

MOWNiT Lab11

Radosław Szepielak, Patryk Blacha

18 czerwca 2025

1 Zadanie1

1.1 Wprowadzenie

Celem zadania było rozwiązanie problemu klasyfikacji binarnej typu nowotworu (złośliwy/łagodny) z wykorzystaniem metody spadku wzdłuż gradientu. Otrzymane wyniki zostały porównane z rozwiązaniem uzyskanym za pomocą analitycznej metody najmniejszych kwadratów (równania normalne) pod kątem dokładności predykcji, czasu obliczeń oraz teoretycznej złożoności obliczeniowej.

1.2 Metodyka

Problem predykcji potraktowano jako zadanie regresji liniowej, gdzie celem jest znalezienie wektora wag \mathbf{w} , który minimalizuje błąd kwadratowy między predykcjami modelu a rzeczywistymi etykietami.

1.2.1 Metoda najmniejszych kwadratów (równania normalne)

Metoda ta polega na analitycznym wyznaczeniu wektora wag \mathbf{w} poprzez rozwiązanie równania normalnego:

$$\mathbf{A}^T \mathbf{A} \mathbf{w} = \mathbf{A}^T \mathbf{b}$$

gdzie \mathbf{A} to macierz cech (z dołączoną kolumną jedynek dla wyrazu wolnego), a \mathbf{b} to wektor etykiet (1 dla nowotworu złośliwego, -1 dla łagodnego). Rozwiązanie można zapisać jako:

$$\mathbf{w} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

W implementacji do rozwiązania układu równań wykorzystano funkcję `numpy.linalg.solve`, która jest numerycznie stabilniejsza niż bezpośrednie obliczanie macierzy odwrotnej.

1.2.2 Metoda spadku wzdłuż gradientu (Gradient Descent)

Jest to metoda iteracyjna, która minimalizuje funkcję kosztu $L(\mathbf{w})$ poprzez przesuwanie wektora wag \mathbf{w} w kierunku przeciwnym do gradientu. Funkcja kosztu to błąd średniokwadratowy:

$$L(\mathbf{w}) = \frac{1}{n} \|\mathbf{A} \mathbf{w} - \mathbf{b}\|_2^2$$

Gradient tej funkcji względem wag wynosi:

$$\nabla_{\mathbf{w}}L(\mathbf{w}) = \frac{2}{n}\mathbf{A}^T(\mathbf{A}\mathbf{w} - \mathbf{b})$$

Krok aktualizacji wag w każdej iteracji k jest dany wzorem:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \nabla_{\mathbf{w}}L(\mathbf{w}_k)$$

Stała ucząca α została wyznaczona tak, aby zapewnić zbieżność algorytmu, tj. spełniając warunek $0 < \alpha < \frac{2}{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}$, gdzie λ_{\max} to największa wartość własna macierzy $\mathbf{A}^T\mathbf{A}$. W eksperymencie przyjęto 1000 iteracji.

1.3 Wyniki

Obie metody zostały zaimplementowane i przetestowane na tym samym zbiorze danych. Poniżej przedstawiono porównanie uzyskanych rezultatów.

1.3.1 Porównanie numeryczne i złożonościowe

W tabeli 1 zestawiono dokładność predykcji na zbiorze walidacyjnym oraz czas obliczeń dla obu metod.

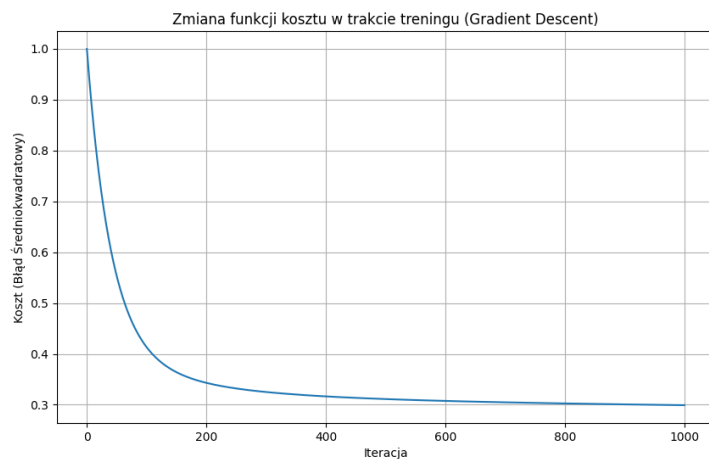
Tabela 1: Porównanie wyników metody najmniejszych kwadratów i spadku wzdłuż gradientu.

Kryterium	Metoda Najmniejszych Kwadratów	Spadek Wzdłuż Gradientu
Dokładność na zbiorze testowym	88.08%	90.38%
Czas obliczeń	0.0010 s	0.0199 s
Złożoność (n - próbki, m - cechy)	$O(n \cdot m^2 + m^3)$	$O(k \cdot n \cdot m)$

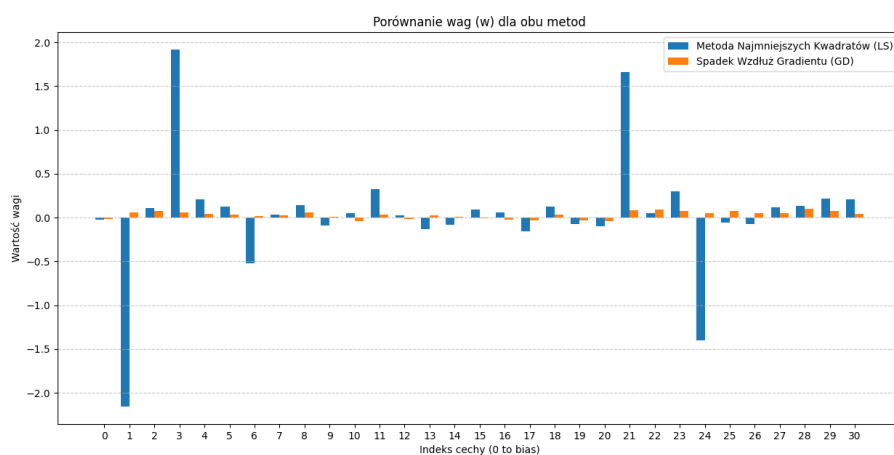
Jak widać, dla analizowanego problemu metoda spadku wzdłuż gradientu uzyskała nieznacznie wyższą dokładność (90.38% vs 88.08%), jednak jej czas wykonania był około 20-krotnie dłuższy. Jest to zgodne z analizą złożoności – dla problemu o małej liczbie cech ($m \approx 30$) koszt odwrócenia macierzy w równaniach normalnych jest stosunkowo niski.

1.3.2 Analiza graficzna

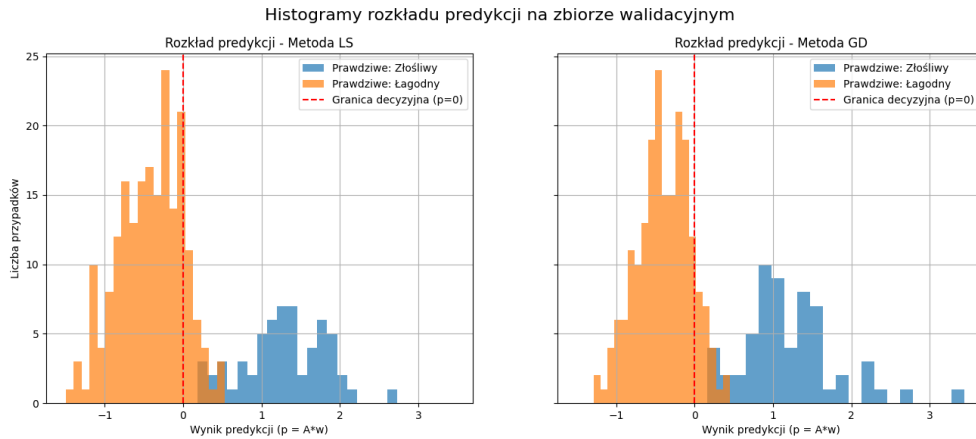
W celu głębszej analizy działania algorytmów wygenerowano poniższe wykresy.



Rysunek 1: Wykres funkcji kosztu (błędu średniokwadratowego) w kolejnych iteracjach metody spadku wzdłuż gradientu. Widoczny jest szybki spadek wartości błędu, co świadczy o prawidłowej zbieżności algorytmu.



Rysunek 2: Porównanie wartości wag wyznaczonych przez obie metody. Mimo że wektory wag różnią się w poszczególnych punktach, obie metody osiągają podobne minima funkcji kosztu. Różnice mogą wynikać z ograniczonej liczby iteracji w metodzie gradientowej, doboru współczynnika uczenia, czy też różnic numerycznych między metodami.



Rysunek 3: Histogramy rozkładu predykcji na zbiorze walidacyjnym. Lepsza separacja rozkładów dla klas złośliwych i łagodnych w metodzie spadku gradientu (prawy wykres) jest spójna z zaobserwowaną wyższą dokładnością tej metody.

1.4 Wnioski

Przeprowadzony eksperyment dostarczył szeregu kluczowych spostrzeżeń dotyczących praktycznych różnic między analitycznym podejściem bazującym na równaniach normalnych a iteracyjną metodą spadku wzdłuż gradientu.

- **Analiza dokładności i stabilności numerycznej:** Metoda spadku wzdłuż gradientu, mimo swojej prostoty, osiągnęła nieznacznie wyższą dokładność (90.38%) niż metoda najmniejszych kwadratów (88.08%). Można to przypisać kilku czynnikom. Po pierwsze, normalizacja cech, która jest standardową praktyką przy stosowaniu metod gradientowych, poprawia "geometrię" funkcji kosztu, czyniąc jej powierzchnię bardziej regularną. Ułatwia to algorytmowi znalezienie dobrego minimum. Po drugie, metody iteracyjne bywają bardziej odporne na złe uwarunkowanie macierzy $\mathbf{A}^T \mathbf{A}$. Analityczne odwracanie macierzy (lub rozwiązywanie układu równań) jest wrażliwe na wysoki współczynnik uwarunkowania, co może prowadzić do niestabilności numerycznej i błędów w wyznaczonych wagach. Metoda spadku gradientu, wykonując małe kroki, może w takich sytuacjach znaleźć bardziej "stabilne" rozwiązanie, które lepiej generalizuje się na danych testowych.
- **Wydajność obliczeniowa i skalowalność:** Różnica w czasie obliczeń jest drastyczna i w pełni zgodna z teorią. Dla analizowanego problemu, gdzie liczba próbek uczących $n \approx 400$ a liczba cech $m = 31$, złożoność metody najmniejszych kwadratów ($O(n \cdot m^2 + m^3)$) okazała się znacznie niższa niż koszt 1000 iteracji metody gradientowej ($O(k \cdot n \cdot m)$). Metoda analityczna jest operacją "jednostrzałą", podczas gdy koszt metody iteracyjnej jest wprost proporcjonalny do liczby kroków k .
- **Praktyczne implikacje i wszechstronność:** Prawdziwa siła metody spadku wzdłuż gradientu ujawnia się w scenariuszach, gdzie metoda analityczna zawodzi lub jest niewykonalna.
 - **Duża liczba cech:** Gdyby liczba cech m sięgała dziesiątek tysięcy (co jest normą np. w przetwarzaniu tekstu), koszt pamięciowy i obliczeniowy związany z macierzą $\mathbf{A}^T \mathbf{A}$ (o wymiarach $m \times m$) oraz jej odwróceniem ($O(m^3)$) byłby zaporowy. Metoda gradientowa, z kosztem $O(n \cdot m)$ na iterację, staje się jedynym praktycznym wyborem.

- **Uczenie online i duże zbiory danych:** Metoda gradientowa posiada warianty (np. stochastyczny spadek gradientu, SGD), które pozwalają na aktualizację wag na podstawie pojedynczych próbek lub małych paczek danych. Umożliwia to trenowanie modeli na zbiorach, które nie mieszczą się w pamięci RAM.
- **Modele nieliniowe:** Metoda spadku wzdłuż gradientu jest uniwersalnym algorytmem optymalizacyjnym, stanowiącym fundament uczenia sieci neuronowych i innych złożonych modeli, dla których analityczne rozwiązania nie istnieją.

Podsumowując, chociaż dla tego konkretnego problemu o niewielkiej skali metoda najmniejszych kwadratów była znacznie wydajniejsza, eksperyment ten doskonale ilustruje fundamentalne kompromisy. Zasady działania i zalety metody spadku wzdłuż gradientu – skalowalność, elastyczność i wszechstronność – czynią ją jednym z najważniejszych narzędzi we współczesnym uczeniu maszynowym, niezbędnym do rozwiązywania problemów na dużą skalę.

2 Zadanie 2

2.1 Wprowadzenie

Celem naszego projektu było wyznaczenie optymalnej, bezkolizyjnej ścieżki dla robota poruszającego się w dwuwymiarowej przestrzeni z przeszkodami. Zadanie to sprowadza się do problemu optymalizacji nieliniowej, który rozwiązaliśmy poprzez minimalizację specjalnie zdefiniowanej funkcji kosztu. Funkcja ta uwzględnia dwa kluczowe aspekty: unikanie przeszkód oraz minimalizację długości samej ścieżki.

2.2 Definicja Problemu

Zdefiniowaliśmy ścieżkę jako sekwencję $n + 1$ punktów $x^{(i)} \in \mathbb{R}^2$, reprezentowaną przez macierz $X \in \mathbb{R}^{(n+1) \times 2}$. Punkty początkowy $x^{(0)}$ i końcowy $x^{(n)}$ były ustalone, natomiast pozycje k przeszkód, reprezentowane przez wektory $r^{(j)} \in \mathbb{R}^2$, były znane.

2.3 Funkcja Celu

Naszym zadaniem była minimalizacja następującej funkcji celu $F(X)$:

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

gdzie parametry λ_1 i λ_2 to wagi określające kompromis między bezpieczeństwem (unikaniem przeszkód) a efektywnością (długością ścieżki), a ϵ to mała stała zapobiegająca dzieleniu przez zero.

3 Metodyka

W celu rozwiązania problemu optymalizacyjnego zaimplementowaliśmy algorytm największego spadku (gradientu prostego) z dokładnym przeszukiwaniem liniowym.

3.1 Algorytm Optymalizacyjny

Algorytm największego spadku jest iteracyjną metodą optymalizacji, która w każdym kroku przemieszcza punkty ścieżki w kierunku przeciwnym do gradientu funkcji celu:

$$X_{k+1} = X_k - \alpha_k \nabla F(X_k) \quad (2)$$

gdzie α_k to długość kroku w k -tej iteracji. Kluczowym elementem naszej implementacji było wyznaczenie gradientu $\nabla F(X)$ oraz dobór optymalnej długości kroku α_k .

3.2 Przeszukiwanie Liniowe

Do znalezienia optymalnej wartości α_k w każdej iteracji użyliśmy **metody złotego podziału**. Jest to technika minimalizacji funkcji jednej zmiennej, która nie wymaga obliczania jej pochodnej i gwarantuje znalezienie minimum w zadanym przedziale dla funkcji unimodalnej.

3.3 Parametry Eksperymentu

Przeprowadziliśmy symulacje, przyjmując parametry zgodne z treścią zadania, które zebraliśmy w Tabeli 2. W celu zbadania wrażliwości metody na warunki początkowe, przeprowadziliśmy 5 prób z różnymi losowymi inicjalizacjami wewnętrznych punktów ścieżki.

Tabela 2: Parametry konfiguracyjne eksperymentu.

Parametr	Wartość
Liczba segmentów ścieżki (n)	20
Liczba punktów ścieżki ($n + 1$)	21
Liczba przeszkód (k)	50
Punkt startowy $x^{(0)}$	[0, 0]
Punkt końcowy $x^{(n)}$	[20, 20]
Waga unikania przeszkód (λ_1)	1
Waga długości ścieżki (λ_2)	1
Parametr regularyzacji (ϵ)	10^{-13}
Maksymalna liczba iteracji	400
Liczba prób	5

4 Wyniki i Dyskusja

4.1 Analiza Ilościowa

W Tabeli 3 przedstawiliśmy zbiorcze wyniki dla wszystkich 5 prób. Zaobserwowaliśmy, że w zależności od losowej inicjalizacji, algorytm zbiegał do różnych minimów lokalnych, co jest typowe dla problemów niekonweksyjnych.

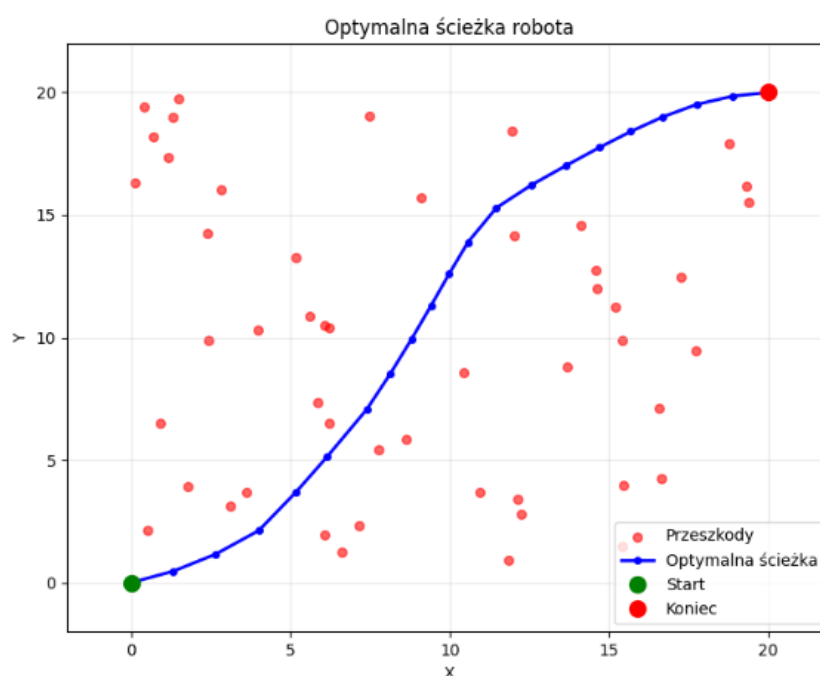
Najlepsze rozwiązanie uzyskaliśmy w **Próbie 5**, gdzie wartość funkcji celu osiągnęła najniższy poziom **72.6652**. Długość tej optymalnej ścieżki wyniosła **29.651** jednostek, a minimalna odległość do najbliższej przeszkody wyniosła **1.258**, co świadczy o wygenerowaniu bezpiecznej trajektorii.

Tabela 3: Wyniki końcowe dla 5 prób optymalizacji.

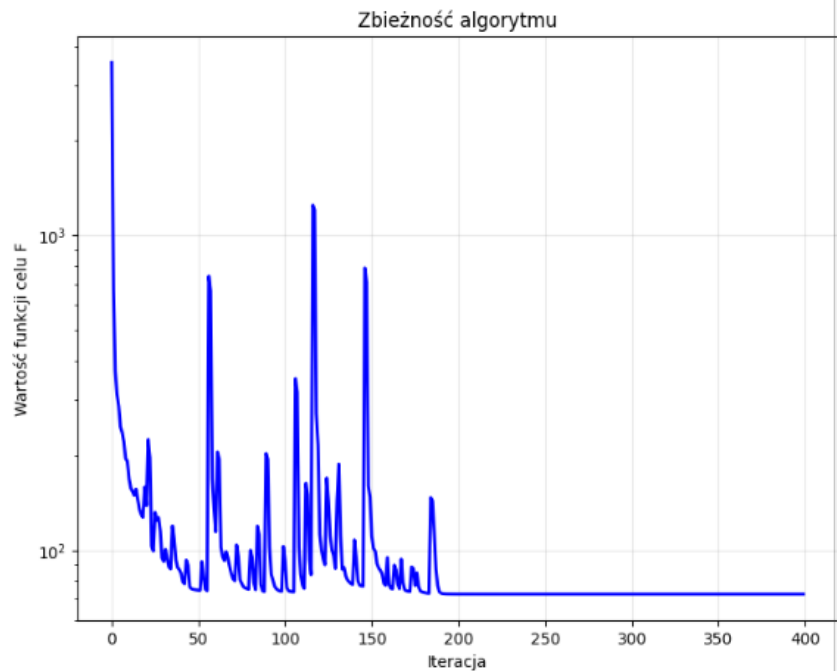
Nr Próby	Końcowa wartość $F(X)$	Norma gradientu $\ \nabla F\ $ w iteracji 400
1	73.5098	0.0020
2	94.2141	2.8978
3	75.2762	3.0737
4	76.9744	0.000035
5 (Najlepsza)	72.6652	0.0005
Średnia	78.5279	-
Odch. std.	± 7.9825	-

4.2 Analiza Graficzna

W celu wizualizacji wyników przygotowaliśmy cztery wykresy, które odpowiadają rezultatom z najlepszej próby (Próba 5).

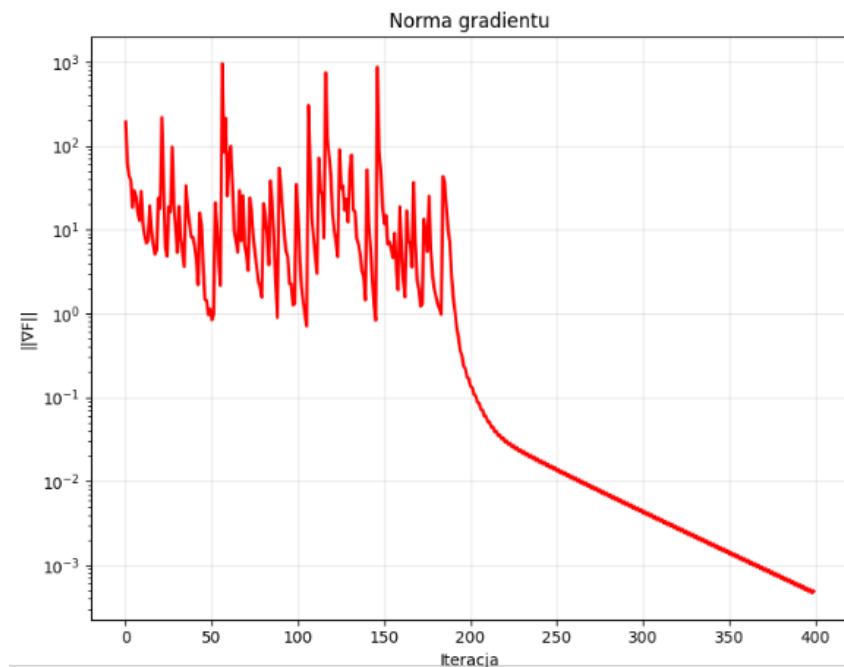


Rysunek 4: Wizualizacja optymalnej ścieżki (kolor niebieski) na tle losowo rozmieszczonych przeszkód (kolor czerwony). Zielony i czerwony punkt oznaczają odpowiednio start i metę.



Rysunek 5: Zmiana wartości funkcji celu $F(X)$ w zależności od numeru iteracji. Skala logarytmiczna na osi Y podkreśla gwałtowny spadek wartości w początkowych iteracjach.

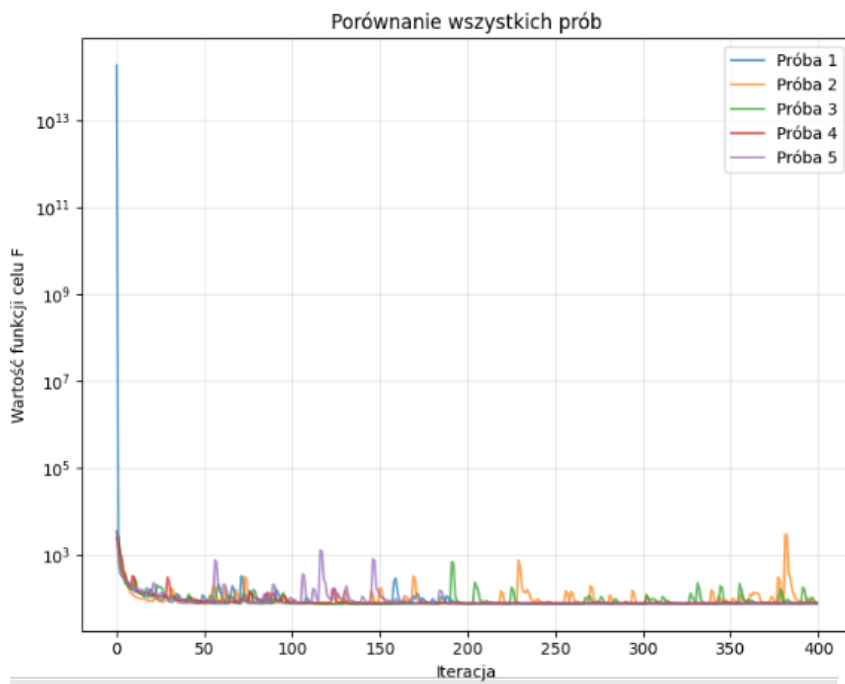
Wykres na rysunku 5 ilustruje typową dla metod pierwszego rzędu dynamikę zbieżności. W początkowej fazie (ok. 50 iteracji) obserwowaliśmy gwałtowny spadek wartości funkcji celu, po czym tempo zbieżności znacznie malało, a algorytm dokonywał już tylko niewielkich korekt.



Rysunek 6: Zmiana normy euklidesowej gradientu $\|\nabla F\|$ w kolejnych iteracjach (skala logarytmiczna).

Norma gradientu (Rysunek 6) również gwałtownie spadała na początku, co świadczy o oddalaniu

się od złych regionów przestrzeni poszukiwań. W żadnej z prób nie osiągnęliśmy zadanego progu tolerancji 10^{-8} , co wskazuje na powolną zbieżność w końcowej fazie.



Rysunek 7: Porównanie przebiegu wartości funkcji celu dla wszystkich 5 prób. Różnice w przebiegach i wartościach końcowych potwierdzają wrażliwość algorytmu na inicjalizację.

Rysunek 7 jest kluczowy dla zrozumienia natury problemu. Jasno pokazuje on, że różne losowe punkty startowe prowadzą do zbieżności do różnych minimów lokalnych, co jest bezpośrednim skutkiem niekonweksowości funkcji celu.

5 Wnioski Końcowe

Przeprowadzone badanie optymalizacji ścieżki robota wykazało zarówno skuteczność zastosowanej metody, jak i jej inherentne ograniczenia.

Zaimplementowany algorytm największego spadku skutecznie znajdował **bezkolizyjne i relatywnie krótkie ścieżki** dla robota w przestrzeni z przeszkodami. Minimalne odległości do przeszkód (np. 1.258 jednostek) i długości ścieżek (29.651 jednostek) w najlepszych próbach potwierdzają efektywność funkcji celu w równoważeniu unikania kolizji z minimalizacją długości trajektorii. Wizualizacje wyraźnie pokazywały płynne omijanie przeszkód.

Jednym z kluczowych wniosków jest **silna wrażliwość algorytmu na początkowe warunki losowe**. Różne inicjalizacje prowadziły do zbieżności do **różnych minimów lokalnych**, co jest typowe dla problemów niekonweksyjnych. Z tego powodu, aby zwiększyć szansę na znalezienie dobrego rozwiązania, **konieczne jest wielokrotne uruchamianie algorytmu** z różnymi punktami startowymi, a następnie wybór najlepszego z uzyskanych wyników.

Mimo początkowego szybkiego spadku wartości funkcji celu i normy gradientu, obserwowaliśmy **powolną zbieżność w końcowej fazie optymalizacji**. W żadnej z prób nie osiągnięto zadanego progu tolerancji dla normy gradientu w ramach dostępnych iteracji, co jest charak-

terystyczną cechą algorytmu największego spadku w pobliżu minimum. Oznacza to, że metoda wymaga dużej liczby iteracji do osiągnięcia wysokiej precyzji.

5.1 Potencjalne Ulepszenia

Aby przyspieszyć zbieżność i potencjalnie znaleźć lepsze rozwiązania, sugeruje się rozważenie bardziej zaawansowanych metod optymalizacji. Przyszłe prace mogłyby obejmować:

- **Metody quasi-Newtonowskie** (np. BFGS, L-BFGS), które wykorzystują przybliżenia hesjanu dla efektywniejszego kierunku spadku.
- **Metody gradientu sprzężonego**, które poprawiają zbieżność poprzez wykorzystanie informacji z poprzednich kroków.
- **Zaawansowane techniki inicjalizacji**, aby generować bardziej obiecujące punkty startowe.

Te modyfikacje mogłyby znacząco poprawić wydajność i robustność algorytmu w praktycznych zastosowaniach.