WSI - ćwiczenie 1.

Przeszukiwanie przestrzeni

grupa 101

Wykonał: Adrian Pruszyński

1. Treść Zadania

W ramach pierwszego ćwiczenia należy zaimplementować metodę realizującą algorytm najszybszego spadku. Następnie należy zbadać zachowanie tych algorytmów dla:

- różnych wartości rozmiaru kroku w tabeli
- różnych wartości punktu początkowego

dla problemów minimalizacji podanych niżej funkcji celu. Proszę w szczególności o zwrócenie uwagi na osiągane wartości funkcji celu w funkcji liczby kroków algorytmu

Funkcje celu

Wymiarowość problemu	$f(x)$	$\nabla f(x)$
1	x^2	$2x$
2	$(x_1+a)^2+(x_2-a)^2-5\cos\left(10\sqrt{(x_1+a)^2+(x_2-a)^2}\right)$	$\begin{bmatrix}(x_1+a)(2+g_a(x))\\(x_2-a)(2+g_a(x))\end{bmatrix}$

gdzie $g_a(x) = \frac{50\sin\left(10\sqrt{(x_1+a)^2+(x_2-a)^2}\right)}{\sqrt{(x_1+a)^2+(x_2-a)^2}}$, x_i oznacza i-ty element wektora x, a a jest ostatnią cyfrą numeru indeksu studenta wykonującego zadanie.

2. Założenia

Implementacja algorytmu zawiera warunek stopu złożony z warunku ograniczającego maksymalną ilość kroków algorytmu, dla przypadków, gdy nie uda się wystarczająco zbliżyć do ekstremum oraz warunku gradientu mniejszego od ε, dla przypadków, gdy uda się zbliżyć dostatecznie blisko do ekstremum.

Dla badań przyjęto:

- Maksymalną ilość kroków algorytmu równą 50000.
- $\varepsilon = 0.001$

3. Prezentacja wyników badań

Wyniki w postaci tabelarycznej dla funkcji celu nr. 1

Tabela 1

wielkosc_kroku	epsilon	punkt_poczatkowy	f(x)	kroki	punkt_koncowy_x
0,8	0,001	3	0,00	18	0,00
0,1	0,001	3	0,00	39	0,00
0,01	0,001	3	0,00	431	0,00
0,001	0,001	3	0,00	4346	0,00
0,8	0,001	-49	0,00	23	0,00
0,1	0,001	-49	0,00	52	0,00
0,01	0,001	-49	0,00	569	0,00
0,001	0,001	-49	0,00	5741	0,00
0,8	0,001	50	0,00	23	0,00
0,1	0,001	50	0,00	52	0,00
0,01	0,001	50	0,00	570	0,00
0,001	0,001	50	0,00	5751	0,00
0,8	0,001	-500	0,00	28	0,00
0,1	0,001	-500	0,00	62	0,00
0,01	0,001	-500	0,00	684	0,00
0,001	0,001	-500	0,00	6901	0,00
0,8	0,001	1000	0,00	29	0,00
0,1	0,001	1000	0,00	66	0,00
0,01	0,001	1000	0,00	719	0,00
0,001	0,001	1000	0,00	7248	0,00

Wyniki w postaci tabelarycznej dla funkcji celu nr. 2

Tabela 2

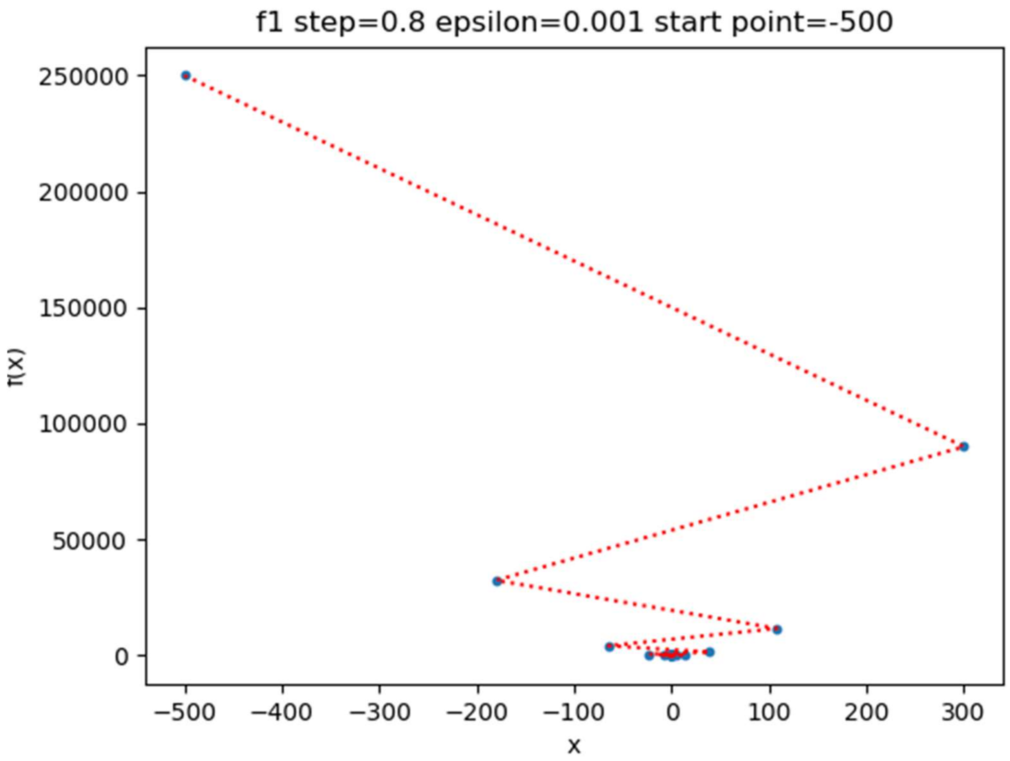
wielkosc_kroku	epsilon	punkt_poczatkowy_x1	punkt_poczatkowy_x2	f(x)	kroki	punkt_koncowy_x1	punkt_koncowy_x2
0,8	0,001	3	6	3299,56	50000	-25,68	-51,35
0,1	0,001	3	6	-3,42	50000	-0,56	-1,12
0,01	0,001	3	6	5,89	14699	0,42	0,85
0,001	0,001	3	6	42,58	20	3,08	6,16
0,8	0,001	-49	-53	623,78	4676	-16,96	-18,34
0,1	0,001	-49	-53	13,03	4127	1,93	2,08
0,01	0,001	-49	-53	504,46	50000	-15,26	-16,51
0,001	0,001	-49	-53	623,78	995	-16,96	-18,34
0,8	0,001	100	-30	592,81	7457	23,30	-6,99
0,1	0,001	100	-30	1,32	50000	-1,34	0,40
0,01	0,001	100	-30	504,46	50000	21,54	-6,46
0,001	0,001	100	-30	623,78	1197	23,93	-7,18
0,8	0,001	-500	800	686,48	50000	13,84	-22,14
0,1	0,001	-500	800	42,58	252	3,65	-5,84
0,01	0,001	-500	800	423,28	50000	-10,92	17,47
0,001	0,001	-500	800	623,78	2290	-13,24	21,18
0,8	0,001	1000	1000	533,08	37072	-16,36	-16,36
0,1	0,001	1000	1000	26,85	26652	3,98	3,98
0,01	0,001	1000	1000	623,78	229	17,66	17,66
0,001	0,001	1000	1000	623,78	2484	17,66	17,66

4. Analiza wyników badań

Przyglądając się tabeli 1 widać, że algorytm pomimo różnych punktów startowych za każdym razem dąży do osiągnięcia tego samego punktu końcowego, punktu x=0. Jest to zgodne z oczekiwaniami, gdyż funkcja ta ma jedno minimum w punkcie 0.

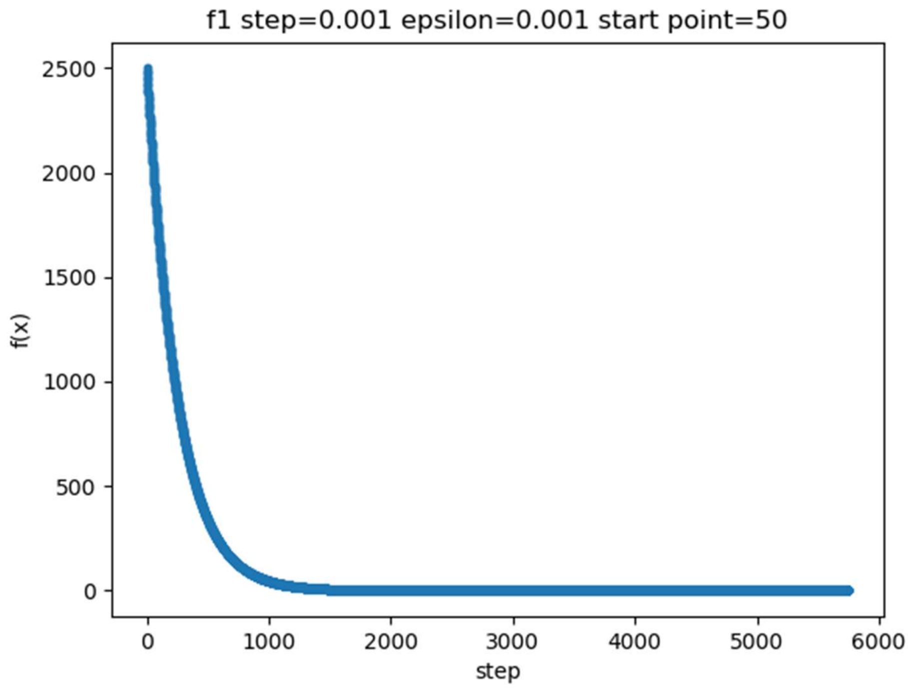
Chcąc zbadać jak na przebieg algorytmu wpływa parametr kroku warto przeanalizować wykres wartości funkcji od x.

Rysunek 1



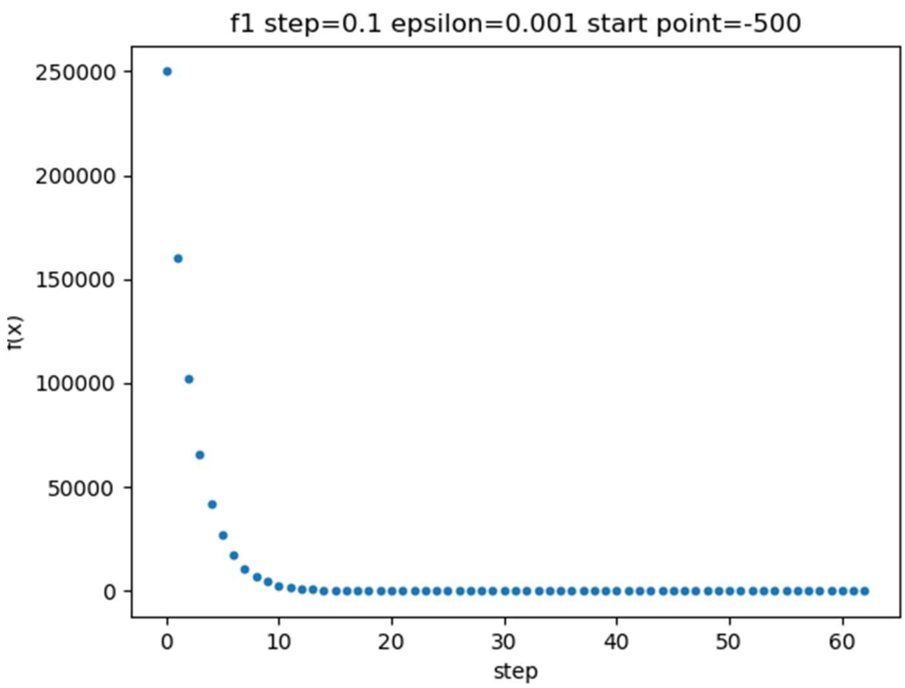
Jak łatwo zauważyć na rysunku 1, zbyt duża wartość kroku powoduje przeskoki kolejno wyznaczonych punktów nad ekstremum. Może to spowodować zwiększenie liczby kroków, potrzebnych do osiągnięcia minimum. Przy dostatecznie dużej wartości kroku może dojść do wzrostu wartości funkcji celu, a w rezultacie problem z osiągnięciem ekstremum.

Rysunek 2

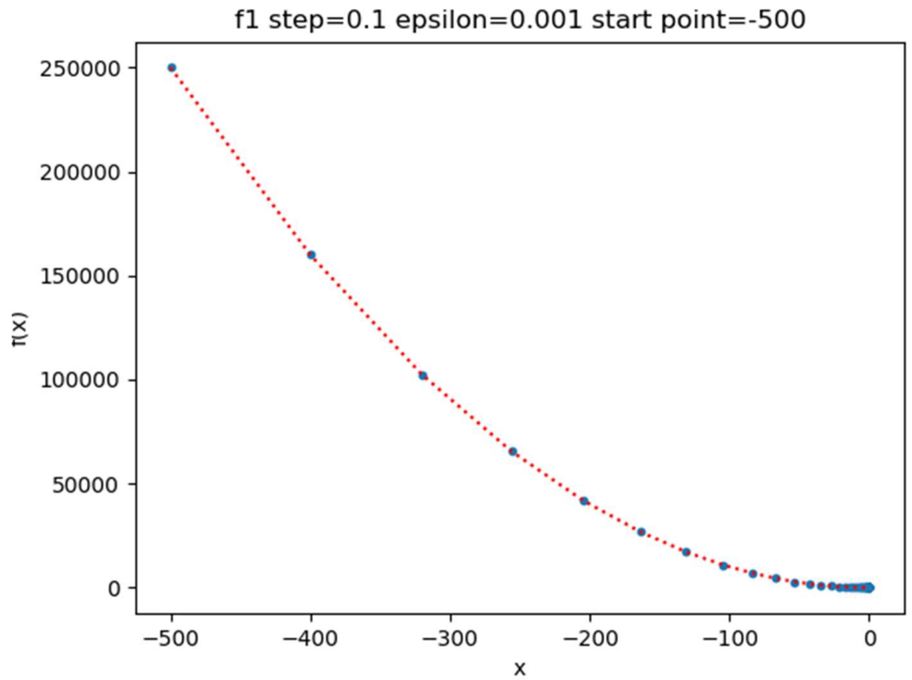


Na rysunku 2 możemy zauważyć sytuację, gdy wartość kroku jest zbyt niska przez co ilość kroków, które muszą zostać wykonane dla osiągnięcia minimum znacząco wzrasta.

Rysunek 3



Rysunek 4

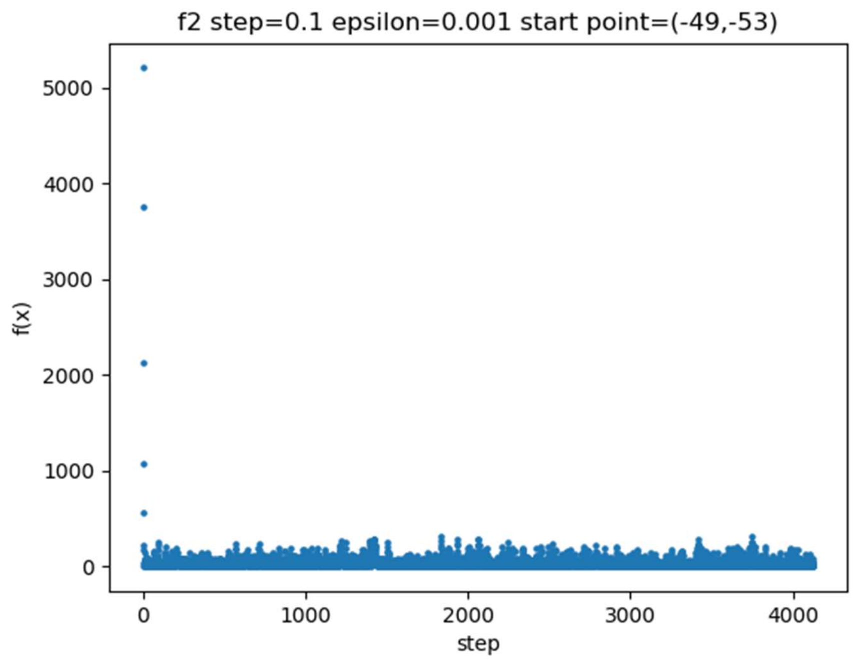


Na rysunku 3 oraz 4 widać sytuację, gdy wartość kroku została rozsądnie dobrana, dzięki temu minimum jest osiągalne w stosunkowo niewielkiej liczbie kroków. Bez występowania przeskoków nad ekstremum. Widać jednak, że wartość kroku mogłaby zostać dobrana jeszcze lepiej- dzięki czemu zmniejszona zostanie liczba kroków algorytmu.

W przypadku tabeli 2 widać, że nie dla każdego przypadku udało się dostatecznie zbliżyć do ekstremum. Ponadto zauważyć można, że algorytm wyznaczył różne punkty końcowe w zależności zarówno od punktu startowego jak i wartości skoku. Pozwala to wywnioskować, że funkcja posiada wiele ekstremów lokalnych.

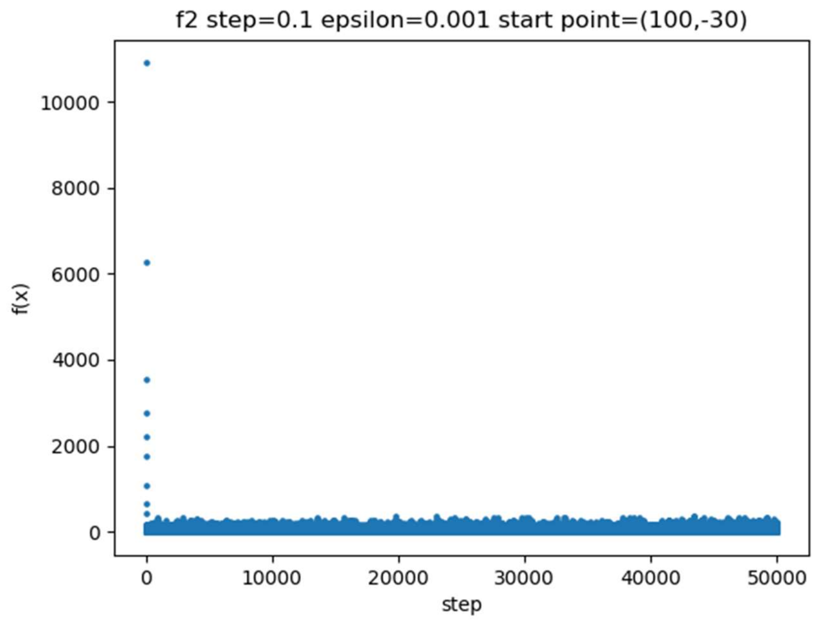
Przyglądając się punktom końcowym dla punktu początkowego (1000, 1000) można łatwo zauważyć, że dla niektórych wartości kroku algorytm dąży do osiągnięcia innego ekstremum. Przyczyną takiego zjawiska jest możliwość zrobienia kroku w kierunku jednego z minimów lokalnych jednak na tyle dużego, że w kolejnych krokach algorytm dążyć będzie do innego minimum niż ww.

Rysunek 5



Rysunek 5 przedstawia sytuację, gdy przez zbyt dużą wartość kroku wartość funkcji oscyluje w okolicy ekstremum, przez to wydłuża się ilość kroków niezbędna do dostatecznego zbliżenia się do minimum.

Rysunek 6



Jak widać na rysunku 6, w niektórych przypadkach sytuacja poruszona w opisie rysunku 5, może to spowodować nieosiągnięcie dostatecznego zbliżenia się do ekstremum.

5. Wnioski i podsumowanie

Analiza wyników uwidoczniła jak ważny dla efektywności algorytmu jest dobór odpowiednich parametrów wartości kroku oraz punktu startowego

Istnieją zagrożenia związane ze zbyt wysokim parametrem wartości kroku mogą one wydłużyć czas, w którym zostanie osiągnięte ekstremum lub doprowadzić do nieosiągnięcia ekstremum.

Pomimo relatywnego bezpieczeństwa niskiej wartości kroku wiąże się ono ze znacznym zwiększeniem liczby kroków niezbędnych do osiągnięcia minimum

Warto zauważyć, że algorytm nie gwarantuje osiągnięcia ekstremum globalnego. Osiągnięte ekstremum, o ile uda się je osiągnąć, zależeć będzie zarówno od parametru kroku jak i punktu początkowego.

Pokrycie przestrzeni punktami początkowymi zwiększa prawdopodobieństwo znalezienia korzystniejszego rozwiązania.

Algorytm jest szczególnie użyteczny dla funkcji posiadających jedno ekstremum.