

Laboratorium 11

Optymalizacja

Maciej Borowiec

18.06.2025

1 Wprowadzenie

Celem ćwiczenia było wykonanie dwóch zadań związanych z zagadnieniem optymalizacji. Pierwsze porównało spadek wzdłuż gradientu z metodą najmniejszych kwadratów dla układu równań liniowych, a drugie pokazało użycie metody największego spadku dla optymalizacji funkcji nieliniowej.

2 Zadania

2.1 Zadanie 1.

Zadanie polegało na rozwiązaniu zadania z laboratorium drugiego przy użyciu spadku wzdłuż gradientu i porównaniu otrzymanych wyników. Należało zaimplementować funkcję rozwiązującą zagadnienie, używając spadku wzdłuż gradientu ze stałą uczącą wyznaczoną z najmniejszej i największej wartości własnej macierzy $A^T A$ ($\alpha = \frac{1}{\lambda_{min} + \lambda_{max}}$). W poniższej tabeli znajduje się porównanie tych metod.

Metoda	TP	TN	FP	FN	Dokładność	Czas obliczeń
Najmniejszych kwadratów	58	194	6	2	96.923%	0.000169s
Spadku wzdłuż gradientu	38	200	0	22	91.538%	0.010647s

Tabela 1: Porównanie metod dla rozwiązania zagadnienia z laboratorium 2.

Jak widać w danej tabeli (Tabela 1), dla tego problemu lepszą metodą okazała się metoda najmniejszych kwadratów. Dla tej metody dokładność jest zauważalnie większa. Metoda spadku wzdłuż gradientu nie zanotowała żadnych przypadków fałszywie dodatnich, ale za to uzyskała znacznie więcej przypadków fałszywie ujemnych. Czasowo też metoda najmniejszych kwadratów wyszła znacznie lepiej. Jest to zgodne z teorią, gdyż metoda najmniejszych kwadratów ma złożoność czasową równą $O(nm^2 + m^3)$, a dla metody spadku wzdłuż gradientu wynosi ona $k * O(nm)$ (n, m - wymiary macierzy, k - liczba iteracji w metodzie spadku wzdłuż gradientu). Macierz ma wymiary 300×31 , a iteracji

wykonano 5000 (dla mniejszej liczby iteracji dokładność była mniejsza od 90%), dlatego czasy obliczeń są zgodne z teorią.

2.2 Zadanie 2.

W tym zadaniu należało wyznaczyć najkrótszą ścieżkę od punktu startowego do końcowego dla robota, używając metody największego spadku. Mamy dane:

- $x^{(0)}, x^{(1)}, \dots, x^{(n)}$ ($x^{(i)} \in \mathbb{R}^2$) - punkty ścieżki jako argumenty funkcji kosztu, gdzie $x^{(0)}$ i $x^{(n)}$ to niezmiennicze punkty startu i końca ścieżki
- $r^{(1)}, r^{(2)}, \dots, r^{(k)}$ ($r^{(i)} \in \mathbb{R}^2$) - punkty symbolizujące przeszkody

W celu optymalizacji ścieżki robota użyjemy zadanej funkcji kosztu:

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

gdzie:

- λ_1 określa wagę składnika zapobiegającego zbyt niemu zbliżaniu się do przeszkody
- λ_2 określa wagę składnika zapobiegającego tworzeniu bardzo długich ścieżek
- ϵ zapobiega dzieleniu przez 0

2.2.1 Wyznaczenie gradientu ∇F dla funkcji F

Należało wyznaczyć gradient dla funkcji kosztu F . Chcemy, żeby punkty $x^{(0)}$ i $x^{(n)}$ były stałe, zatem względem nich gradient zostanie wyzerowany. Otrzymujemy, więc:

$$\nabla F = [0, \frac{\partial F}{\partial x^{(1)}}, \frac{\partial F}{\partial x^{(2)}}, \dots, \frac{\partial F}{\partial x^{(n-1)}}, 0]$$

Do obliczenia gradientu potrzebujemy zatem obliczyć $\frac{\partial F}{\partial x^{(i)}}$, $i = 1, 2, \dots, n-1$. Zdefiniujemy:

$$S_1 = \sum_{i=0}^n \sum_{j=1}^k \frac{1}{\epsilon + \|x^{(i)} - r^{(j)}\|_2^2} \quad (1)$$

$$S_2 = \sum_{i=0}^{n-1} \|x^{(i+1)} - x^{(i)}\|_2^2 \quad (2)$$

Wtedy:

$$F = \lambda_1 S_1 + \lambda_2 S_2 \implies \frac{\partial F}{\partial x^{(i)}} = \lambda_1 \frac{\partial S_1}{\partial x^{(i)}} + \lambda_2 \frac{\partial S_2}{\partial x^{(i)}}$$

Obliczmy pochodną cząstkową dla równania (1):

$$\frac{\partial S_1}{\partial x^{(i)}} = \frac{\partial}{\partial x^{(i)}} \left(\sum_{l=0}^n \sum_{j=1}^k \frac{1}{\epsilon + \|x^{(l)} - r^{(j)}\|_2^2} \right)$$

Możemy pominąć składniki bez zmiennej $x^{(i)}$:

$$\frac{\partial S_1}{\partial x^{(i)}} = \frac{\partial}{\partial x^{(i)}} \left(\sum_{j=1}^k \frac{1}{\epsilon + \|x^{(i)} - r^{(j)}\|_2^2} \right)$$

$$\frac{\partial S_1}{\partial x^{(i)}} = -2 \sum_{j=1}^k \frac{x^{(i)} - r^{(j)}}{(\epsilon + \|x^{(i)} - r^{(j)}\|_2^2)^2}$$

Obliczmy też pochodną cząstkową dla równania (2):

$$\frac{\partial S_2}{\partial x^{(i)}} = \frac{\partial}{\partial x^{(i)}} \left(\sum_{l=0}^{n-1} \|x^{(l+1)} - x^{(l)}\|_2^2 \right)$$

$$\frac{\partial S_2}{\partial x^{(i)}} = \frac{\partial}{\partial x^{(i)}} (\|x^{(i)} - x^{(i-1)}\|_2^2 + \|x^{(i+1)} - x^{(i)}\|_2^2)$$

$$\frac{\partial S_2}{\partial x^{(i)}} = 2(x^{(i)} - x^{(i-1)}) - 2(x^{(i+1)} - x^{(i)})$$

$$\frac{\partial S_2}{\partial x^{(i)}} = 2(-x^{(i+1)} + 2x^{(i)} - x^{(i-1)})$$

Ostatecznie:

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

2.2.2 Opis algorytmu

Zaimplementowany algorytm wygląda następująco:

1. Inicjalizujemy punkty startowe $x^{(i)}$ (w tym punkt startowy i końcowy) oraz listę do przechowywania wartości funkcji kosztów w danej iteracji. Dodajemy do listy kosztów wartość funkcji kosztu dla punktów początkowych.
2. Obliczamy gradient funkcji dla danych punktów $x^{(i)}$.
3. Znajdujemy długość kroku używając metody złotego środka - w tym celu zakładamy, że funkcja F jest unimodalna (mimo, że w rzeczywistości tak nie jest).
4. Aktualizujemy punkty $x^{(i)}$ zgodnie z długością kroku i gradientem.

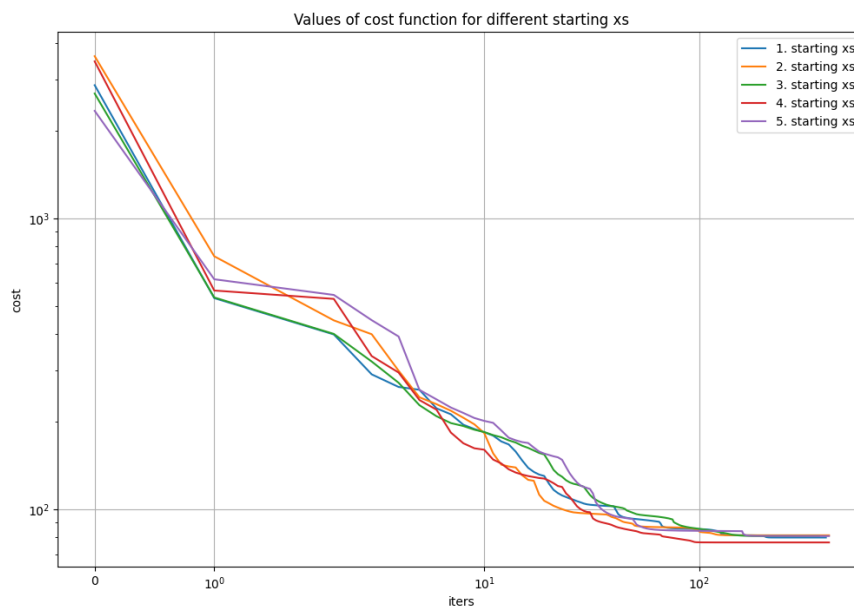
5. Dodajemy do listy kosztów wartość funkcji kosztu dla zaktualizowanych punktów $x^{(i)}$.
6. Wracamy do kroku 2. i powtarzamy czynności aż wartość bezwzględna różnicy kolejnych kosztów jest wystarczająco mała lub dojdziemy do limitu iteracji.

2.2.3 Przedstawienie otrzymanych wyników

Należało znaleźć najkrótszą ścieżkę robota, używając algorytmu z zadanymi parametrami:

- $n = 20, k = 50$
- $x^{(0)} = [0, 0], x^{(n)} = [20, 20]$
- $r^{(i)} \sim \mathcal{U}(0, 20) \times \mathcal{U}(0, 20)$
- $\lambda_1 = \lambda_2 = 1$
- $\varepsilon = 10^{-13}$
- liczba iteracji = 400

Obliczenia należało przeprowadzić dla pięciu różnych, losowych inicjalizacji punktów początkowych (za wyjątkiem punktu startowego i końcowego ścieżki). Poniżej przedstawiony został wykres funkcji kosztu w zależności od danej iteracji oraz tabela zestawiająca koszty początkowe z kosztami końcowymi.



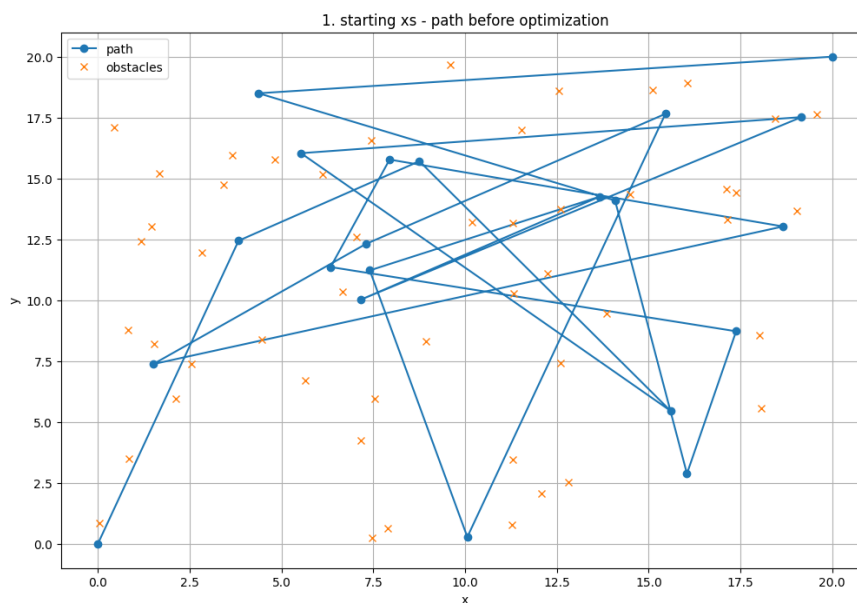
Rysunek 1: Wykres wartości funkcji kosztu w zależności od iteracji

Punkty startowe	1.	2.	3.	4.	5.
Koszt początkowy	2868.114	3607.629	2684.610	3462.098	2340.272
Koszt końcowy	79.815	81.243	80.731	76.775	80.745

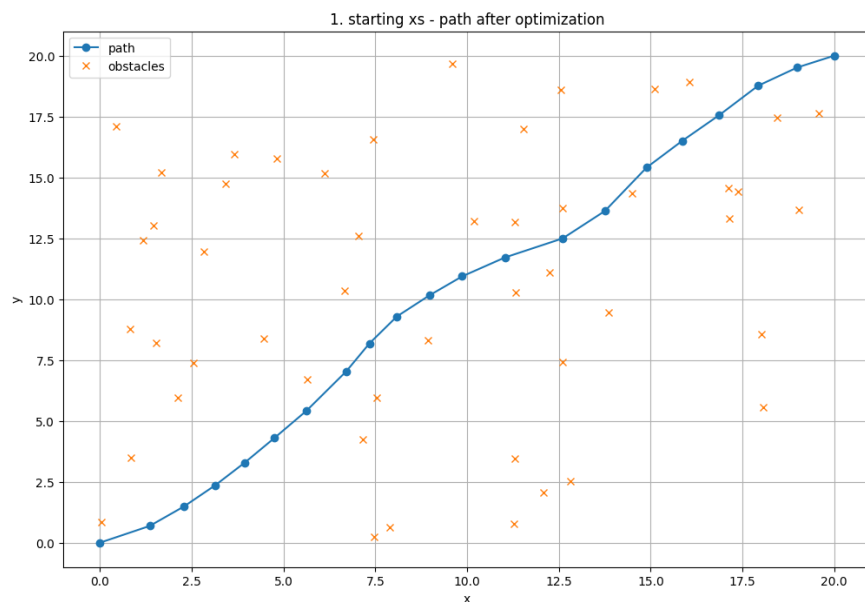
Tabela 2: Zestawienie kosztów początkowych i końcowych dla zadanych zbiorów punktów startowych

Z danego wykresu (Rysunek 1) i tabeli (Tabela 2) widać, że metoda największego spadku dokonała dobrej optymalizacji funkcji zadanej w tym problemie. Udało się zmniejszyć koszt początkowy z kilku tysięcy do kilkudziesięciu dla każdego zbioru punktów startowych. Największa redukcja kosztu nastąpiła dla pierwszego kroku, a dla dalszych było już to zależne od startowych punktów. Można jednak zauważyć, że dla około setnej iteracji koszt zaczął się zmniejszać w bardzo małym stopniu.

Przedstawmy również przykładową ścieżkę przed i po optymalizacji.



Rysunek 2: Przykładowa ścieżka przed optymalizacją



Rysunek 3: Przykładowa ścieżka po optymalizacji

Jak widać z podanych rysunków (Rysunek 2, 3), znaleziona ścieżka, używając optymalizacji, jest znacznie lepsza od startowej, losowej ścieżki.

3 Podsumowanie

Zadania z tego laboratorium pokazały metody rozwiązywania zadań z zagadnienia optymalizacji.

Zadanie 1. pokazało, że nie należy zawsze, bezmyślnie, stosować metod spadku wzdłuż gradientu dla każdego zadania związanego z optymalizacją. Dla prostych układów równań (np. liniowych), szczególnie dla niewielkiej liczby danych i parametrów, lepiej poradzą sobie prostsze metody. W tym przypadku metoda najmniejszych kwadratów okazała się szybsza i dokładniejsza.

Zadanie 2. pokazało zastosowanie gradientu do optymalizacji funkcji nieliniowej. Pokazało, że jeśli jesteśmy w stanie obliczyć gradient funkcji kosztu, to możemy go użyć z dużym sukcesem do optymalizacji. Nie było to jednak bez swoich wad. Funkcja kosztu i jej gradient były na tyle skomplikowane, że wykonanie obliczeń dla wszystkich początkowych zbiorów punktów zajęło w sumie kilka minut.

4 Bibliografia

- Materiały zamieszczone wraz z zadaniem