

# Politechnika Warszawska

## Wprowadzenie do sztucznej inteligencji

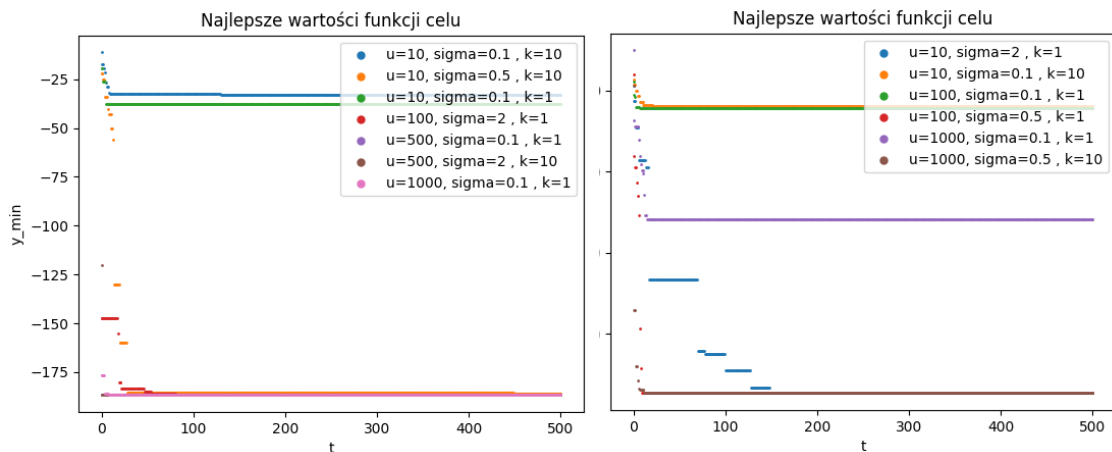
### Ćwiczenie 2

Zaimplementować algorytm ewolucyjny dla problemu minimalizacji funkcji n-zmiennych.  
Wykorzystać selekcję turniejową i sukcesję elitarną.

Przemysław Krasnodębski

Link do repozytorium: [WSI-21Z/Cwiczenie 2 at master · p-krasnodebski/WSI-21Z \(github.com\)](https://github.com/p-krasnodebski/WSI-21Z)

**Wyniki eksperymentów dla f-cji Schuberta:**  $(\sum_{i=1}^5 \cos((i+1)x_1 + i)) (\sum_{i=1}^5 \cos((i+1)x_2 + i))$



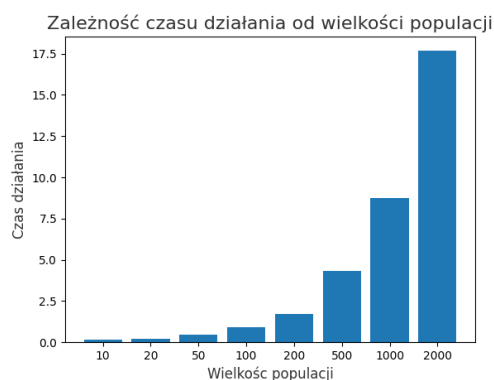
gdzie:  
 $u$  - liczba osobników  
 $\sigma$  - siła mutacji  
 $k$  - rozmiar elity  
 $t$  - iteracje

Rysunek 1. Funkcja celu dla różnych populacji z losowymi osobnikami populacji początkowej

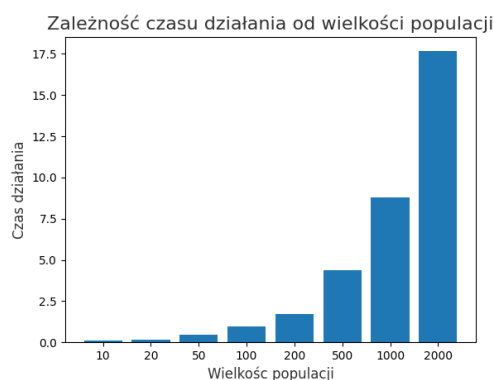
Rysunek 2. Funkcja celu dla różnych populacji z klonami w populacji początkowej

Na wykresach wyraźnie widać, że dla dobranych parametrów znacznie lepiej zachowywała się populacja zainicjowana różnymi losowymi osobnikami. Tylko jeden algorytm z rys. 1 utknął w minimum lokalnym. Natomiast dla algorytmów z rys. 2 aż połowa nie wykazała globalnego minimum, wartość optymalną osiągają algorytmy z większym wpływem mutacji osobników.

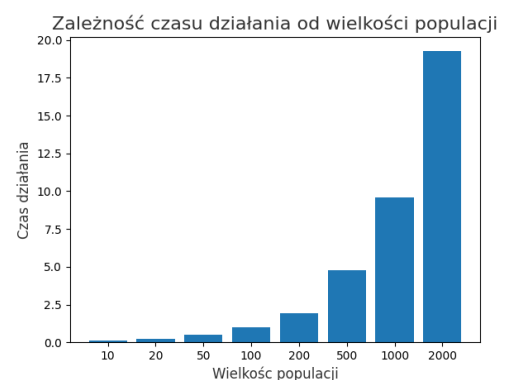
Powyższe wyniki przedstawiono dla liczby iteracji równej 500, dalsze zwiększanie liczby iteracji nie wniosło nic pozytywnego, a wydłużało czas działania algorytmu, lecz w przypadku bardziej skomplikowanych algorytmów warto przeprowadzić więcej iteracji, gdyż może to rozwiązać problem utknięcia algorytmu w minimum lokalnym.



Rysunek 3. Losowa populacja początkowa,  $\sigma = 0.5, k=1$

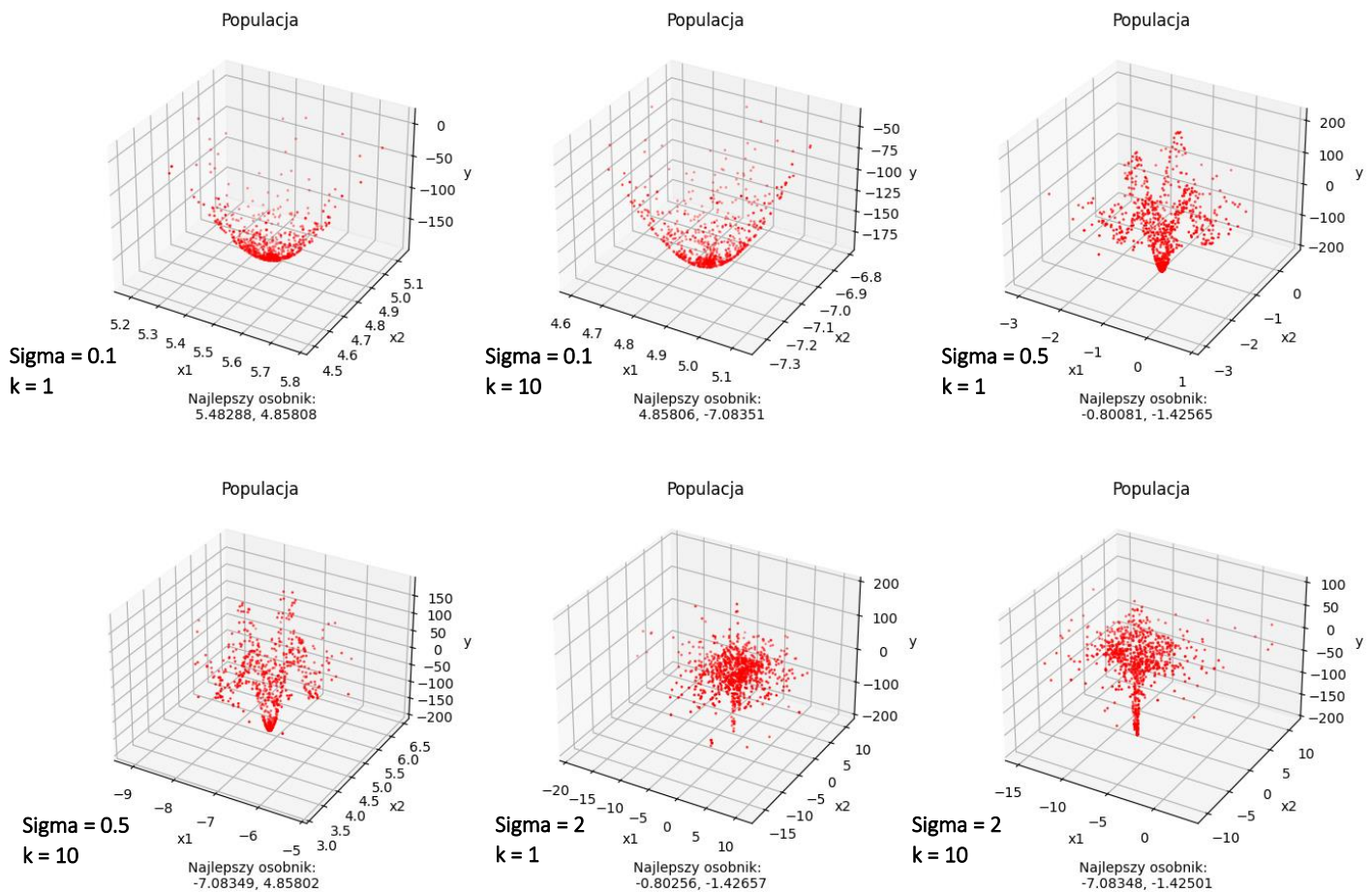


Rysunek 4. Populacja początkowa z klonów,  $\sigma = 0.5, k=1$



Rysunek 5. Losowa populacja początkowa,  $\sigma = 0.5, k=10$

Czas działania algorytmu znacznie wzrasta wraz ze wzrostem populacji. Natomiast różna konfiguracja parametrów ma pomijalny wpływ na długość działania.



Wykresy przedstawiają końcowe populacje po ewolucji algorytmu z różnymi parametrami wywołania. Wszystkie powyższe eksperymenty wykonano dla populacji liczącej 1000 osobników z losową populacją początkową. Mała siła mutacji powoduje, że algorytm działa w obrębie jednego minimum, rozmiar elity praktycznie nie ma znaczenia. Zwiększenie siły mutacji powoduje rozszerzenie zakresu populacji, zauważalny staje się wpływ rozmiaru elity. Populacja obejmuje znacznie więcej różnych punktów, lecz przy rozmiarze elity równym 1 wyznacza dokładnego minimum. Najlepszą populacją wydaje się być populacja o sigma równym 0.5 oraz rozmiarze elity równym 10.

## Wnioski ogólne

Algorytm z poprawnie zdefiniowanymi parametrami skutecznie rozwiązuje problem optymalizacji znalezienia minimum globalnego funkcji, lecz dobór parametrów nie jest zadaniem trywialnym dla początkowych użytkowników.

Zbyt mała siła mutacji może powodować zatrzymanie się algorytmu w minimum lokalnym. Podobny wpływ ma zbyt duży rozmiar elity w populacji. Takie połączenie tych dwóch parametrów zwiększa możliwości eksploatacji algorytmu.

Zwiększenie siły mutacji i zmniejszenie elity w populacji zwiększa natomiast możliwości eksploracji. Algorytm ma większą szansę znaleźć minimum globalne, lecz z mniejszą dokładnością.

Kluczowe jest wypośrodkowanie wspomnianych parametrów, w celu uzyskania optymalnego rozwiązania - globalnego minimum z satysfakcjonującym przybliżeniem.