^T x \\
\text{przy ograniczeniach:} \\
Ax \geqslant b, \\
x \geqslant 0,
\end{aligned}$$

Sformułuj zadanie dualne dla powyższego problem. Następnie rozwiąż poniższe zadanie:

$$\begin{aligned}
\min \quad & \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 6x_2 \\
\text{p.o.} \quad & x_1 + x_2 \leqslant 2, \\
& -x_1 + 2x_2 \leqslant 2, \\
& 2x_1 + x_2^2 \leqslant 2, \\
& x_1 \geqslant 0, \\
& x_2 \geqslant 0.
\end{aligned}$$

Poniżej kod realizujący zadanie:

```
    import cvxpy as cp

# Zdefiniowane zmienne
x1 = cp.Variable()
x2 = cp.Variable()

# Funkcja celu (do zminimalizowania)
objective = cp.Minimize(x1**2 + x2**2 - 4*x1 - 2*x2 - 6)

# Ograniczenia
constraints = [x1 + x2 <= 2, -x1 + 2*x2 <= 2, 2*x1 + x2 <= 2, x1 >= 0,
    x2 >= 0]

# Problem programowania kwadratowego
prob = cp.Problem(objective, constraints)
```

```
# Rozwiązanie problemu
prob.solve()

# śWywietlenie wyników
print("śćWarto minimalna funkcji:", prob.value)
print("x1:", x1.value)
print("x2:", x2.value)
```

Wyniki:

$$\begin{cases} min.funkcji = -9.2 \\ x1 = 0.7999999999999999 \\ x2 = 0.39999999999999997 \end{cases} \tag{1}$$

5 Zadanie nr. 5

Rozwiąż zadanie:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_3^2 \\ \text{p.o.} \quad & x_1 + 2x_2 - x_3 = 4, \\ & x_1 - x_2 + x_3 = -2. \end{aligned}$$

Kod realizujący zadanie:

```
import cvxpy as cp

# Definiowanie zmiennych
x1 = cp.Variable()
x2 = cp.Variable()
x3 = cp.Variable()

# Funkcja celu
objective = cp.Minimize(x1**2 + x2**2 + x3**2)

# Ograniczenia
constraints = [
    x1 + 2*x2 - x3 == 4,
    x1 - x2 + x3 == -2
]

# Definiowanie problemu
problem = cp.Problem(objective, constraints)

# Rozwiązywanie problemu
problem.solve()
```

```
# śWywietlanie wyników
print("Optymalne śwartoci zmiennych:")
print("x1 =", x1.value)
print("x2 =", x2.value)
print("x3 =", x3.value)
print("Minimalna śćwarto funkcji celu =", problem.value)
```

Wyniki:

$$\begin{cases} x1 = 0.28571428571428564 \\ x2 = 1.4285714285714284 \\ x3 = -0.8571428571428571 \end{cases} \tag{2}$$

Minimalna wrtość funkcji celu = 2.8571428571428568.

6 Zadanie nr. 6

Rozwiąż zadanie:

$$\begin{aligned} \min \quad & x_1^2 - x_2^2 \\ \text{p.o.} \quad & x_1 + 2x_2 \geqslant 2, \\ & -5x_1 + 4x_2 \leqslant 10, \\ & x_1 \leqslant 0, \\ & x_2 \geqslant 0. \end{aligned}$$

Kod realizujący zadanie:

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

# Define the objective function
def objective(x):
    return x[0]**2 - x[1]**2

# Define the constraints
def constraint1(x):
    return x[0] + 2*x[1] - 2

def constraint2(x):
    return -5*x[0] + 4*x[1] - 10

# Define the bounds manually since COBYLA doesn't support bounds
    directly
def cobyla_bounds(x):
    return [x[0], -x[0], x[1]]
```

```python
# Define the initial guess
x0 = [-0.1, 1.0] # Feasible initial guess

# Define the constraints dictionary
cons = [{'type': 'ineq', 'fun': constraint1},
        {'type': 'ineq', 'fun': constraint2},
        {'type': 'ineq', 'fun': lambda x: x[0]}, # x1 <= 0 --> x1 is
            negative
        {'type': 'ineq', 'fun': lambda x: x[1]}] # x2 >= 0 --> x2 is
            positive

# Solve the problem
solution = minimize(objective, x0, method='COBYLA', constraints=cons)

# Print the results
print(f"Optimal value: {solution.fun}")
print(f"Optimal x1: {solution.x[0]}")
print(f"Optimal x2: {solution.x[1]}")
```

Wyniki:

$$\begin{cases} x1 = 9.038781217079983 \\ x2 = 974.2149836274926 \end{cases} \tag{3}$$

Optymalna wartość: -949013.1347584255

7 Zadanie nr. 7