WSI Laboratorium 1
Optymalizacja za pomocą Algorytmu Gradientowego

Filip Misztal 310276

28 października 2023

1 Opis Implementowanych Algorytmów

Przedmiotem tego zadania była implementacja algorytmu gradientu prostego do znajdowania lokalnego minimum funkcji celu. Sposób jego działania wygląda następująco:

1. Wybieram wektor punktów startowych x_0 (konkretnie w tym przypadku; w ogólności może to być pojedyńczy punkt)
2. Wyliczam nowy wektor według wzoru: $x_{k+1} = x_k - \alpha \cdot \bigtriangledown f(x_k)$
3. Sprawdzam kryteria stopu (czy osiągnięty wynik nie mieści się w ustalonej tolerancji lub nie przebiegła maksymalna liczba iteracji) - jeśli któryś z nich jest spełniony, zwracam wektor x_{k+1} jako rozwiązanie
4. Powtarzam kroki 2 i 3 aż do spełnienia jednego z kryterium stopu

2 Opis planowanych eksperymentów numerycznych

W celu przetestowania algorytmu, zaplanowałem serię eksperymentów badających wpływ zmiennych na jego działanie, a konkretnie kroku oraz wartości parametru α funkcji celu. Wartości początkowe wektora x_0 są jednorazowo losowane z zakresu $[-100, 100]$ dla wszystkich testów. Zdecydowałem się także użyć stałych wartości pozostałych parametrów:

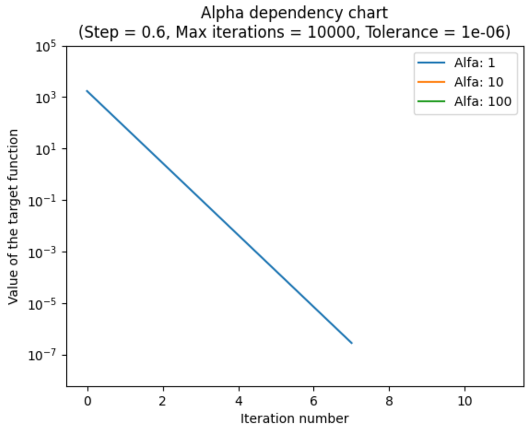
- Tolerancja: 10^{-6}
- Maksymalna liczba iteracji: 10^4.

Badane wartości parametrów:

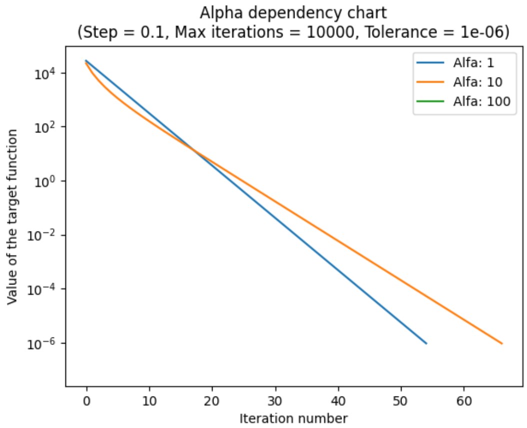
- Krok: 0.6, 0.1, 0.05, 0.01 oraz 0.001
- α: 1, 10 oraz 100

3 Opis uzyskanych Wyników

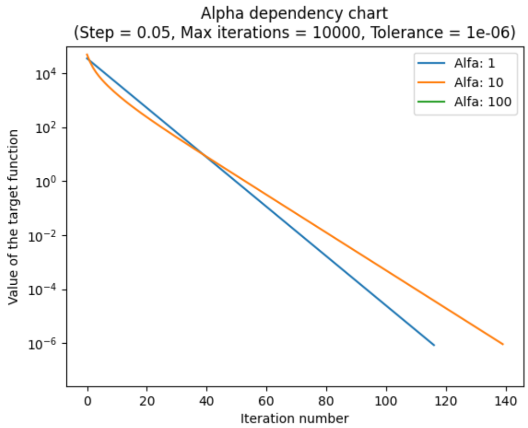
Wykresy przedstawiające wyniki działania algorytmu w czasie w zależności od parametru α:



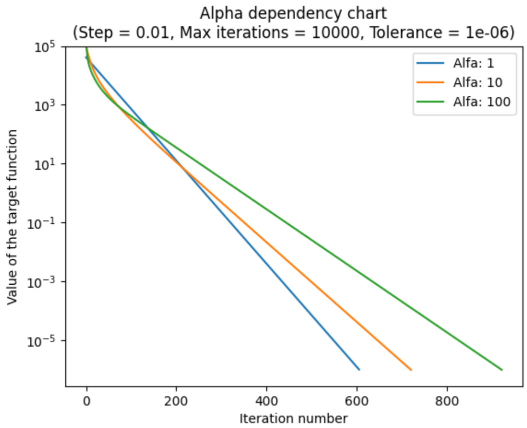
(a) Krok = 0.6



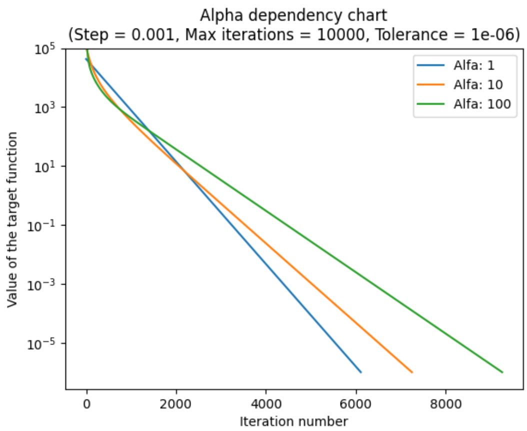
(b) Krok = 0.1



(c) Krok = 0.05



(d) Krok = 0.01



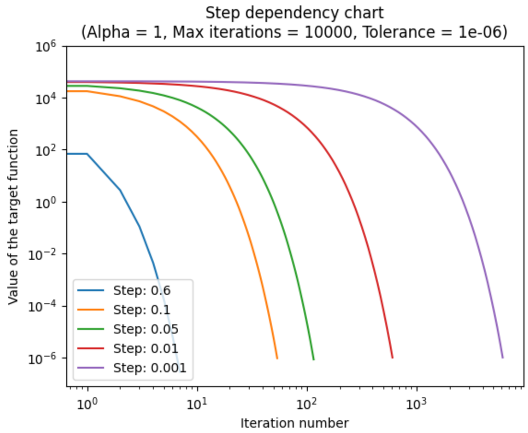
(e) Krok = 0.001

Rysunek 1: Wykresy zależności od parametru α

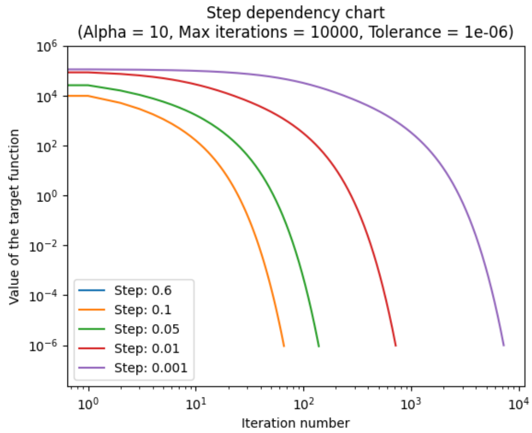
Na powyższych wykresach wyraźnie widoczna jest ogólna tendencja, że algorytm gradientu prostego potrzebuje więcej iteracji, aby znaleźć minimum funkcji celu o wyższym współczynniku α (szczególnie widoczne na wykresach 1d i 1e). Warto także zauważyć, że na wykresie 1a przedstawiono jedynie przebieg optymalizacji dla funkcji celu o α równym 1, a na wykresach 1b i 1c brak danych dla funkcji celu z $\alpha = 100$. Wynika to z faktu, że krok optymalizacyjny dla tych brakujących funkcji był zbyt duży, prowadząc do niestabilności działania algorytmu gradientu prostego – wartości funkcji zamiast zbliżać się do zera, dążyły w kierunku nieskończoności.
Możemy oszacować, kiedy to zjawisko zajdzie, obliczając hesjan funkcji celu. Odwrotność największej

wartości własnej macierzy hesjanu określa, od jakiej wartości kroku można spodziewać się oscylacji, a wartość ta pomnożona przez 2 wskazuje, kiedy mogą zachodzić rozbieżności.

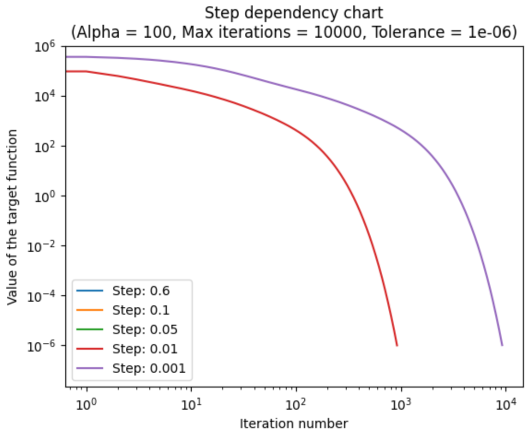
Wykresy przedstawiające wyniki działania algorytmu w czasie w zależności od kroku:



(a) Krok = 1



(b) Krok = 10



(c) Krok = 100

Rysunek 2: Wykresy zależności od kroku

Powyższe wykresy doskonale ilustrują, w jaki sposób niewielka wartość kroku znacząco wydłuża czas trwania działania algorytmu. Na przykład, na wykresie 2b można zauważyć, że algorytm osiągnął wartość bliską zeru po o rząd wielkości większej liczbie iteracji w przypadku kroku o wartości 0.001 w porównaniu z krokiem o wartości 0.1.

4 Wnioski z przeprowadzonych badań

Na podstawie przeprowadzonych eksperymentów można wyciągnąć następujące wnioski:

- Wybór optymalnej wartości kroku jest kluczowym czynnikiem wpływającym na efektywność algorytmu gradientowego. Konieczne jest dostosowanie tej wartości do konkretnej funkcji celu. Zbyt duża wartość kroku może prowadzić do rozbieżności, podczas gdy zbyt mała może znacząco wydłużyć czas trwania procesu optymalizacji.
- W przypadku naszej konkretnie analizowanej funkcji celu, większa wartość parametru alfa powoduje, że składniki z większymi wykładnikami mają większy wpływ na wartość funkcji. W związku z tym algorytm gradientu prostego musi działać ostrożnie, korzystając z mniejszego kroku, aby uniknąć problemów z rozbieżnością.

- Należy unikać zbyt małego kroku, ponieważ może to znacząco wydłużyć czas trwania algorytmu bez znacznego wpływu na jego dokładność. Warto przeprowadzić eksperymenty, aby znaleźć optymalną wartość kroku, która zapewni dokładne wyniki przy minimalnym koszcie obliczeniowym.
- Eksperymentowanie z różnymi parametrami, takimi jak krok, tolerancja i maksymalna liczba iteracji, jest kluczowe w procesie optymalizacji. Dzięki temu można znaleźć optymalne ustawienia, które zapewnią satysfakcjonujące rezultaty dla konkretnego problemu optymalizacji.