Kacper Ptaszek
nr indeksu 331427

Sprawozdanie WSI ćwiczenia 1.

Zadanie 1

Treść zadania:

1. Znaleźć rozwiązanie optymalne przez przegląd wyczerpujący (analizuje wszystkie kombinacje).
2. Rozwiązać problem przy użyciu heurystyki: do plecaka pakujemy przedmioty według kolejności wynikającej ze stosunku *p*/*m*. Przeglądamy listę przedmiotów do końca, chyba że plecak jest już pełen lub zostało w nim tak mało miejsca, że już na pewno nic się nie zmieści. Uwaga: heurystyka to nie funkcja heurystyczna. Nie używamy tu jeszcze funkcji heurystycznej i algorytmu A*.

Pytania i odpowiedzi:

- Jakie rozwiązania i jaką wartość funkcji oceny uzyskano? Czy uzyskano takie same rozwiązania?

  Dla podanych danych podczas wykładu i w treści zadania:
  m = np.array([8, 3, 5, 2]) #masa przedmiotów
  M = np.sum(m)/2 #niech maksymalna masa plecaka będzie równa połowie masy przedmiotów, czyli M=9
  p = np.array([16, 8, 9, 6]) #wartość przedmiotów

  Rozwiązanie optymalne uzyskane przez przegląd wyczerpujący to funkcja oceny równa 17(wybrane przedmioty o wartościach 8 i 9).
  Za to dla rozwiązania przy użyciu heurystyki funkcja oceny to 14(wybrane przedmioty o wartościach 6 i 8).
  Zatem rozwiązania różnią się, co wynika z tego że przy użyciu heurystyki zachłannie bierzemy przedmioty które mają największy stosunek wartości/wagi.

- Jak dużą instancję problemu (liczba przedmiotów) da się rozwiązać w około minutę metodą przeglądu wyczerpującego? Ile czasu zajmie rozwiązanie tego problemu metodą zachłanną (używając heurystyki)? Odpowiednio długie wektory *m* i *p* należy wylosować, *M* = np.sum(*m*)/2.

  Metodą przeglądu wyczerpującego jesteśmy w stanie rozwiązać problem z 23 przedmiotami. Z przeprowadzonych doświadczeń wynika, że średni czas działania metody wyczerpującej dla 23 przedmiotów to 57 sekund. Czas rozwiązania tego samego problemu metodą zachłanną jest bliski zera.

- Jak bardzo wydłuży obliczenia dodanie jeszcze jednego przedmiotu?

  Dodanie jeszcze jednego przedmiotu wydłuży czas około dwukrotnie, czyli o 57 sekund.

- Jakie wnioski można wyciągnąć na podstawie wyników tego ćwiczenia?

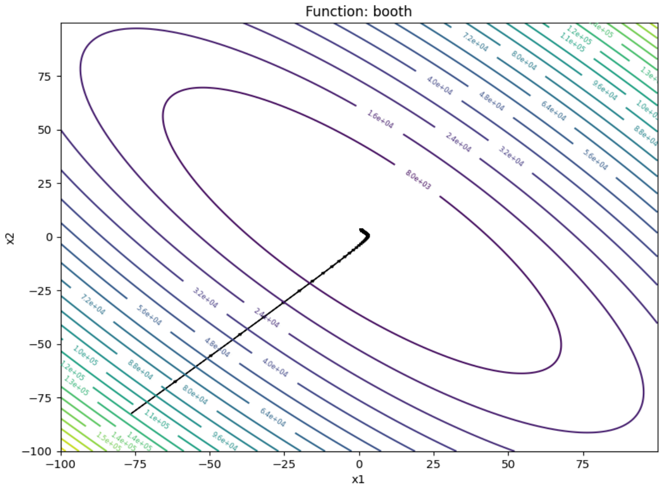
  Znajdowanie optymalnego rozwiązania metodą przeglądu wyczerpującego trwa bardzo długo już dla kilkudziesięciu przedmiotów, co nie jest pożądane. Zatem dla dużych problemów plecakowych lepiej użyć metody zachłannej, dla której nie mamy pewności uzyskania optymalnego rezultatu, lecz wynik ten jest zazwyczaj zbliżony do optimum, a czas znajdowania rozwiązania jest znacznie krótszy. Dla 1000 przedmiotów średni czas znajdowania rozwiązania przez metodę zachłanną to tylko 0,6 sekundy.

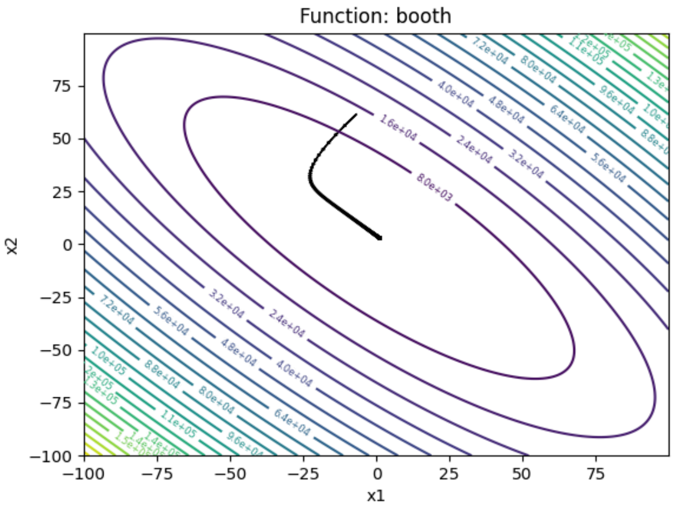
Zadanie 2

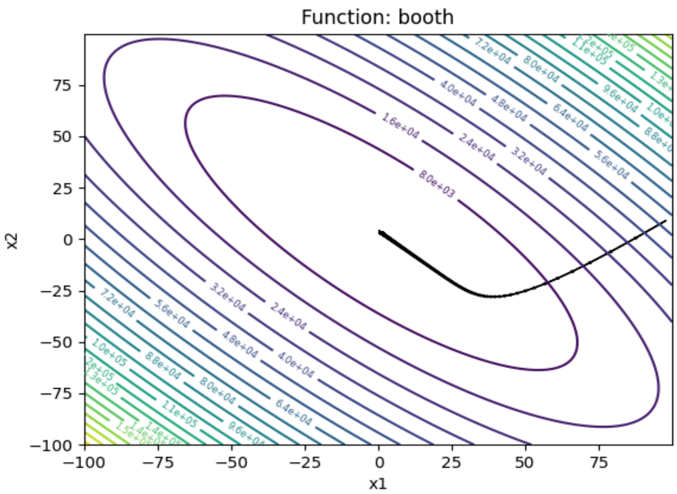
Treść zadania:

1. Zaimplementować metodę najszybszego wzrostu (minimalizacja, spodziewam się stałego współczynnika kroku, jeśli jednak ktoś chce zrobić więcej i zastosować zmienny współczynnik to ma taką możliwość). Gradient wyliczamy numerycznie.
2. Narysować zachowanie algorytmu (kolejne kroki algorytmu jako strzałki na tle poziomic funkcji celu). Uwaga: w praktycznych zadaniach optymalizacji nie da się narysować funkcji celu ponieważ zadania mają wiele wymiarów (np. 100), oraz koszt wyznaczenia oceny jednego punktu jest duży.
3. Zastosować metodę do znalezienia optimum funkcji booth w 2 wymiarach, po czym do znalezienia optimum funkcji o numerach od 1 do 3 z CEC 2017 w 10 wymiarach (na wykresie narysować kroki w wybranych 2 wymiarach z 10). Ograniczenia kostkowe przestrzeni to [-100, 100]. Uwzględnianie ograniczeń przez rzutowanie, np. dla x>100 x=100. Uwaga: wszystkie funkcje są unimodalne. Funkcje CEC są bardzo trudne, nie należy spodziewać się znalezienia optimum dla wszystkich. Należy jednak podjąć próby (z różnymi ustawieniami parametru beta).

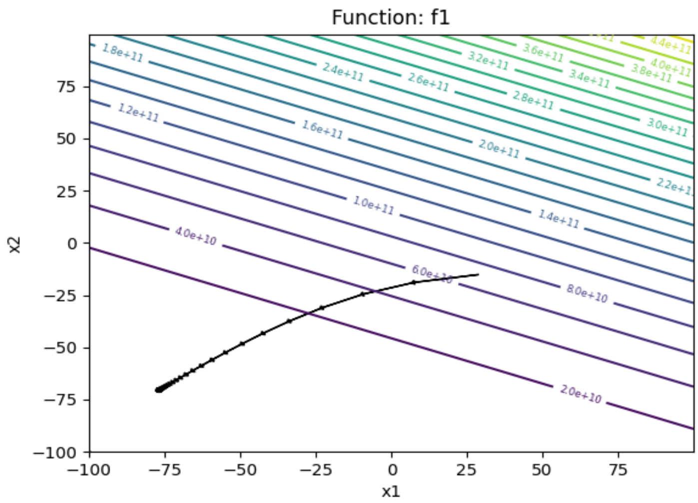
Wykresy ze znajdowania minimum dla odpowiednich funkcji:
Funkcja booth, gdzie beta = 0.01. Uzyskiwania wartość funkcji celu odpowiednio dla kolejnych wykresów: 2,36e-11, 1,58e-11, 2,33e-11

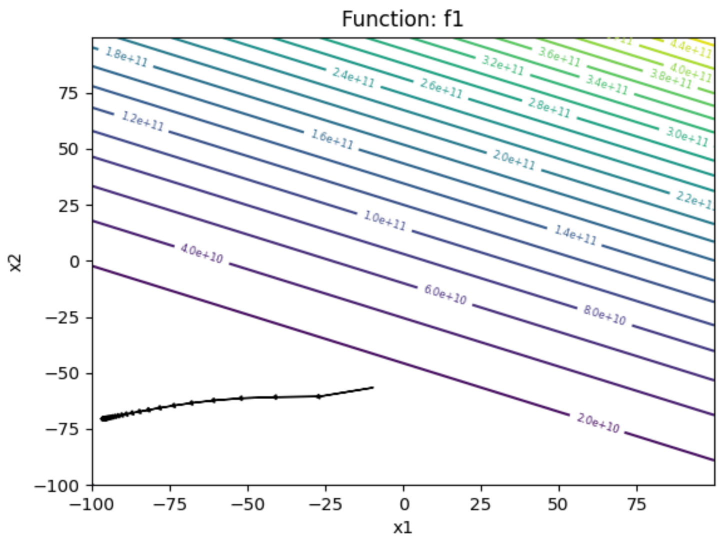


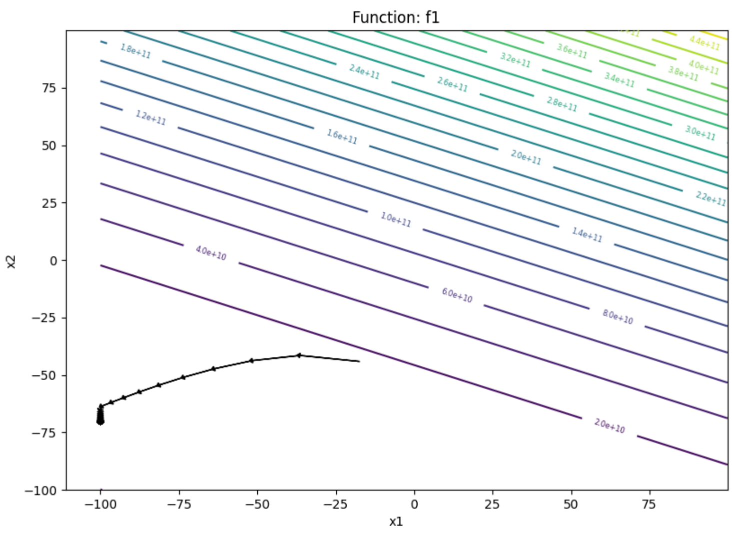




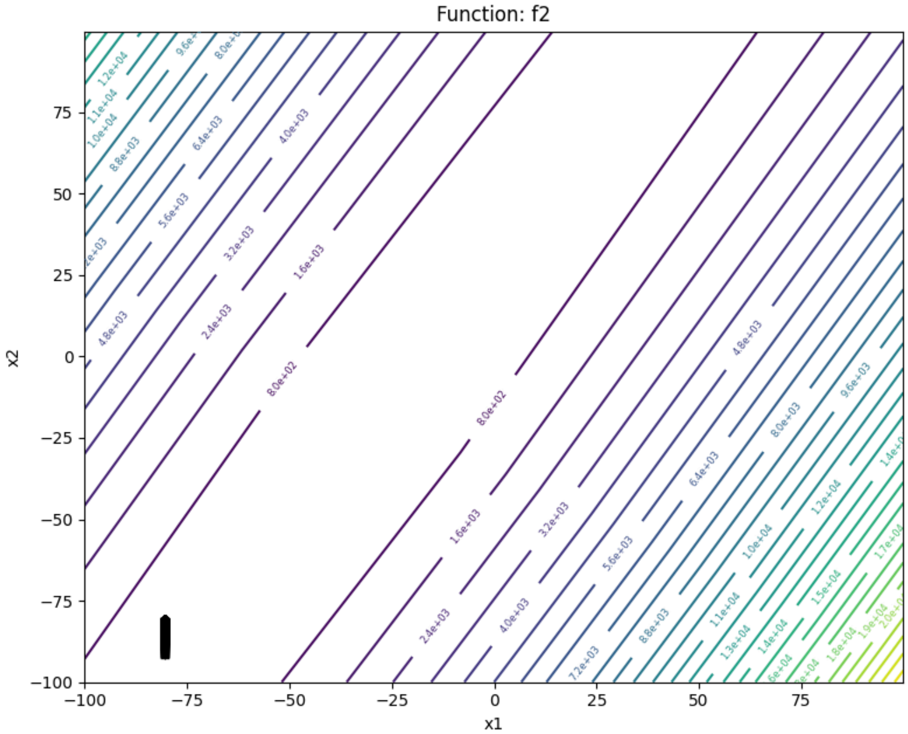
Funkcja f1 gdzie beta = 1e-08. Uzyskiwania wartość funkcji celu odpowiednio dla kolejnych wykresów: 400,02 1387,34 3678,32

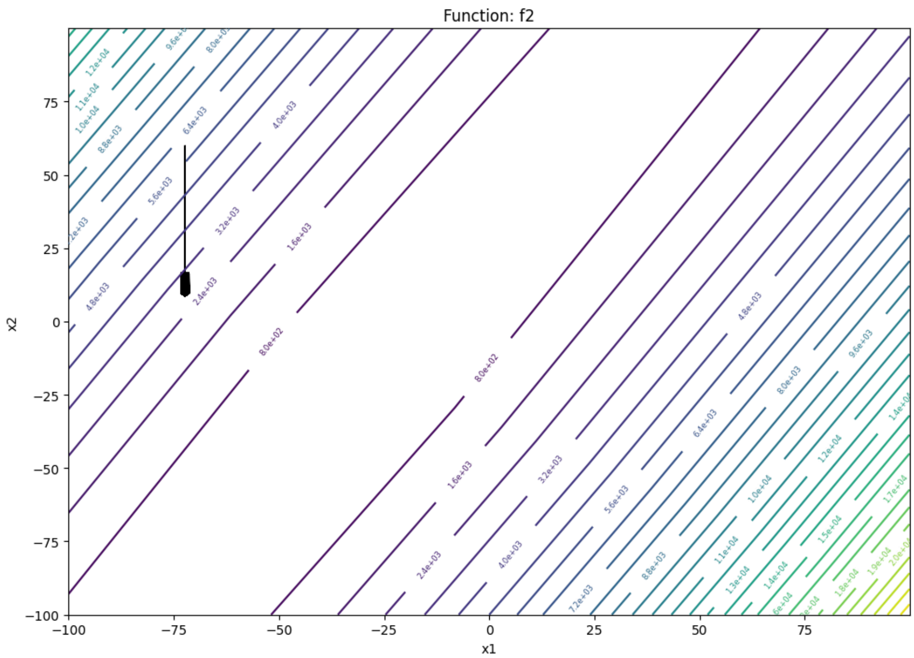


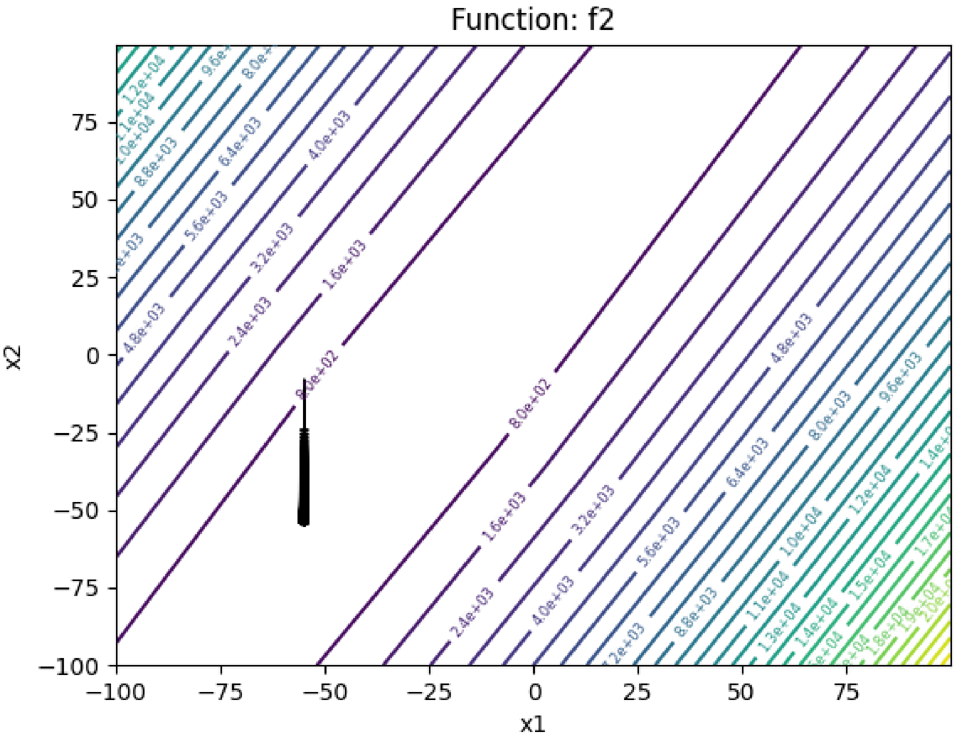




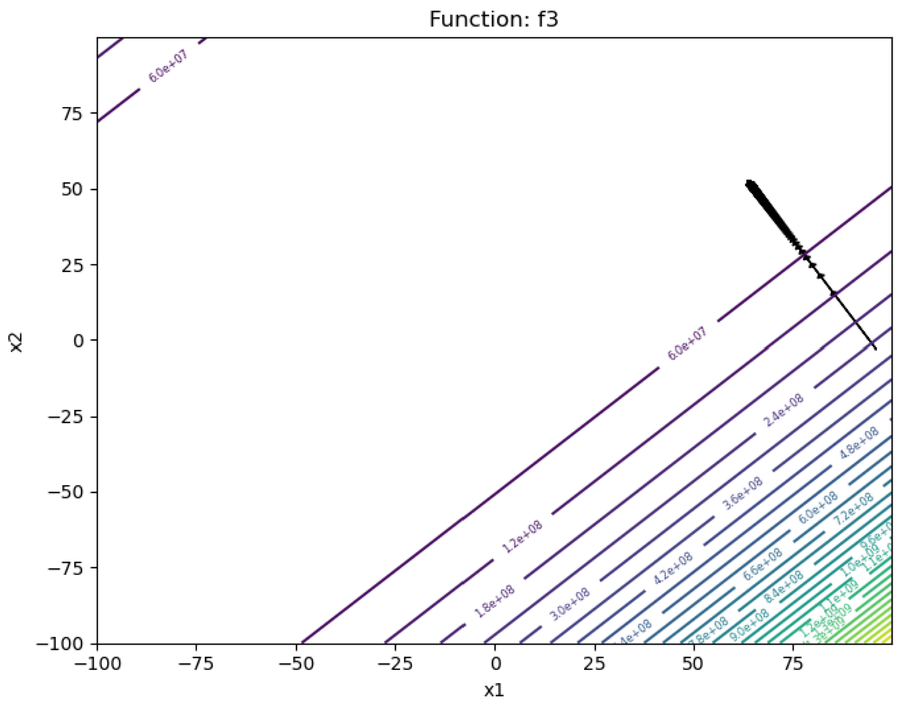
Funkcja f2 gdzie beta=2e-17. Uzyskiwania wartość funkcji celu odpowiednio dla kolejnych wykresów: 265, 2543, 432

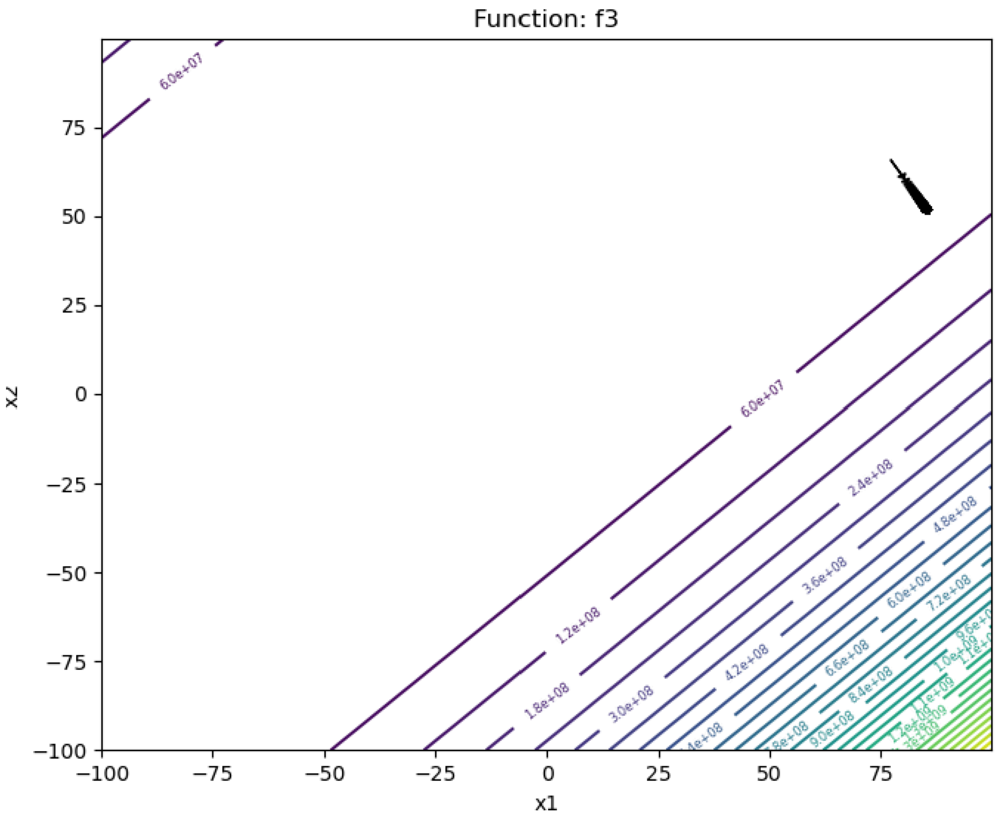


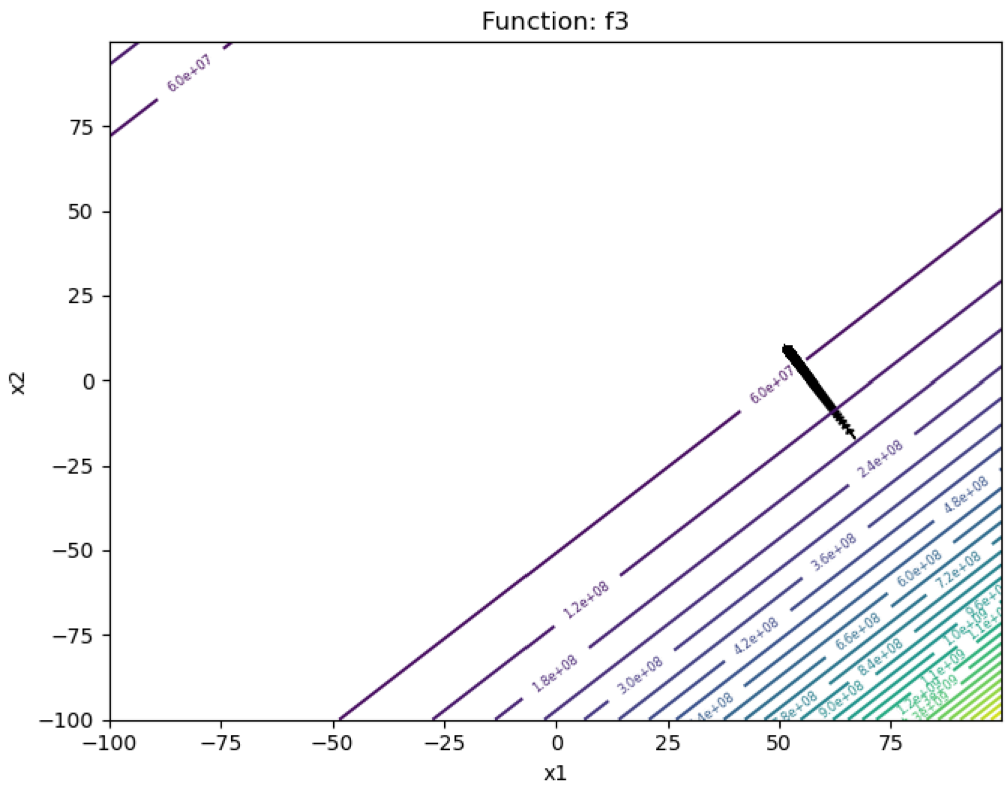




Funkcja f3 dla bety=5e-10. Uzyskiwania wartość funkcji celu odpowiednio dla kolejnych wykresów: 4,83e+06 6,6e+04 1,1e+05



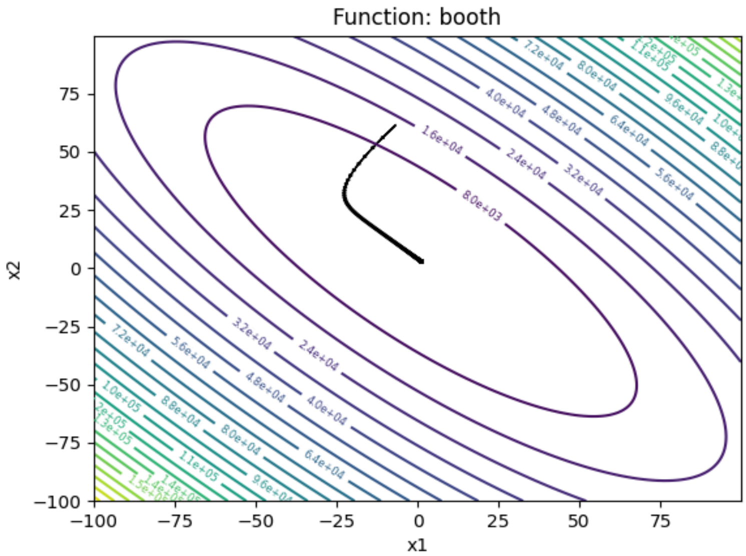

Function: f3
x1
x2
6.0e+07
1.2e+08
1.8e+08
2.4e+08
3.0e+08
3.6e+08
4.2e+08
4.8e+08
6.0e+08
6.6e+08
7.2e+08
8.4e+08
9.0e+08


Function: f3
x1
x2
6.0e+07
1.2e+08
1.8e+08
2.4e+08
3.0e+08
3.6e+08
4.2e+08
4.8e+08
6.0e+08
6.6e+08
7.2e+08
8.4e+08
9.0e+08

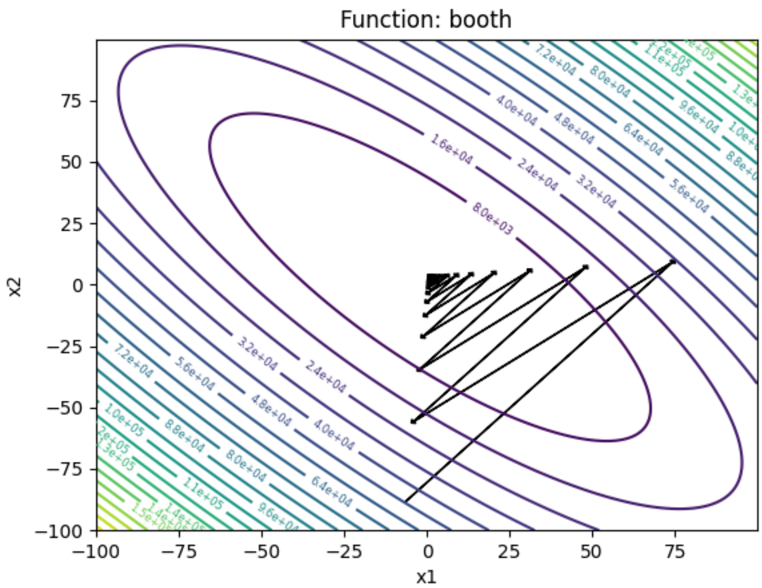
1. Jak wartość parametru beta wpływa na szybkość dojścia do optimum i zachowanie algorytmu? Jakiej bety użyto dla każdej z funkcji?

   Wartość parametru beta wpływa na szybkość docierania do minimum lokalnego funkcji. Czym mniejsza beta tym wolniej docieramy do minimum(robimy mniejsze kroki). A po przekroczeniu pewnej wartości tego współczynnika znalezienie minimum nie jest możliwe, bo powoduje to zbytnie oscylacje i rozbieżności. Dla przykładu znajdowanie minimum funkcji booth w zależności od parametru beta:

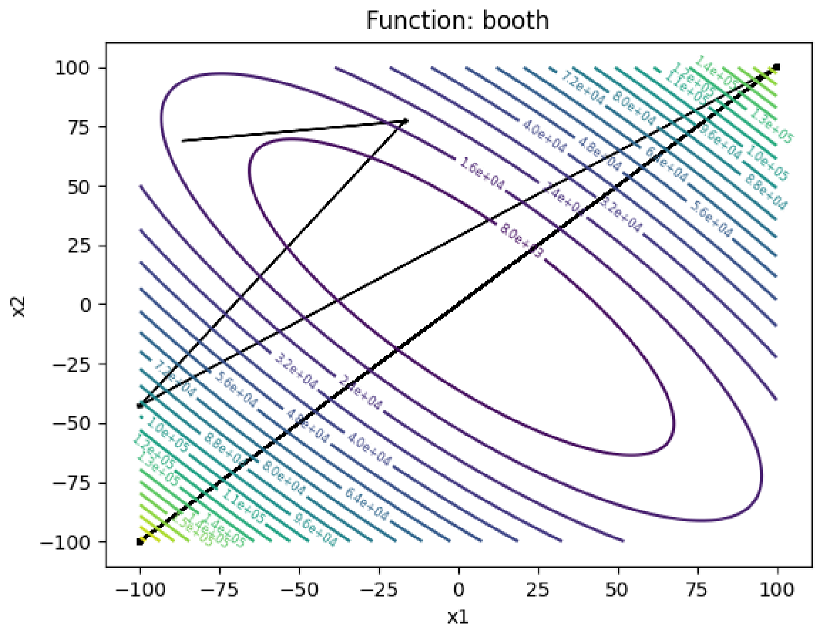
   beta = 0.01



beta = 0.1



beta = 0.2 Nieudana próba znalezienia minimum lokalnego - za duże oscylacje przy szukaniu optimum



2. Zalety/wady algorytmu?

   Zaletą algorytmu jest prosta implementacja, ale dla bardziej skomplikowanych funkcji (na przykład funkcje CEC) rzadko znajduje optimum lub punkt bliski optimum funkcji. Ponadto wadą jest również potrzeba testowania algorytmu dla wielu wartości beta, gdyż niektóre funkcje są bardzo wrażliwe na zmiany wartości współczynnika beta.

3. Wnioski

   Metoda gradientu prostego jest szybka i nieskomplikowana w implementacji, jednak działa ona przyzwoicie przy stałym współczynniku kroku jedynie dla niezbyt skomplikowanych funkcji. Przy używaniu tej metody należy również poświęcić sporo czasu do znalezienia dobrego współczynnika kroku, który nie będzie powodował zbytnich oscylacji.