Laboratorium 11 - Optymalizacja

Mateusz Podmokły - II rok Informatyka WI

6 czerwiec 2024

1 Treść zadania

Zadanie 1. Wyznacz punkty krytyczne każdej z poniższych funkcji. Scharakteryzuj każdy znaleziony punkt jako minimum, maksimum lub punkt siodłowy. Dla każdej funkcji zbadaj, czy posiada minimum globalne lub maksimum globalne na zbiorze $\mathbb{R}^2$.

$$f_1(x, y) = x^2 - 4xy + y^2$$
$$f_2(x, y) = x^4 - 4xy + y^4$$
$$f_3(x, y) = 2x^3 - 3x^2 - 6xy(x - y - 1)$$
$$f_4(x, y) = (x - y)^4 + x^2 - y^2 - 2x + 2y + 1$$

Zadanie 2. Należy wyznaczyć najkrótszą ścieżkę robota pomiędzy dwoma punktami $x^{(0)}$ i $x^{(n)}$. Problemem są przeszkody usytuowane na trasie robota, których należy unikać. Zadanie polega na minimalizacji funkcja kosztu, która sprowadza problem nieliniowej optymalizacji z ograniczeniami do problemu nieograniczonej optymalizacji.
Macierz $X \in \mathbb{R}^{(n+1)\times 2}$ opisuje ścieżkę złożoną z $n + 1$ punktów $x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(n)}$. Każdy punkt posiada 2 współrzędne, $x^{(i)} \in \mathbb{R}^2$. Punkty początkowy i końcowy ścieżki, $x^{(0)}$ i $x^{(n)}$, są ustalone.
Punkty z przeszkodami (punkty o 2 współrzędnych), $r^{(i)}$ dane są w macierzy przeszkód $R \in \mathbb{R}^{k\times 2}$.
W celu optymalizacji ścieżki robota należy użyć metody największego spadku. Funkcja celu użyta do optymalizacji $F(x^{(0)}, x^{(1)}, \ldots, x^{(n)})$ zdefiniowana jest jako

$$F(x^{(0)}, x^{(1)}, \ldots, x^{(n)}) = \lambda_1 \sum_{i=0}^{n} \sum_{j=1}^{k} \frac{1}{\epsilon + ||x^{(i)} - r^{(j)}||_2^2} + \lambda_2 \sum_{i=0}^{n-1} ||x^{(i+1)} - x^{(i)}||_2^2$$

Symbole użyte we wzorze mają następujące znaczenie:

- Stałe λ_1 i λ_2 określają wpływ każdego członu wyrażenia na wartość $F(X)$

- λ_1 określa wagę składnika zapobiegającego zbytniemu zbliżaniu się do przeszkody,
- λ_2 określa wagę składnika zapobiegającego tworzeniu bardzo długich ścieżek,

- n jest liczbą odcinków, a $n+1$ liczbą punktów na trasie robota,
- k jest liczbą przeszkód, których robot musi unikać,
- dodanie ϵ w mianowniku zapobiega dzieleniu przez zero.

1. Wyprowadź wyrażenie na gradient ∇F funkcji celu F względem $x^{(i)}$:

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x^{(0)}} \\ \vdots \\ \frac{\partial F}{\partial x^{(n)}} \end{bmatrix}$$

   Wzór wyraź poprzez wektory $x^{(i)}$ i ich składowe, wektory $r^{(i)}$ i ich składowe, ϵ, λ_1, λ_2, n, i k (niekoneicznie wszystkie).
   Wskazówka: $\frac{\partial ||x||^2}{\partial x} = 2x$.

2. Opisz matematycznie i zaimplementuj kroki algorytmu największego spadku z przeszukiwaniem liniowym, który służy do minimalizacji funkcji celu F. Do przeszukiwania liniowego (ang. line search) użyj metody złotego podziału (ang. golden section search). W tym celu załóż, że F jest unimodalna (w rzeczywistości tak nie jest) i że można ustalić początkowy przedział, w którym znajduje się minimum.

3. Znajdź nakrótszą ścieżkę robota przy użyciu algorytmu zaimplementowanego w poprzednim punktcie. Przyjmij następujące wartości parametrów:
   - $n = 20$,
   - $k = 50$,
   - $x^{(0)} = [0, 0]$,
   - $x^{(n)} = [20, 20]$,
   - $r^{(i)} \sim \mathcal{U}(0, 20) \times \mathcal{U}(0, 20)$,
   - $\lambda_1 = 1$,
   - $\lambda_2 = 1$,
   - $\epsilon = 10^{-13}$,
   - liczba iteracji = 400.

   Ponieważ nie chcemy zmieniać położenia punktu początkowego i końcowego, $x^{(0)}$ i $x^{(n)}$, wyzeruj gradient funkcji F względem tych punktów.
   Obliczenia przeprowadź dla 5 różnych losowych inicjalizacji punktów wewnątrz ścieżki $x^{(1)}, \ldots, x^{(n-1)}$.
   Narysuj przykładowy wykres wartości funkcji F w zależności od iteracji.

2 Specyfikacja użytego środowiska

Specyfikacja:

- Środowisko: Visual Studio Code,
- Język programowania: Python,
- System operacyjny: Microsoft Windows 11,
- Architektura systemu: x64.

3 Rozwiązanie problemu

3.1 Biblioteki

W realizacji rozwiązania wykorzystane zostały następujące biblioteki:

```
import numpy as np
import matplotlib.pyplot as plt
```

3.2 Zadanie 1.

Punkty krytyczne zostały znalezione przy pomocny metody Newtona. Metoda Newtona wybiera punkt startowy x_0 i wykonuje kolejne iteracje przybliżające rozwiązanie:

$$x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$$

gdzie $\nabla f(x, y)$ to gradient funkcji f dany wzorem

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{bmatrix}$$

a $H(x, y)$ to macierz Hessego dana wzorem

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2}(x,y) & \frac{\partial^2 f}{\partial x \partial y}(x,y) \\ \frac{\partial^2 f}{\partial y \partial x}(x,y) & \frac{\partial^2 f}{\partial y^2}(x,y) \end{bmatrix}$$

Punkt krytyczny jest minimum, jeżeli wszystkie wartości własne macierzy Hessego są większe od 0, maksimum jeżeli są mniejsze od 0, a punktem siodłowym w każdej innej sytuacji. Wartości własne macierzy A można wyznaczyć z równania

$$det(A - \lambda I) = 0$$

gdzie $det()$ to wyznacznik macierzy, a I to macierz jednostkowa. Do obliczenia wartości własnych wykorzystałem funkcję z biblioteki `NumPy`

`np.linalg.eig`

3.3 Zadanie 2.

3.3.1 Gradient funkcji celu

Na początku wyprowadzamy wzór na pochodną cząstkową funkcji celu F.
Dla $i \in [1, n-1]$ mamy

$$\frac{\partial F}{\partial x^{(i)}} = 2\lambda_2(2x^{(i)} - x^{(i-1)} - x^{(i+1)}) - 2\lambda_1 \sum_{j=1}^{k} \frac{x^{(i)} - r^{(j)}}{(\epsilon + ||x^{(i)} - r^{(j)}||^2)^2}$$

dla $i \in \{0, n\}$

$$\frac{\partial F}{\partial x^{(i)}} = (0, 0)$$

ponieważ chcemy, aby punkty x_0 oraz x_n nie zmieniały swojego położenia. Zatem, gradient funkcji F dany jest wzorem

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial x^{(0)}} \\ \vdots \\ \frac{\partial F}{\partial x^{(n)}} \end{bmatrix}$$

3.3.2 Algorytm największego spadku

Algorytm iteracyjny polega na przybliżaniu minimum zadanej funkcji celu F. Polega na wybraniu punktu początkowego x_0 i obliczaniu w kolejnych iteracjach

$$x_{k+1} = x_k + \alpha_k d_k$$

gdzie

$$d_k = -\nabla F(x_k)$$

a α_k to minimum funkcji F wzdłuż kierunku d_k. Możemy je znaleźć metodą złotego podziału (ang. golden section search).

3.3.3 Metoda złotego podziału

Wybiearmy początkowe oszacowanie przedziału $[a, b]$, w naszym przypadku

$$a = 0$$

$$b = 1$$

W każdej iteracji przybliżamy szukane minimum obliczając

$$\alpha_1 = b - \frac{b-a}{\rho}$$

$$\alpha_2 = a + \frac{b-a}{\rho}$$

gdzie

$$\rho = \frac{\sqrt{5}+1}{2}$$

Jeżeli $f(\alpha_1) < f(\alpha_2)$ to

$$b = \alpha_2$$

w przeciwnym wypadku

$$a = \alpha_1$$

Ostateczny wynik to

$$\alpha = \frac{a+b}{2}$$

4 Przedstawienie wyników

4.1 Zadanie 1.

Zestawienie punktów krytycznych funkcji.
Funkcja f_1:

- $(0, 0)$ - punkt siodłowy, $x_0 = (1, 1)$

Funkcja f_2:

- $(-1, -1)$ - minimum, $x_0 = (-2, -2)$
- $(1, 1)$ - minimum, $x_0 = (2, 2)$
- $(0, 0)$ - punkt siodłowy, $x_0 = (-0.5, 0)$

Funkcja f_3:

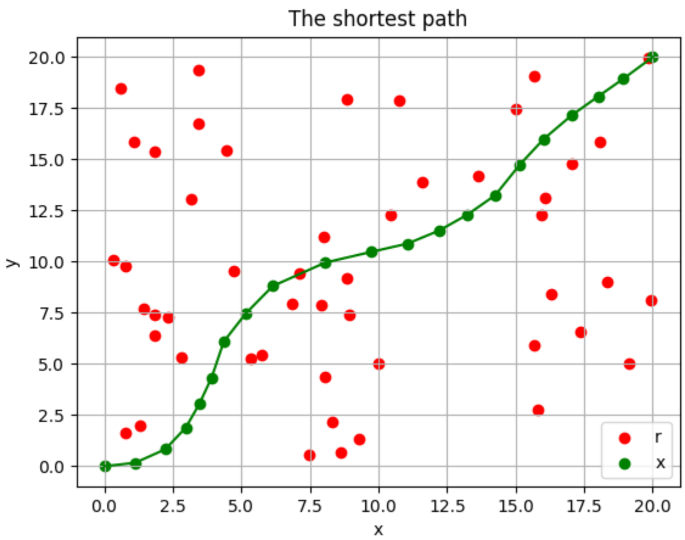
- $(-1, -1)$ - maksimum, $x_0 = (-2, -2)$
- $(1, 0)$ - minimum, $x_0 = (2, 0)$
- $(0, -1)$ - punkt siodłowy, $x_0 = (0, -2)$
- $(0, 0)$ - punkt siodłowy, $x_0 = (0, 1)$

Funkcja f_4:

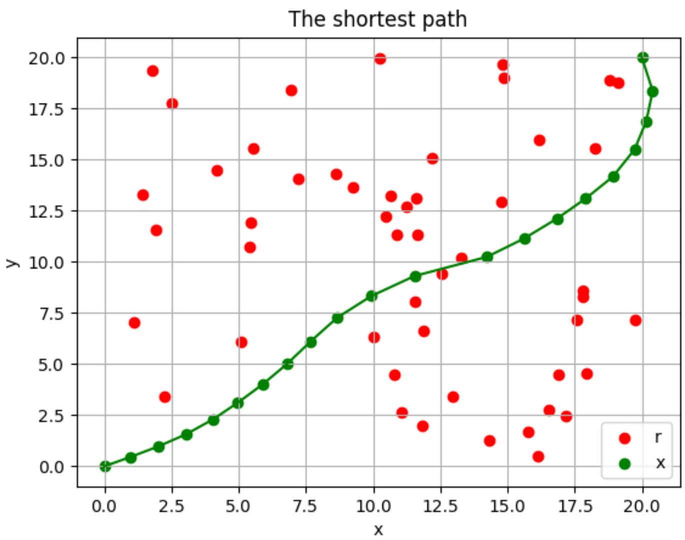
- $(1, 1)$ - punkt siodłowy, $x_0 = (0, 0)$

4.2 Zadanie 2.

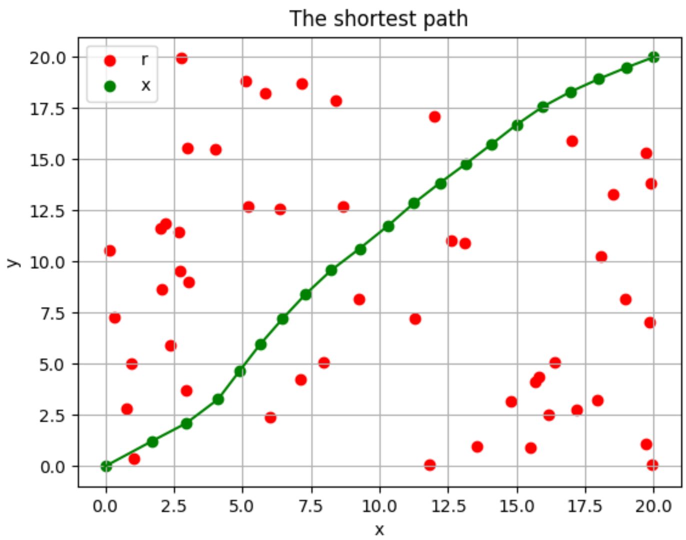
Najkrótsze ścieżki robota dla 5 losowych inicjalizacji punktów początkowych x_i oraz r_i:



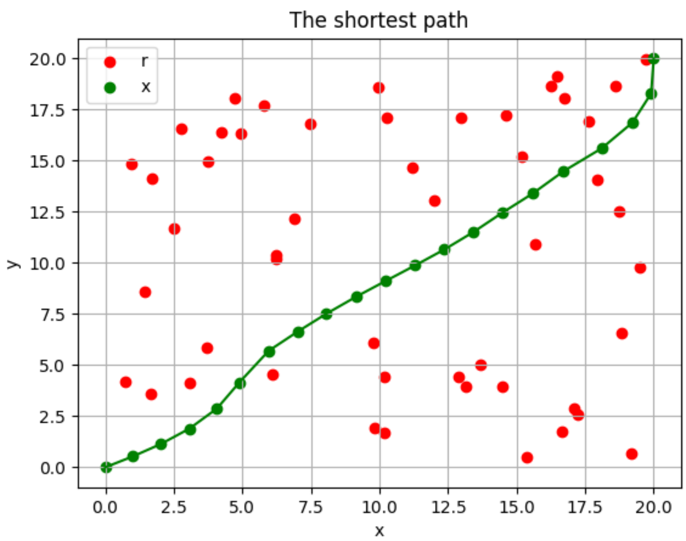
Rysunek 1: Ścieżka 1.



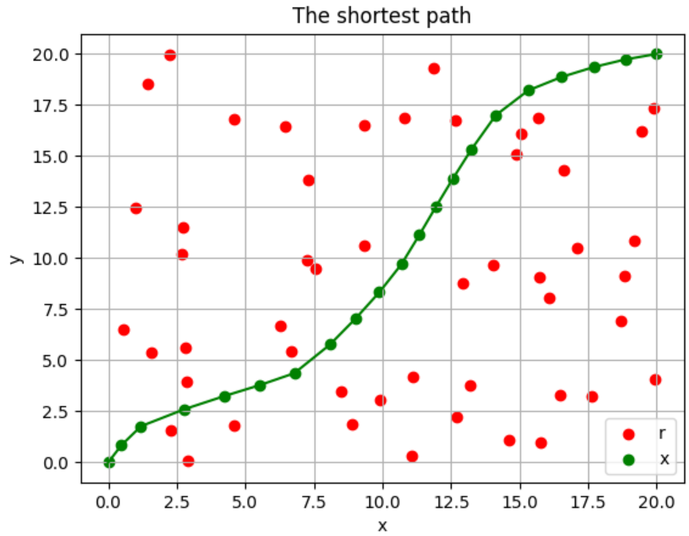
Rysunek 2: Ścieżka 2.



Rysunek 3: Ścieżka 3.

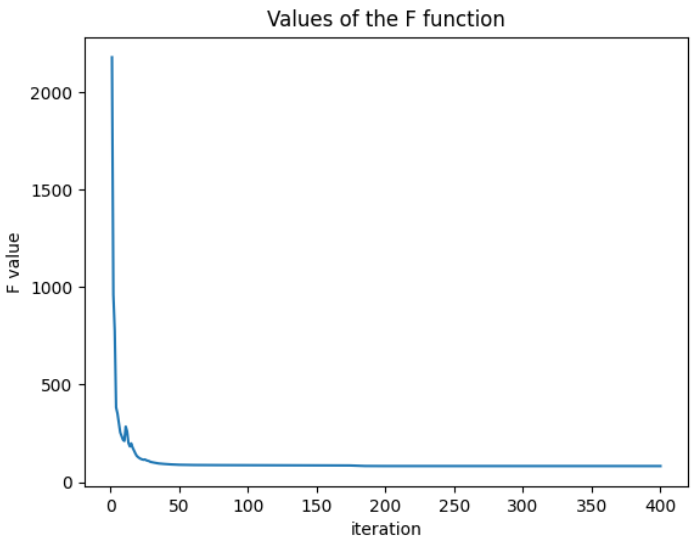


Rysunek 4: Ścieżka 4.



Rysunek 5: Ścieżka 5.

Przykładowy wykres wartości funkcji F:



Rysunek 6: Wartości funkcji F w zależności od iteracji.

5 Wnioski

Metoda Newtona jest skutecznym sposobem znajdowania punktów krytycznych funkcji. Powinniśmy pamiętać o odpowiednim doborze punktów początkowych x_0.

Algorytm największego spadku, z wykorzystaniem metody złotego podziału do znajdowania minimum funkcji jednej zmiennej, poprawnie przybliża najmniejszą wartość funkcji celu w każdej iteracji. Także wymaga doboru punktu początkowego x_0, a dodatkowo, początkowego przedziału $[a, b]$ dla funkcji wzdłuż wektora d.
Optymalizacja funkcji celu jest przydatnym zagadnieniem odpowiadającym na szeroki zakres problemów. Jednak, jak w przypadku wielu algorytmów numerycznych, konieczny jest odpowiedni dobór parametrów.

6 Bibliografia

https://pl.wikipedia.org/wiki/Metoda_Newtona_(optymalizacja)
https://pl.wikipedia.org/wiki/Metoda_najszybszego_spadku
https://pl.wikipedia.org/wiki/Gradient_(matematyka)
https://home.agh.edu.pl/~gora/algebra/Wyklad07.pdf
https://pl.wikipedia.org/wiki/Macierz_Hessego
https://pl.wikipedia.org/wiki/Pochodna_cz%C4%85stkowa
https://pl.wikipedia.org/wiki/Metoda_z%C5%82otego_podzia%C5%82u