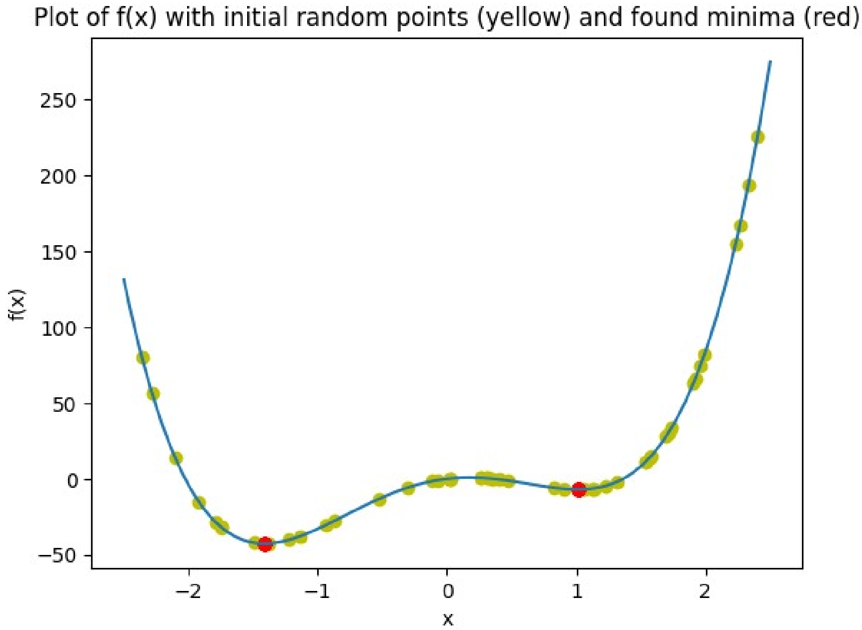
WSI Ćwiczenie 1 - metoda gradientu prostego

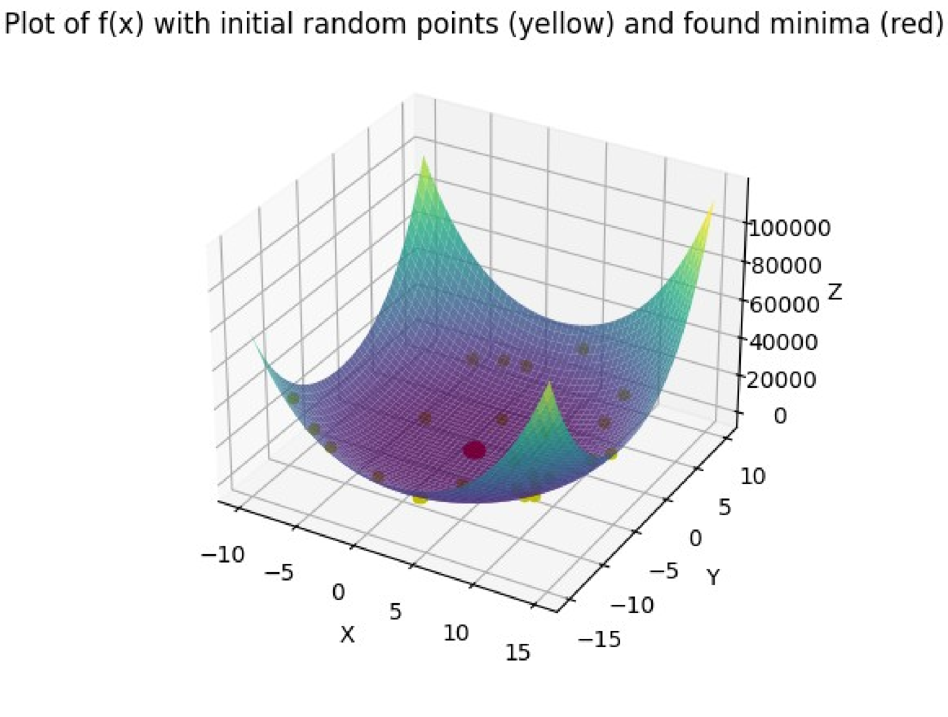
Jakub Romankiewicz, 325063

Utworzono klasę GradientDescentSolver posiadającą funkcję solve() znajdującą minimum funkcji metodą gradientu prostego. Funkcja f(x) blisko siebie posiada minimum globalne oraz minimum lokalne, oba udało się odkryć stworzoną funkcją.



Rys. 1 – Poglądowy przebieg funkcji f(x) z zaznaczonymi losowymi punktami początkowymi algorytmu (żółte) i końcowymi punktami reprezentującymi znalezione minima funkcji (czerwone) .

Funkcja g(x1, x2) posiada jedno, bardzo płaskie minimum.



Rys. 2 – Poglądowy przebieg funkcji g(x1, x2) z zaznaczonymi losowymi punktami początkowymi algorytmu (żółte) i końcowymi punktami reprezentującymi znalezione minima funkcji (czerwone).

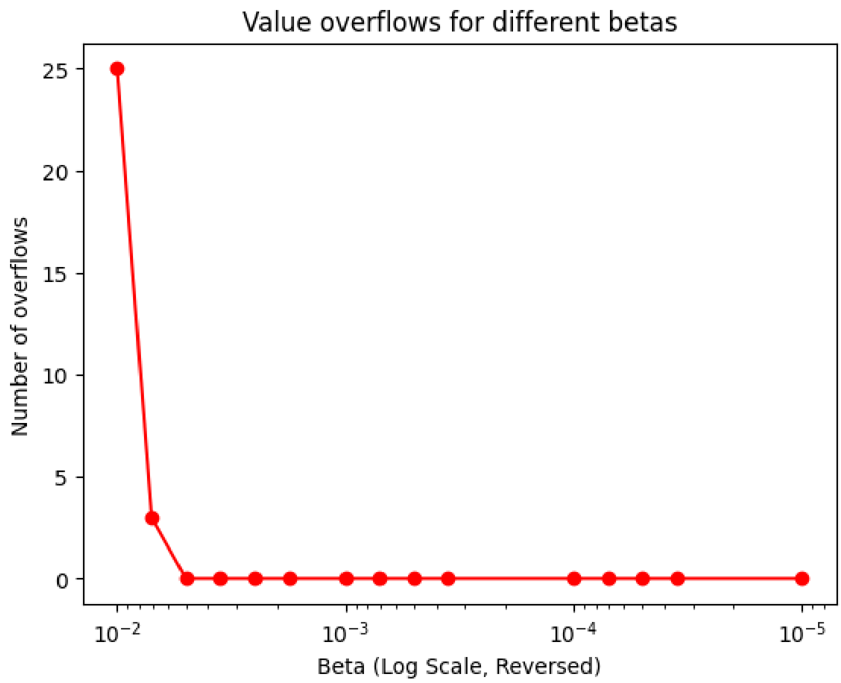
Algorytm zainicjalizowano z hiperparametrami:
- beta – prametr reprezentujący wielkość kroku
- eps – (ustawiono na 10^-9) maksymalna różnica w pozycji pomiędzy kolejnymi iteracjami przy której algorytm kończy pracę.
- max_iter – (ustawiono na 10000) maksymalna ilość iteracji jaką może wykonać algorytm.

Wykonano badanie polegające na wybraniu kilku parametrów beta (kroku), i dla każdego z nich sprawdzono wyniki działania algorytmu na 50 losowo wybranych punktach w rozsądnych granicach dziedziny funkcji. W ten sposób będzie można wybrać optymalny krok dla funkcji.

Statystyki dla funkcji f(x)

1. Zbieżność algorytmu

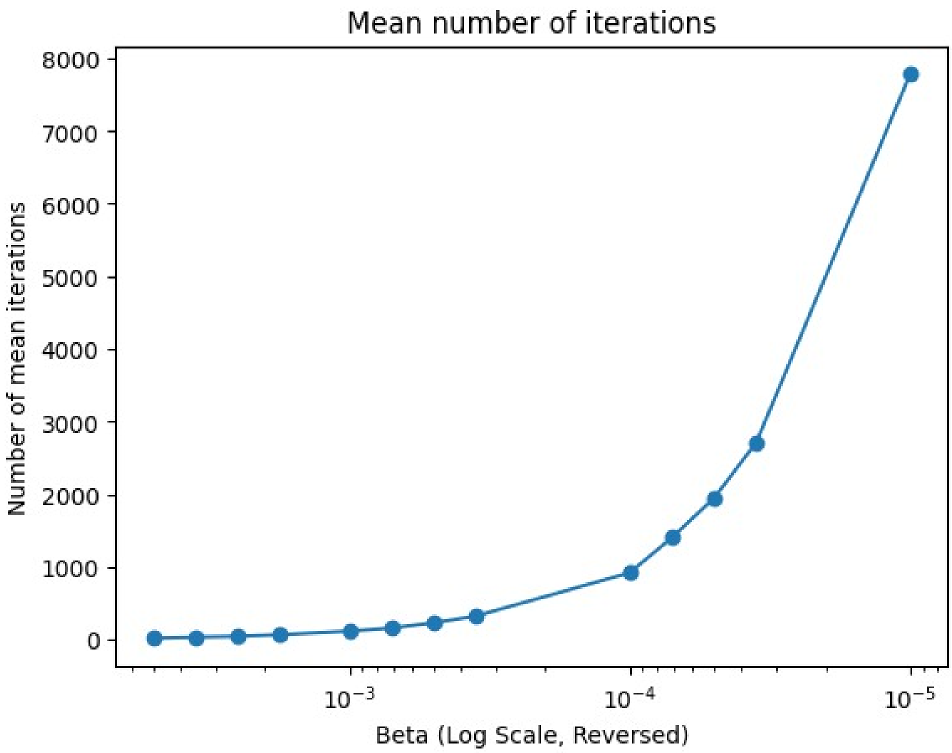
Jeżeli krok był ustawiony na zbyt duży algorytm wpadał w oscylacje i nsatępowało przepełnienie zmiennych. Zawężono obszar poszukiwań wartości bety eliminując takie przypadki.



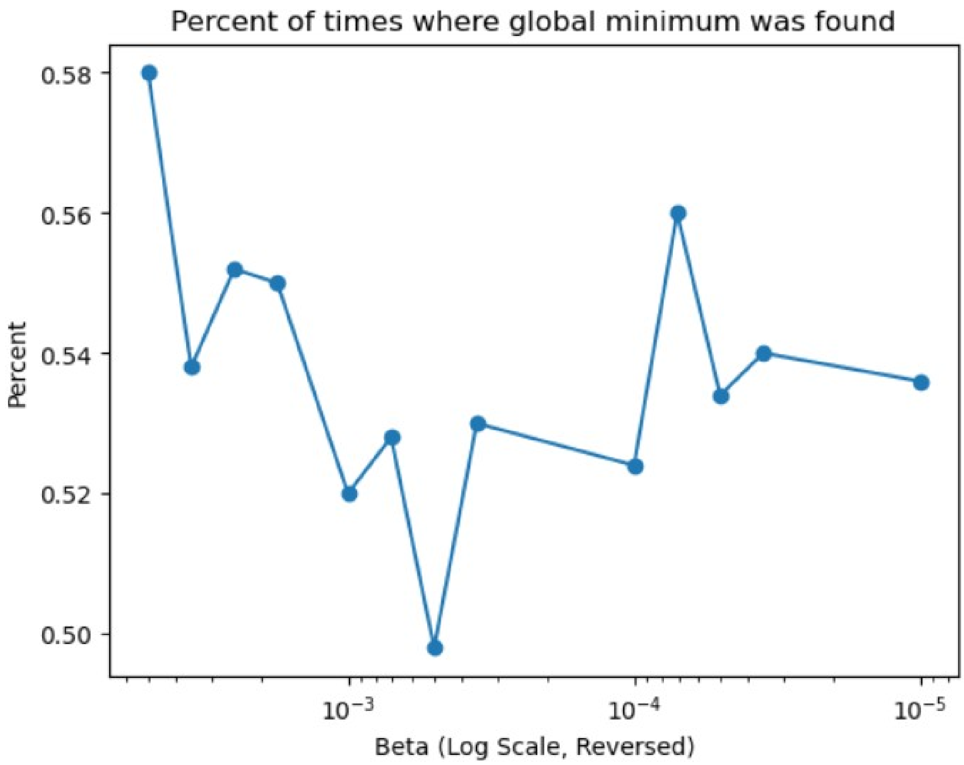
Dla wybranych granic funkcji f(x) algorytm nie jest zbieżny dla wszystkich przypadkowo wybranych wartości beta

2. Średnia ilość iteracji algorytmu

Zmniejszanie kroku zwiększa ilość iteracji jaką musi wykonać algorytm.



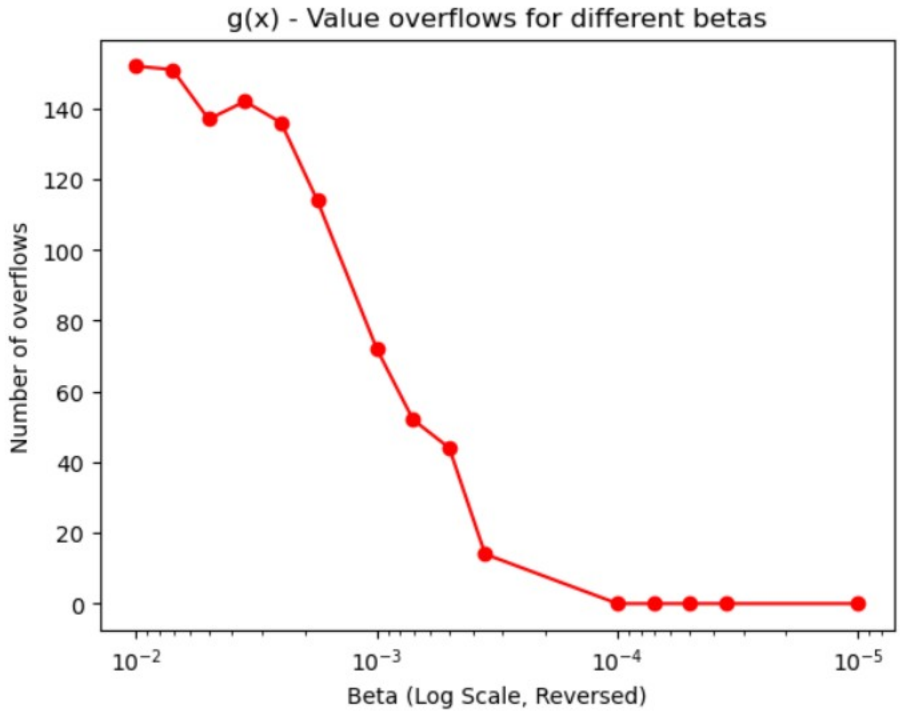
Ponieważ funkcja f(x) ma bardzo blisko siebie minimum lokalne i globalne bardzo często zostało znajdowane niewłaściwe minimum. Postanowiono więc wyznaczyć zależność częstości znajdowania właściwego minimum od bety dla bardzo dużej liczby losowych punktów =500.



Jak widać w każdym przypadku poprawność rozwiązania jest niewiele większa niż 50% (bo 2 minima). Pokazuje to, że istotnym jest wykonanie wielu testów dla losowych punktów i wybranie najmniejszego rozwiązania. Po wykonaniu wielu testów widać jednak większą powtażalność poprawnych rozwiążań dla większych dużych kroków. Kroki te częściej pomijają minimum lokalne i wpadają w globalne.

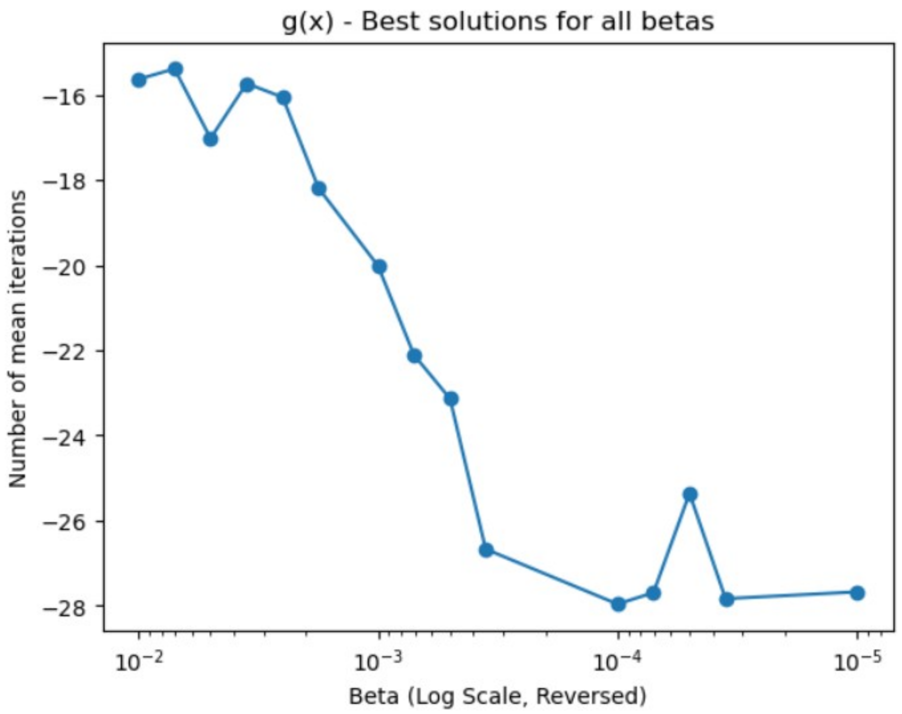
Statystyki dla funkcji g(x)

1. Zbieżność algorytmu



Jak widać algorytm jest zbieżny dopiero dla dużo mniejszych współczynników beta, Jest to jednak tak zależne od samej funkcji jak i od przyjętych granic pujktów początkowych.

2. Wartość najlepszego z rozwiązań dla każdego z współczynników beta



Widać że dostajemy lepsze rozwiązania im mniejszy krok, lecz zależność ta wypłaszcza się dla bardzo małych współczynników beta ze względu na płaskość w minimum samej funkcji.

3. Liczba Iteracji