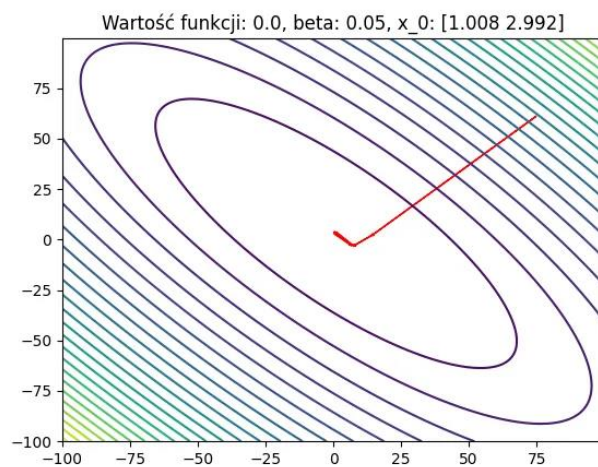
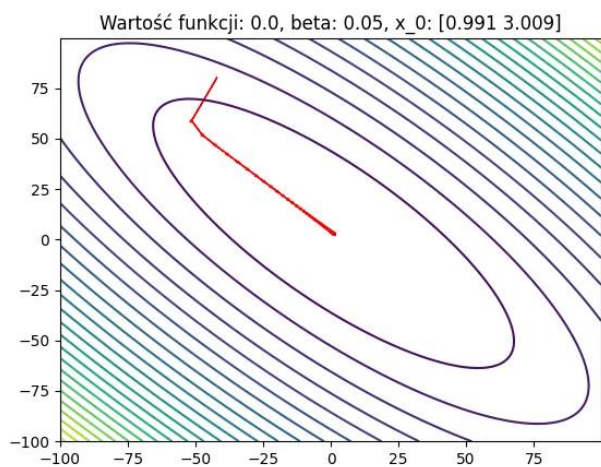
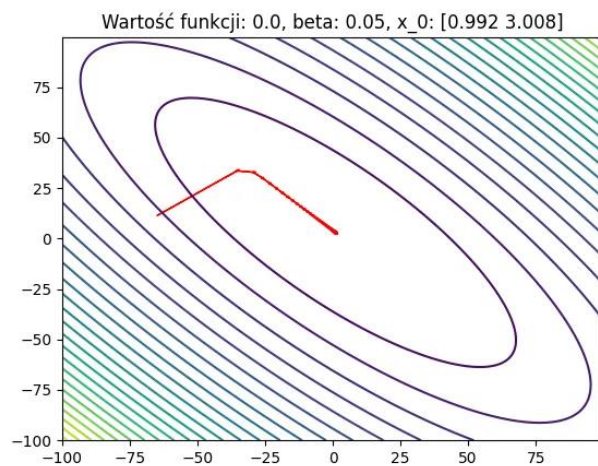
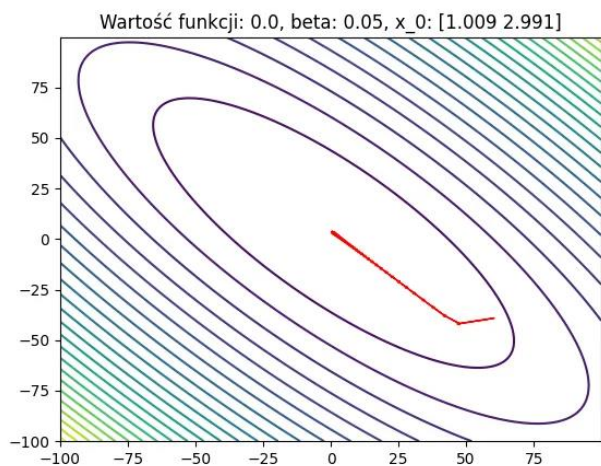


Wprowadzenie do sztucznej inteligencji - ćwiczenie 1, spotkanie 2

Kacper Marchlewicz

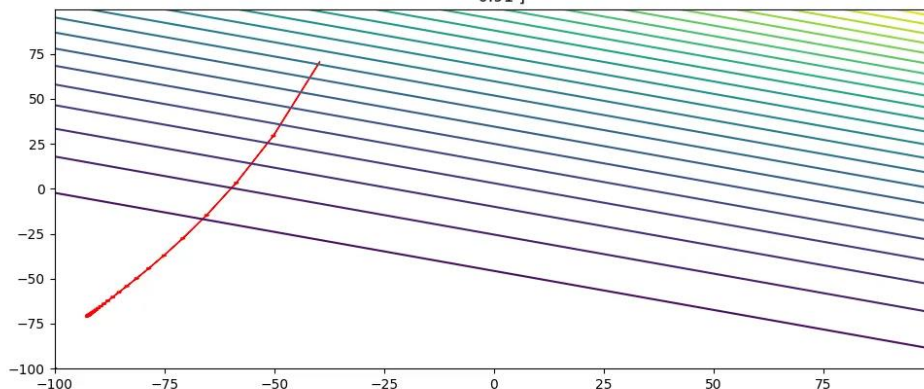
Przyjąłem za warunek kończący pętlę błąd równy 0,01.

Funkcja booth w 2 wymiarach:

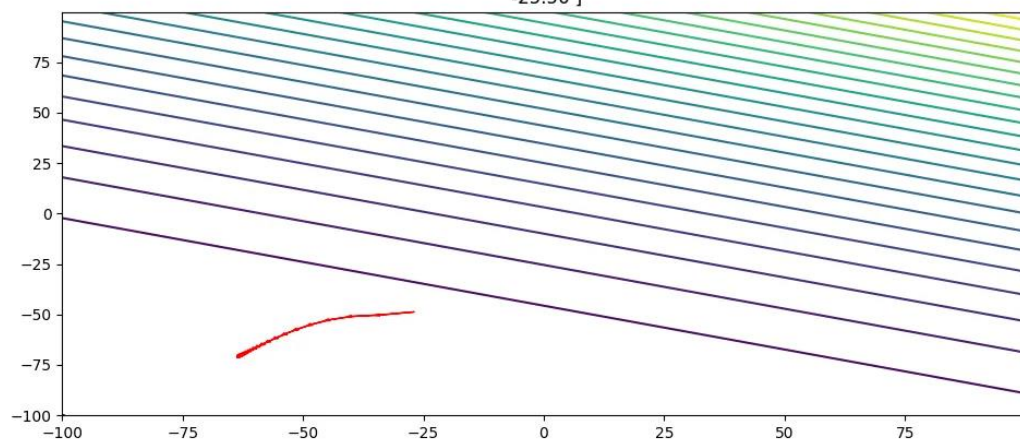


Funkcja f1:

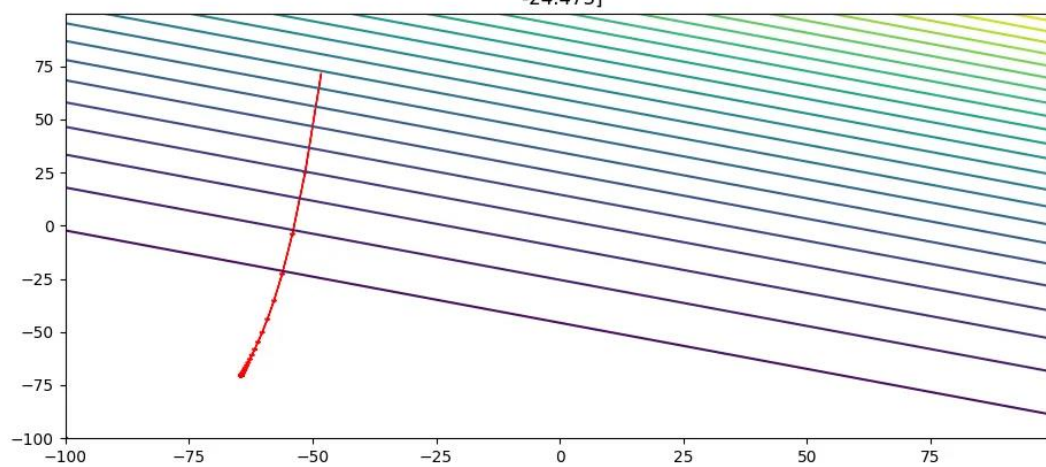
Wartość funkcji: 739387016356.78, beta: 1e-08, x_0: [-92.339 -70.427 -29.607 -58.323 22.089 59.938 -7.495 18.557 76.678 -0.91]



Wartość funkcji: 652289821415.68, beta: 1e-08, x_0: [-63.107 -70.428 -29.609 -58.327 22.089 59.935 22.527 18.558 76.679 -25.56]

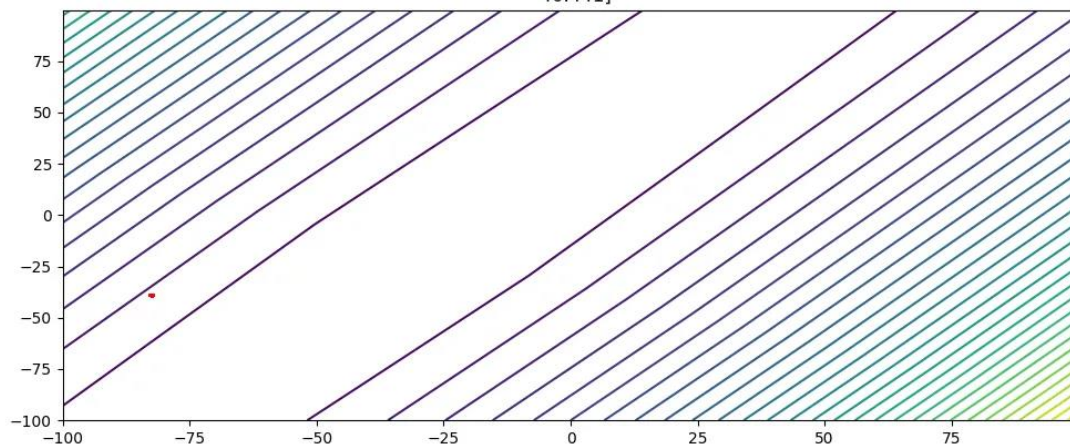


Wartość funkcji: 543505628159.41, beta: 1e-08, x_0: [-64.397 -70.427 -29.608 -58.323 22.088 59.94 21.203 18.561 76.678 -24.473]

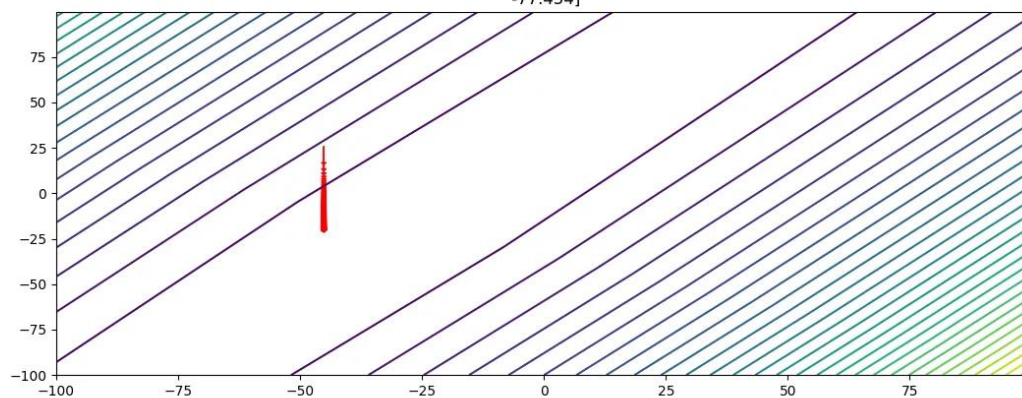


Funkcja f2

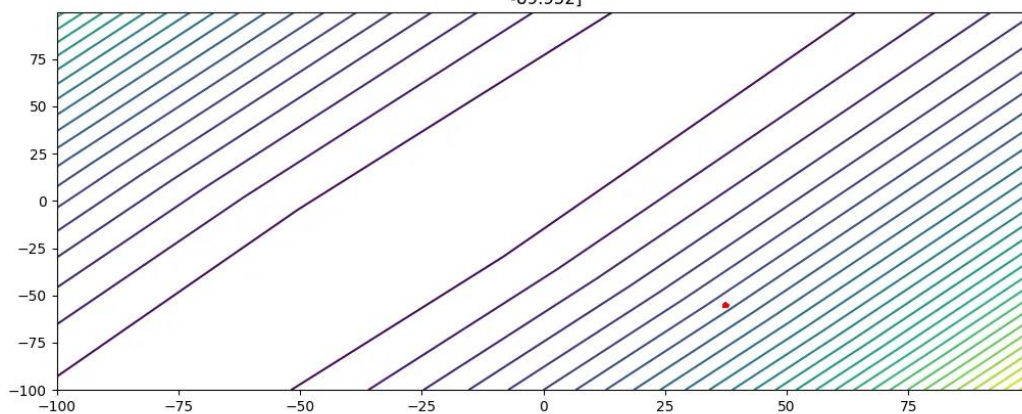
Wartość funkcji: 3.690898816184323e+17, beta: 1e-18, x_0: [-82.462 -38.829 44.826 -41.403 89.184 36.064 98.595 -10.147 10.913 40.441]



Wartość funkcji: 1.0321901588280331e+20, beta: 1e-18, x_0: [-45.14 -20.208 61.191 33.465 61.723 79.735 -75.567 33.501 5.982 -77.454]

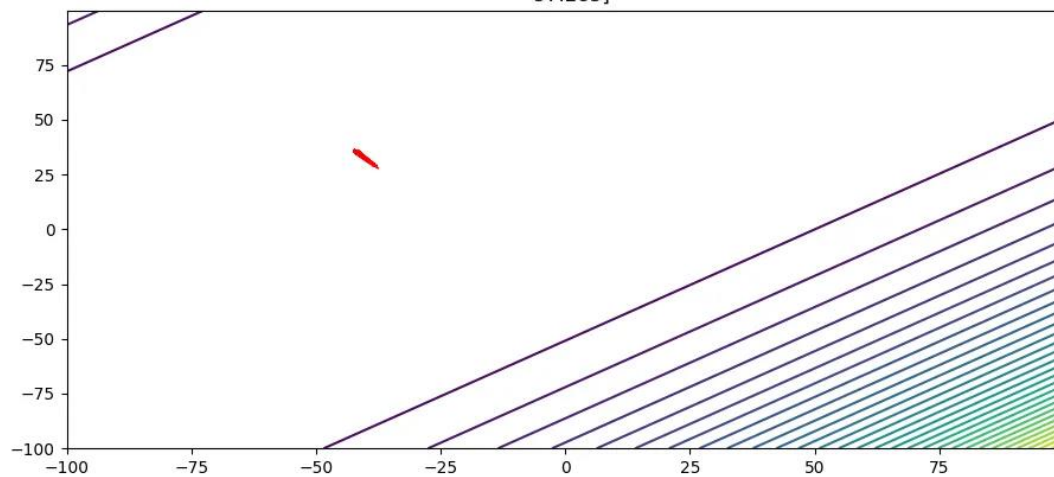


Wartość funkcji: 1.9263623799263762e+18, beta: 1e-18, x_0: [37.46 -55.142 6.159 70.619 -53.512 65.827 76.127 2.309 -72.623 -89.952]

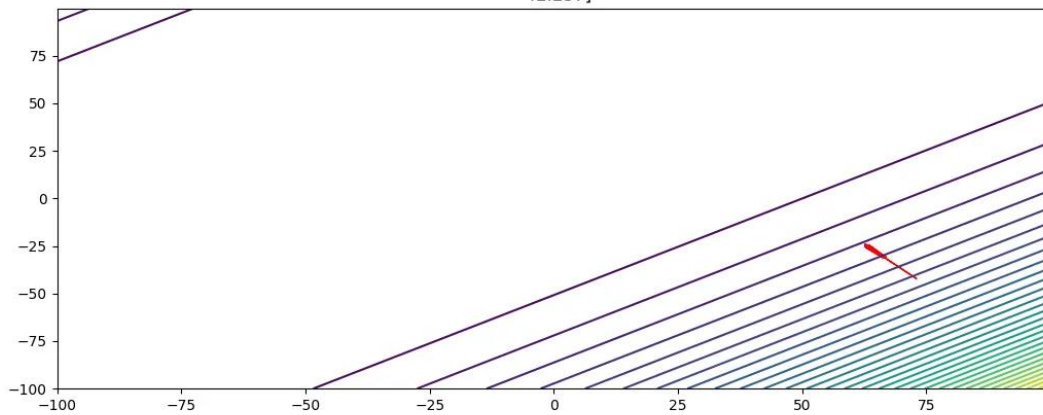


Funkcja f3

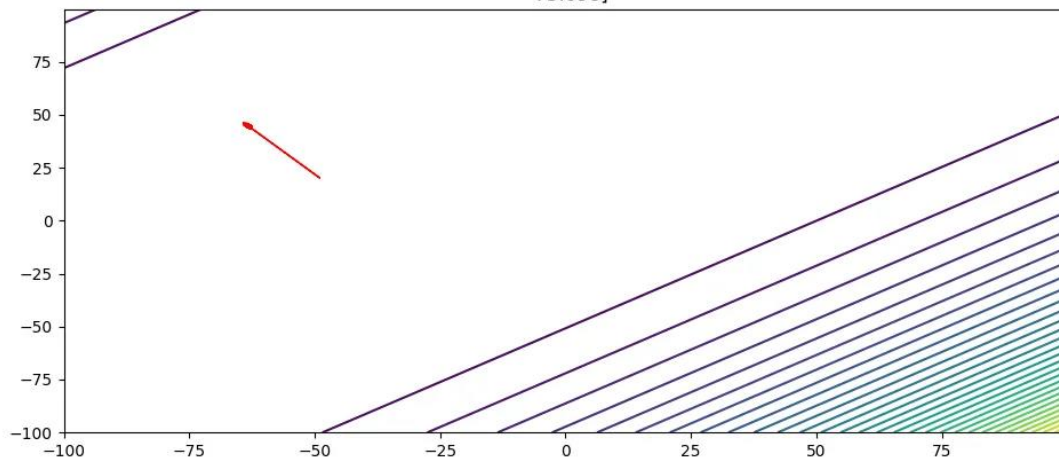
Wartość funkcji: 897670949.54, beta: 1e-08, x_0: [-41.893 35.311 -30.575 -74.137 -81.183 -34.512 60.014 -73.79 -56.868 57.285]



Wartość funkcji: 19005170793.28, beta: 1e-08, x_0: [63.177 -25.222 -80.02 -97.387 101.633 -9.619 -24.995 58.073 -16.908 42.237]



Wartość funkcji: 56708867978.63, beta: 1e-08, x_0: [-62.997 44.306 -13.705 38.531 102.831 59.062 20.745 -41.196 -8.208 75.098]



Pytania:

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

Zbyt mała wartość bety powoduje wydłużenie obliczeń – więcej pętli algorytmu. Gdy beta będzie zbyt duża algorytm nie zadziała poprawnie – wykonywane są zbyt duże skoki i zamiast zbliżyć się do optimum, oscyluje wokół niego.

Wartości beta: booth – 0,05 ; f1 – 10^{-8} ; f2 – 10^{-18} ; f3 – 10^{-8}

2. Zalety/wady algorytmu?

Zalety: Algorytm prosty do implementacji, działa dla funkcji wielowymiarowych, znajduje minima w umiarkowanym czasie przy dobrym korku

Wady: Przy źle dobranym kroku może nie odnaleźć minima, nie ma gwarancji znalezienia minima globalnego.

3. Wnioski

Algorytm dobrze radzi sobie z funkcją booth, zawsze znajdował minimum globalne. Dla funkcji z cec2017 raczej znajdował minima lokalne. Wynik zależał od punktów początkowych, co ma sens w przypadku funkcji 10-wymiarowych, które zapewne zawierają sporo minimów lokalnych.