Data: 04.06.2025

Laboratorium 11

Autorzy: Kosman Mateusz, Ludwin Bartosz

Temat laboratorium:

Celem laboratorium jest analiza efektywności metody spadku wzdłuż gradientu na przykładzie predykcji typu nowotworu oraz optymalizacja nieliniowej funkcji celu dla wyznaczenia najkrótszej ścieżki robota w środowisku z przeszkodami. W pierwszej części porównamy dokładność i złożoność obliczeniową rozwiązania opartego na gradient descent z klasycznymi metodami najmniejszych kwadratów, a w drugiej zastosujemy największy spadek z przeszukiwaniem liniowym do minimalizacji funkcji kosztu opisującej dystans robota i unikanie przeszkód.

Treść:

Zadanie 1. Rozwiąż ponownie problem predykcji typu nowotworu (laboratorium 2), używając metody spadku wzdłuż gradientu (ang. gradient descent). Stałą uczącą możesz wyznaczyć na podstawie najmniejszej i największej wartości własnej macierzy $A^T A$. Porównaj uzyskane rozwiązanie z metodą najmniejszych kwadratów, biorąc pod uwagę następujące kryteria:

- Dokładność predykcji na zbiorze testowym
- Teoretyczną złożoność obliczeniową
- Czas obliczeń.

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 $X \in R^{(n+1)\times 2}$ opisuje ścieżkę złożoną z $n+1$ punktów $x^{(0)}, x^{(1)}, x^{(2)}, \ldots, x^{(n)}$. Każdy punkt posiada 2 współrzędne, x(i) $\in R^2$. 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 $\in R^{k\times 2}$.
W celu optymalizacji ścieżki robota należy użyć metody największego spadku. Funkcja celu użyta do optymalizacji F(x(0),x(1),...,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 λ1 i λ2 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 $x^{(i)}$:

  $$\nabla F = [\frac{\partial F}{\partial x(0)}, \ldots, \frac{\partial F}{\partial x(n)}]$$

  Wzór wyraź poprzez wektory x^{i} i ich składowe, wektory $r^{(j)}$ i ich składowe, ε,λ_1, λ_2, n i k (niekoniecznie wszystkie).

  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 najkrótszą ścieżkę robota przy użyciu algorytmu zaimplementowanego w poprzednim punkcie. Przyjmij następujące wartości parametrów:

  - n = 20,k = 50
  - x(0) = [0,0], x^{n} = [20,20]
  - r(i) ~ U(0,20) ×U(0,20)
  - λ_1 = λ_2 = 1
  - ε = 10^{-13}
  - liczba iteracji = 400

  Ponieważ nie chcemy zmieniać położenia punktu początkowego i końcowego, x(0),x(n), wyzeruj gradient funkcji F względem tym punktów.
  Obliczenia przeprowadź dla 5 różnych losowych inicjalizacji punktów wewnątrz ścieżki x(1),...,x(n−1).

Narysuj przykładowy wykres wartości funkcji F w zależności od iteracji. Zapewnij powtarzalność wyników, ustawiając wartość odpowiedniego ziarna.

Rozwiązanie:

Zadanie 1:

Po zaimplementowaniu metody spadku wzdłuż gradientu oraz użyciu kodu z laboratorium 2 otrzymaliśmy takie rezultaty:

Metoda	Dokładność	Czas [s]
Gradient Descent	0.976923	0.013650
LS (Normal Eq - Linear)	0.973077	0.000008
LS (Normal Eq - Quadratic)	0.926923	0.000004
LS (SVD)	0.973077	0.000023
LS (Regularized SVD)	0.973077	0.000020

Tabela 1

a obliczona stała uczenia wyniosła: 0.000512.

Widać że wszystkie modele poradziły sobie bardzo dobrze (każdy osiągnął dokładność powyżej 90%), natomiast najlepiej z nich poradziła sobie metoda spadku wzdłuż gradientu. Okupowane jest to jednak czasem wykonania algorytmu, który był rzędy wielkości większy od metod najmniejszych kwadratów.

Teoretyczne złożoności obliczeniowe kształtują się natomiast następująco:

Metoda	Gradient Descent	LS (Normal Eq Linear)	LS (Normal Eq Quadratic)	LS (SVD)	LS (Regularized SVD)
Teoretyczna złożoność	$O(k \cdot n \cdot p)$	$O(n \cdot p^2 + p^3)$	$O(n \cdot q^2 + q^3)$	$O(\min(n^2 \cdot p, n \cdot p^2))$	$O(\min(n^2 \cdot p, n \cdot p^2))$

Tabela 2

gdzie k to liczba iteracji, n - liczba próbek, p - liczba cech, q - liczba cech w macierzy kwadratowej.

Gradient Descent wypada najwolniej ze względu na iteracyjny charakter i konieczność wielu przejść przez dane, ale dobrze radzi sobie przy dużej liczbie próbek i umiarkowanej liczbie cech. W przeciwieństwie do tego, metody Least Squares (LS) są znacznie szybsze, choć ich koszt rośnie przy dużej liczbie cech. LS z równaniami normalnymi działa efektywnie, gdy liczba cech jest umiarkowana (szczególnie wersja liniowa), ale wariant kwadratowy generuje dużo dodatkowych cech, co znacząco zwiększa złożoność. Metody oparte na SVD są najbardziej kosztowne obliczeniowo, ale oferują lepszą stabilność numeryczną. W praktyce LS (liniowe) są najszybsze, GD najwolniejszy, a SVD najbezpieczniejszy numerycznie.

Zadanie 2:

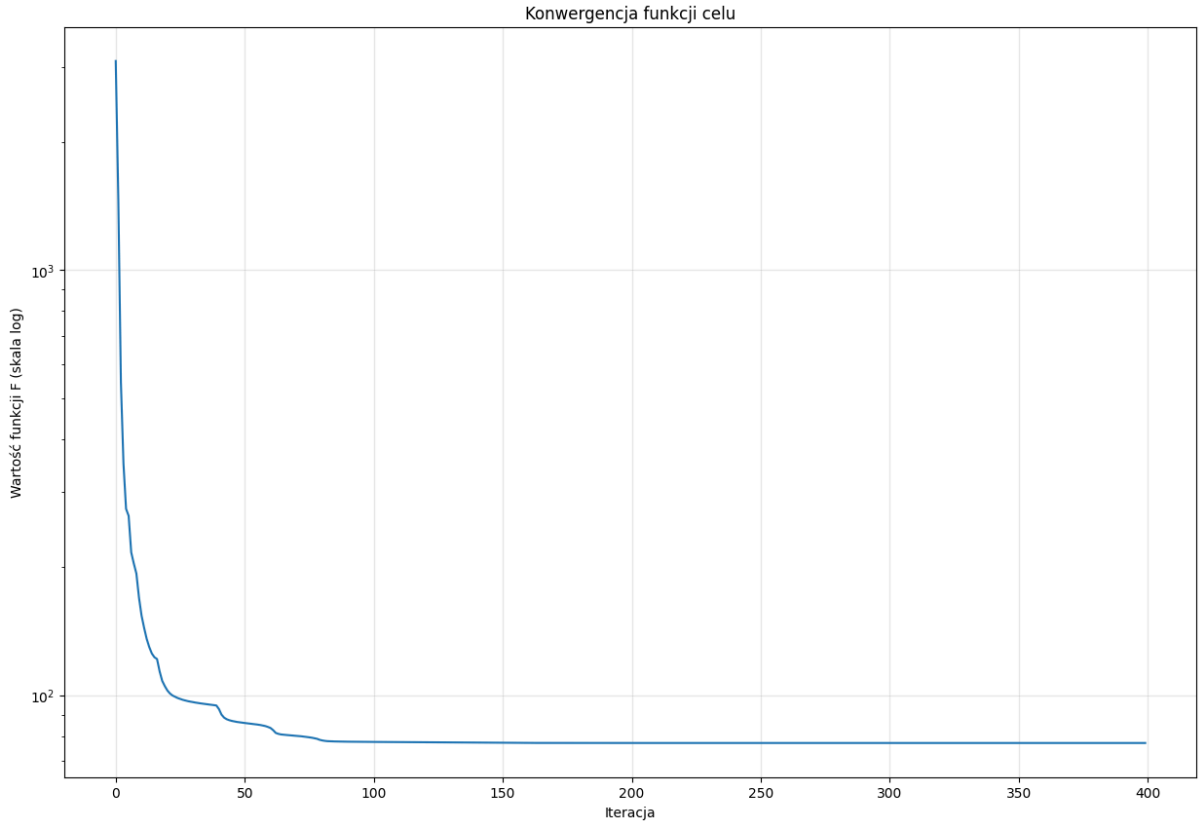
Wstęp

W tym zadaniu wyznaczamy najkrótszą ścieżkę złożoną z n połączonych końcami odcinków pomiędzy dwoma punktami na płaszczyźnie minimalizując przy tym funkcję celu wyrażoną wzorem:

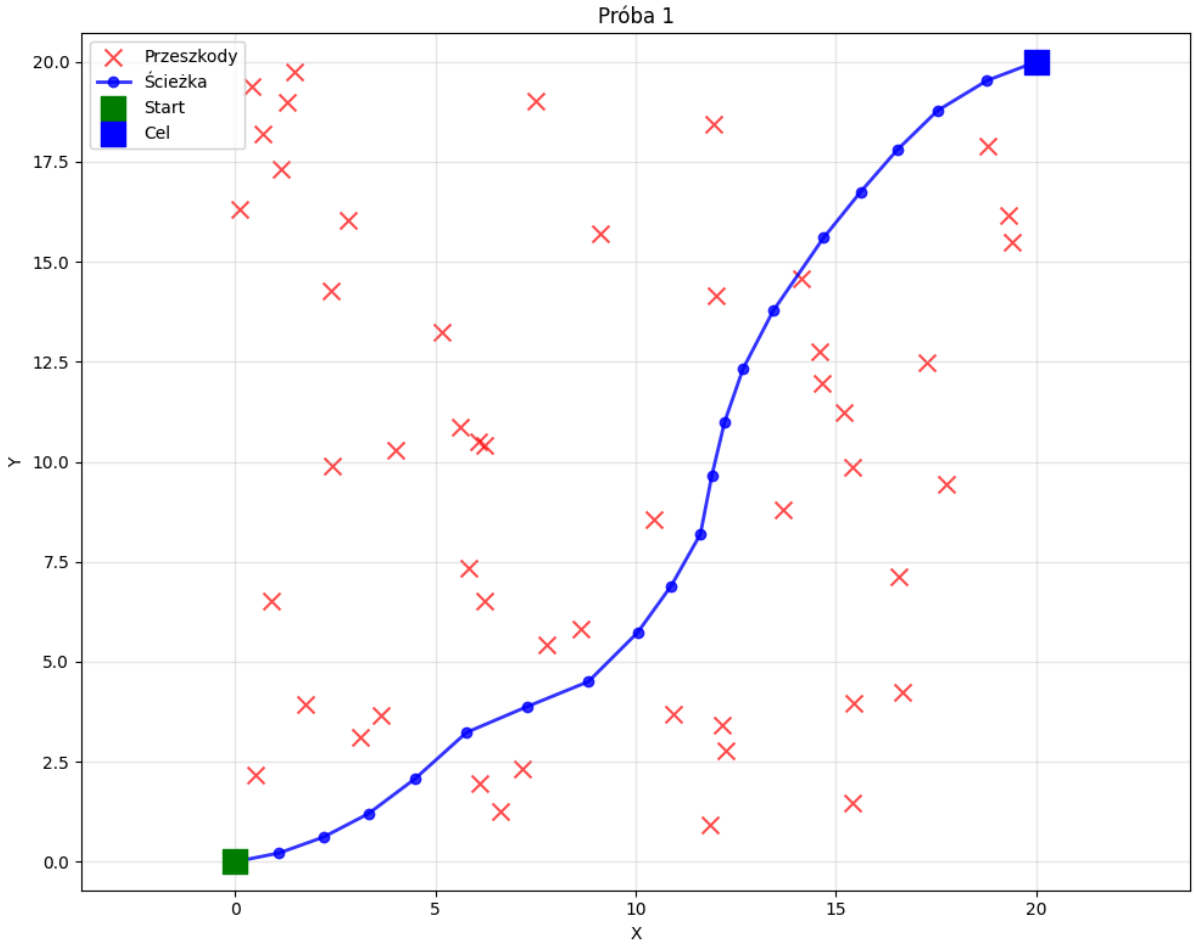
$$F(x^{(0)}, x^{(1)}, ..., x^{(n)}) = \lambda_1 \sum_{i=0}^{n} \sum_{j=1}^{k} \frac{1}{\epsilon + ||x^{(i)} - r^{(j)}||_2^{\,2}} + \lambda_2 \sum_{i=0}^{n-1} ||x^{(i+1)} - x^{(i)}||_2^{\,2},$$

gdzie $x^{(0)}, x^{(1)}, ..., x^{(n)}$ to punkty leżące między odcinkami.

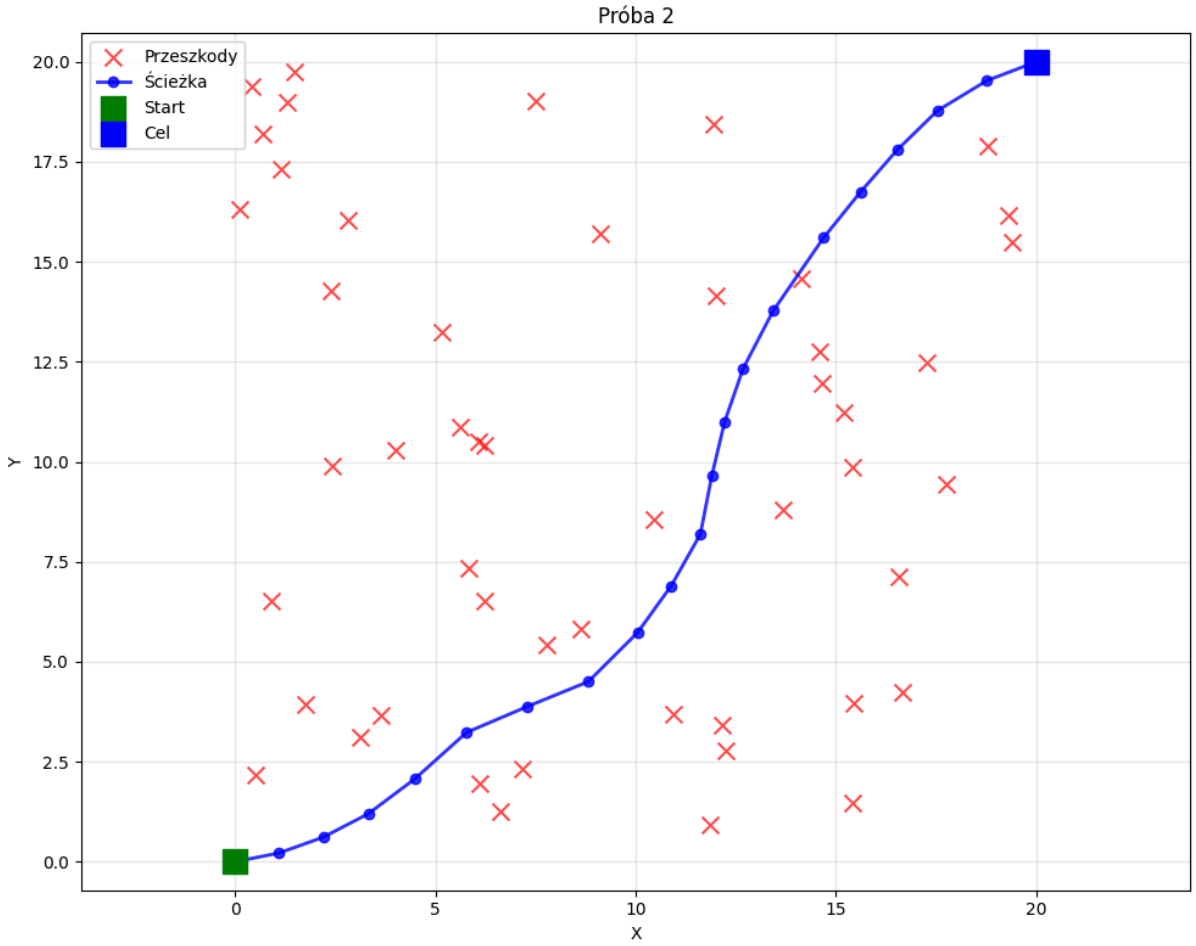
Wykresy



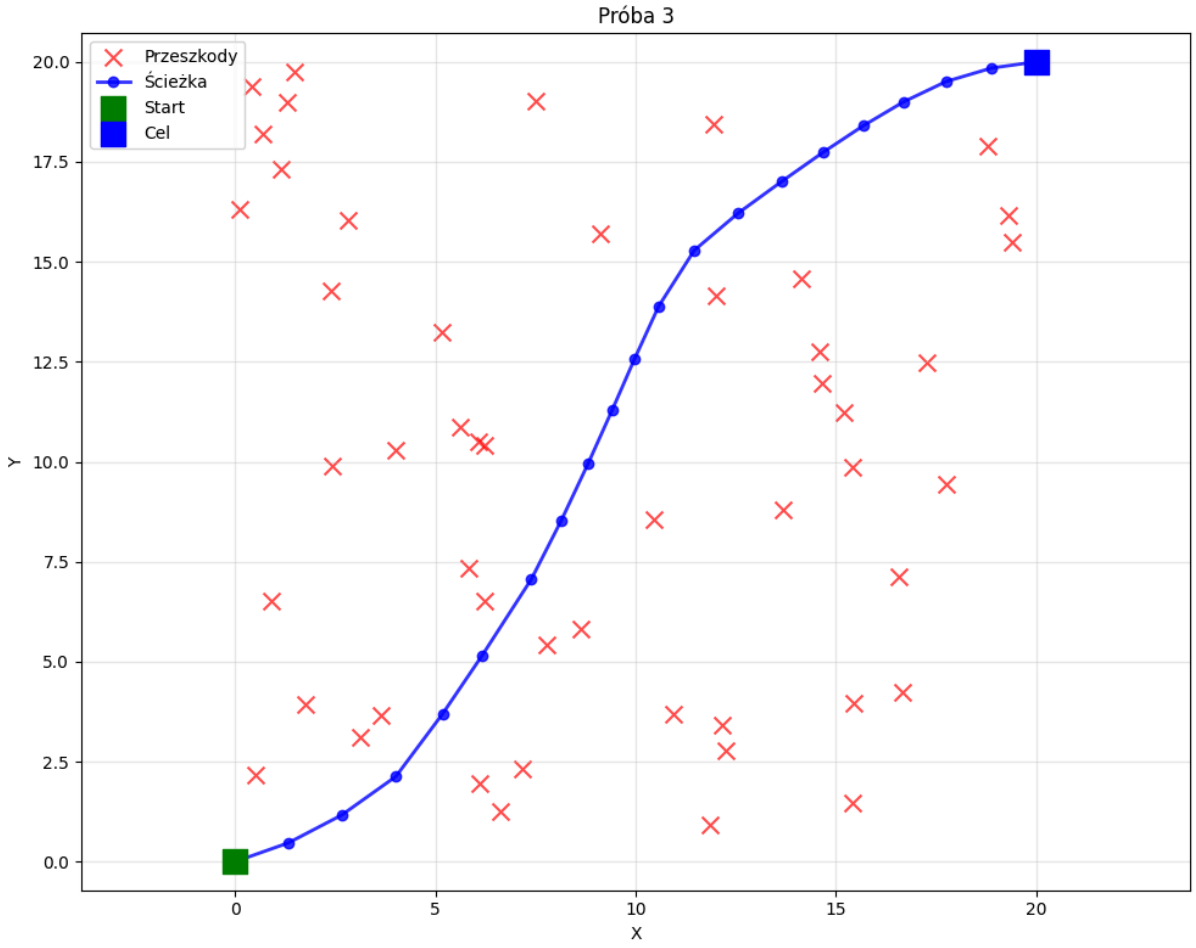
Wykres 1 - Konwergencja funkcji celu



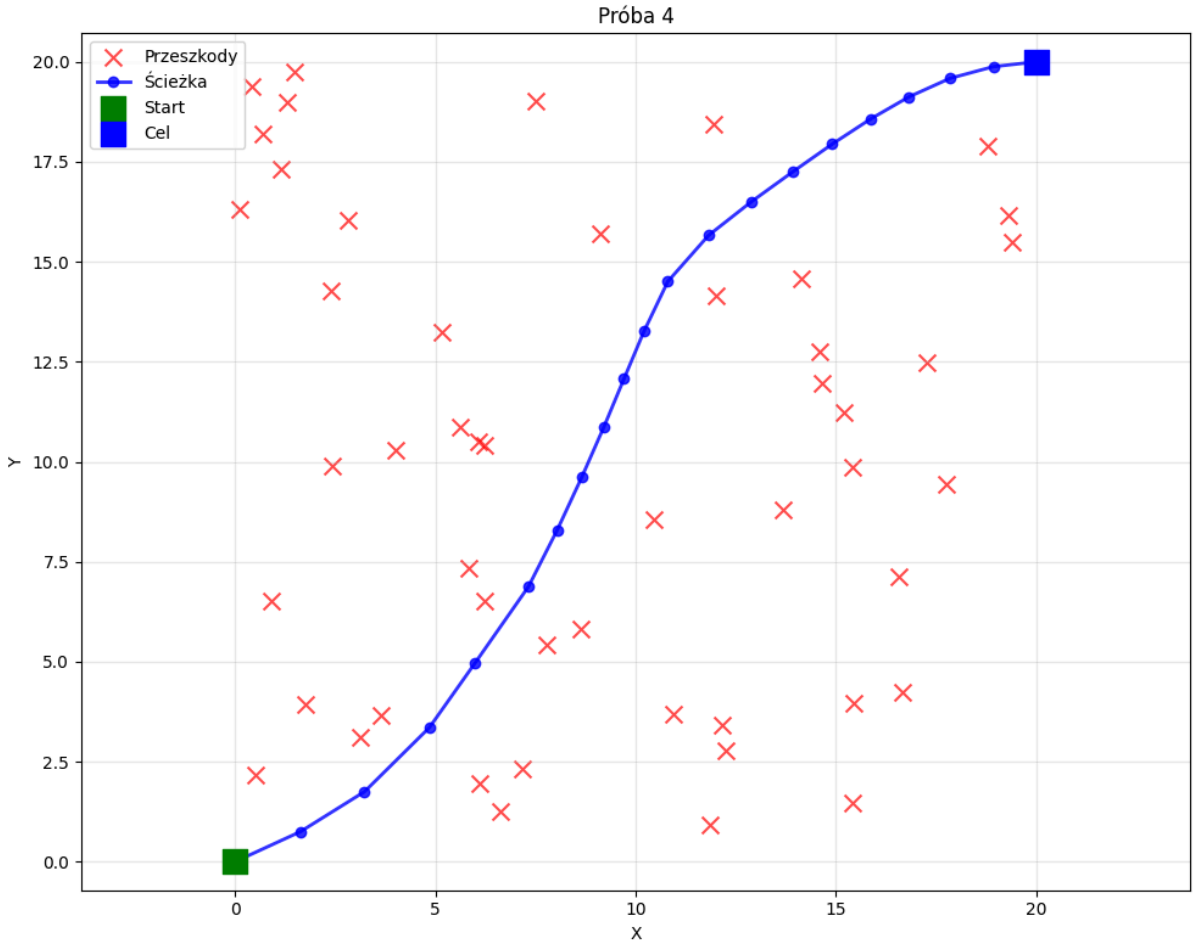
Wykres 2 - Próba 1



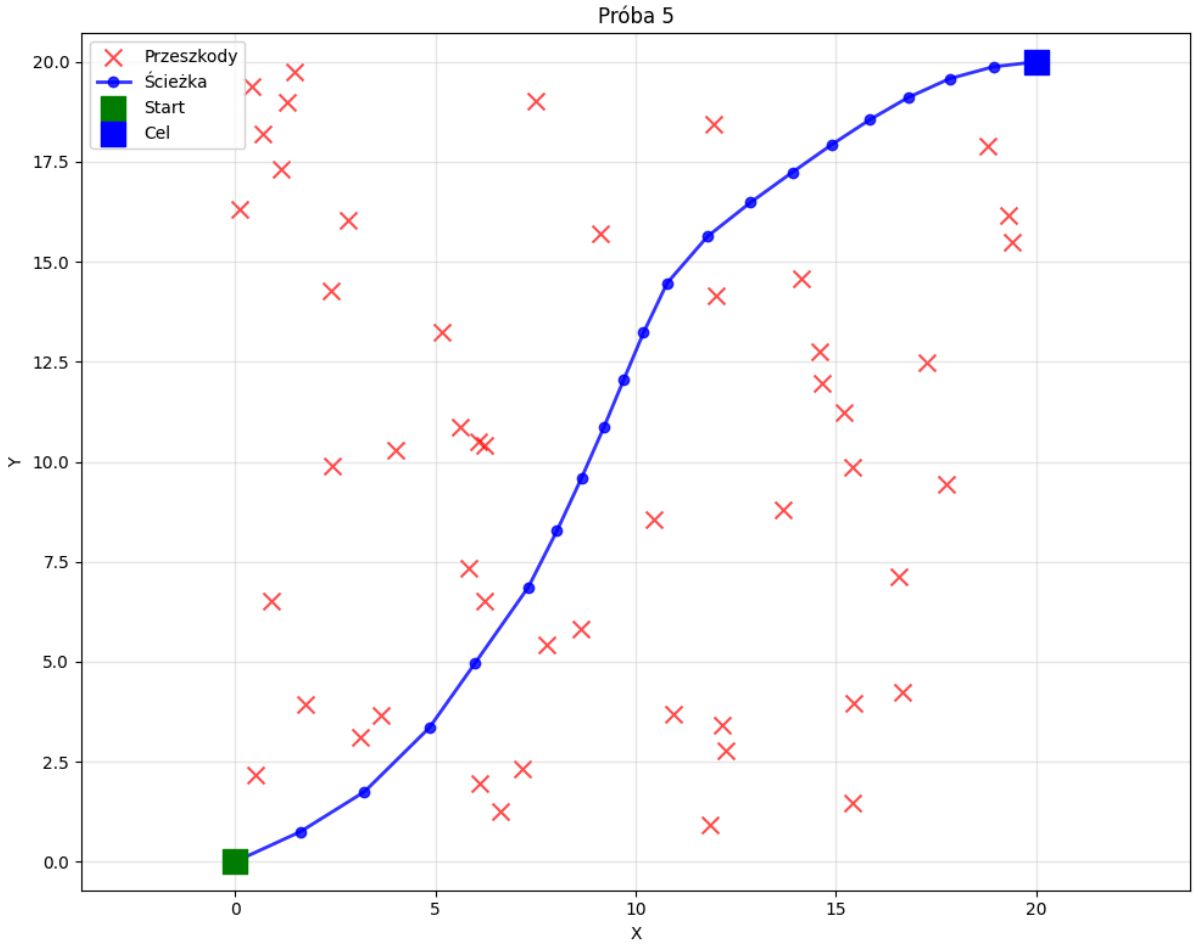
Wykres 3 - Próba 2



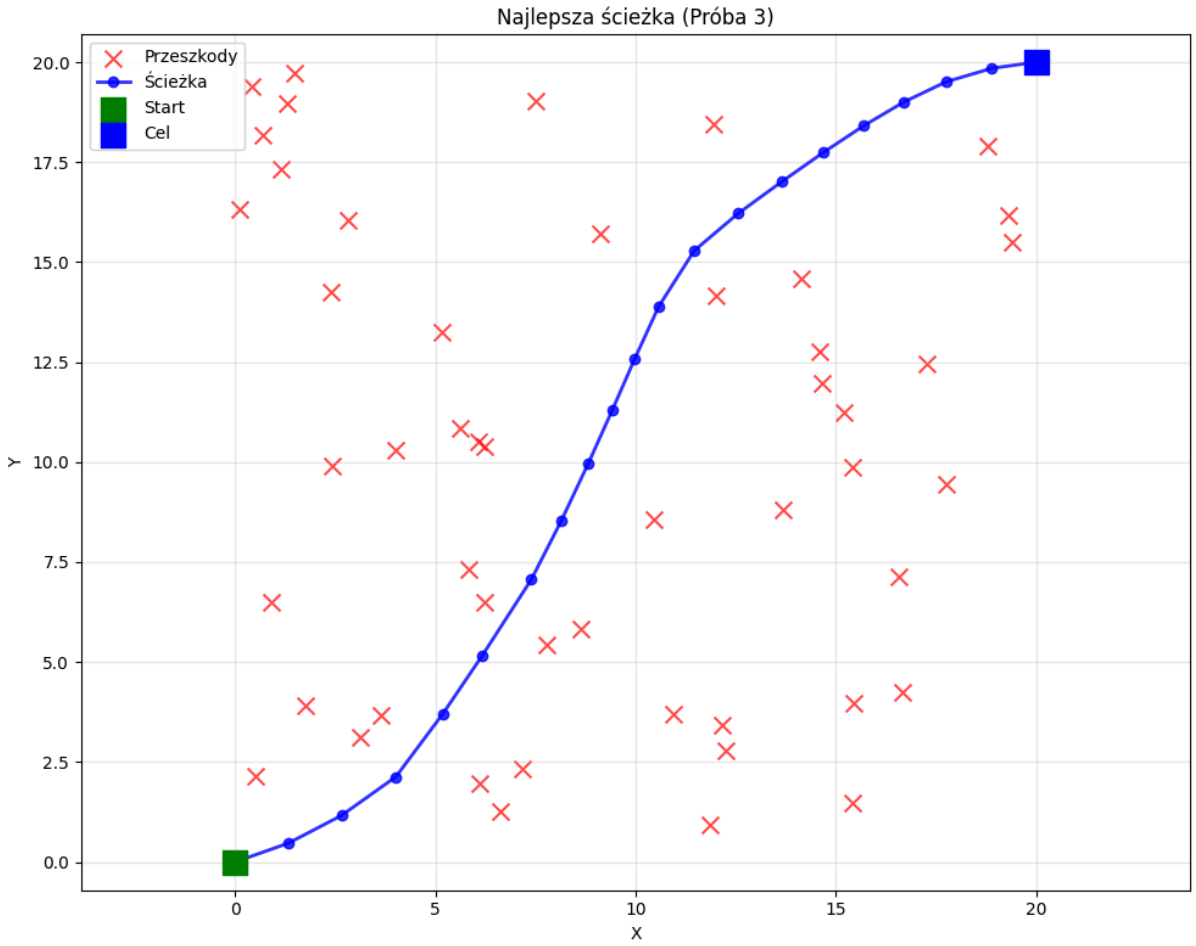
Wykres 4 - Próba 3



Wykres 5 - Próba 4



Wykres 6 - Próba 5



Wykres 7 - Najlepsza próba (próba 3)

Wykres 8 - Konwergencja funkcji celu dla najlepszego rozwiązania

Podsumowanie wyników

Najlepsza próba: 3
Najlepsza wartość funkcji celu: 72.665219
Próba 1: F = 77.210476
Próba 2: F = 77.210476
Próba 3: F = 72.665219
Próba 4: F = 73.773679
Próba 5: F = 73.774295

Wnioski:

- Metoda spadku wzdłuż gradientu uzyskała najwyższą dokładność kosztem około 100× dłuższego czasu obliczeń niż rozwiązania analityczne (normalne równania), co czyni ją opłacalną dopiero przy bardzo dużych zbiorach danych i ograniczonej liczbie cech ze względu na złożoność obliczeniową.
- Dla zadania robota różne losowe punkty startowe prowadziły do rozbieżności w wartościach funkcji celu (min F ≈ 72,7 vs. max F ≈ 77,2),

co wskazuje na istnienie lokalnych minimów i konieczność wielokrotnej inicjalizacji lub zastosowania heurystyk globalnych.
- Rola parametrów λ_1, λ_2 i ε: Ustawienie równych wag $\lambda_1=\lambda_2=1$ oraz stabilizującego $\varepsilon=10^{-13}$ umożliwiło uniknięcie kolizji z przeszkodami bez znacznego wydłużania ścieżki, jednak wyraźnie widać konieczność tuningu λ_1/λ_2, aby dostosować kompromis między dystansem a bezpieczeństwem.

Źródła:

- Prezentacje z MS Teams (zespół MOwNiT 2025)
- pythonnumericalmethods.studentorg.berkeley.edu