

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Wstęp do Sztucznej Inteligencji
Semestr 24L
Sprawozdanie z ćwiczenia nr 2

Algorytmy ewolucyjne

Mikołaj Wewiór

Warszawa,
26 III 2024

1. Opis problemu

Celem ćwiczenia było zaimplementowanie algorytmu ewolucyjnego. Jest to jedna z podstawowych metod optymalizacji bezgradientowej, która o wiele lepiej znajduje optimum globalne dla zadania wielomodalnego, w porównaniu do poprzednio implementowanej metody gradientu prostego. Działanie algorytmu oparte jest na populacji punktów, które poddawane są różnym operacjom. Do podstawowych elementów należą:

- funkcja dopasowania - każdemu osobnikowi w populacji przypisywana jest ranga, najlepsze punkty otrzymują najniższe wartości, a gorsze większe;
- reprodukcja - z populacji macierzystej losowana jest nowa, tzw. "prześciowa" populacja, której osobniki poddawane są kolejnym operacjom;
- krzyżowanie - część punktów jest ze sobą krzyżowanych, co za tym idzie, pewne cechy - w tym przypadku położenie - jest odpowiednią sumą dwóch punktów rodziców ;
- mutacja - do osobnika dodawana jest losowa perturbacja, która zmienia jego położenie.

W implementacji tej przyjęto następujące założenia:

- funkcja dopasowania - odległość punktu od obecnie najlepszego punktu w populacji;
- reprodukcja - realizowana jako turniej dwuosobowy - daje to szansę gorszym punktom przetrwać do kolejnych iteracji, co często pozwala znaleźć optima poza bezpośrednim otoczeniem obecnie najlepszego punktu;
- krzyżowanie - losowa kombinacja współrzędnych z dwóch rodzicielskich osobników;
- mutacja - dodanie do osobników wartości losowanej z rozkładu normalnego o wartości oczekiwanej 0, oraz odchyleniu σ , którego wartość ustala się ręcznie przed uruchomieniem działania algorytmu.

2. Badane funkcje

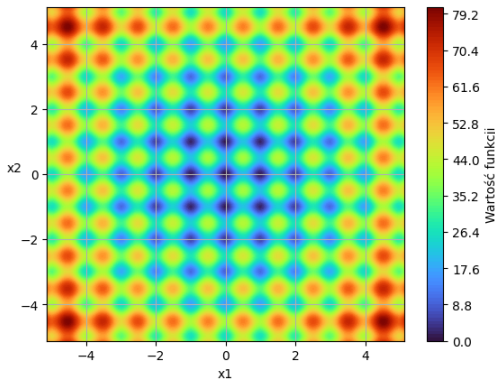
Implementacja oraz analiza zostały przeprowadzone w dwóch wymiarach ($d = 2$) dla dwóch funkcji: Rastringa oraz Griewanka. Dane są one następującymi wzorami:

Rastrigin:

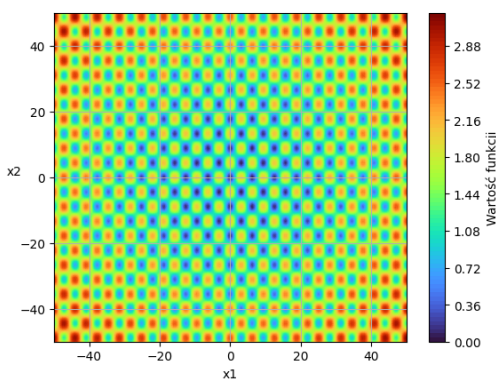
$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)],$$

Griewank:

$$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$



(a) Rastrigin

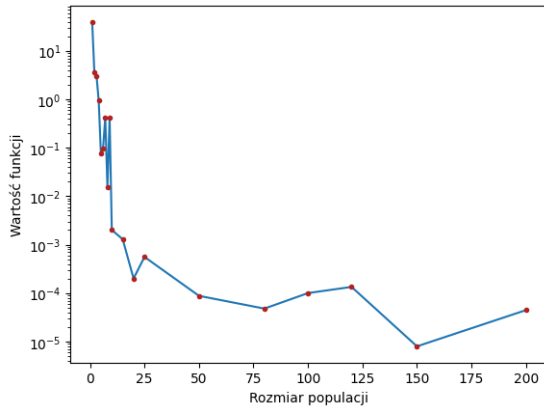


(b) Griewank

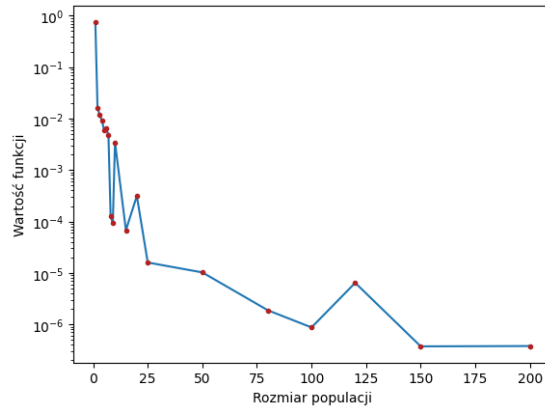
3. Badanie wpływu parametrów

Dla każdego badania testy zostały uruchomione 5 razy, a następnie wyciągnięto średnie wartości i przedstawiono poniżej. Hiperparametry algorytmu: liczba iteracji = 200, rozmiar populacji = 25, $\sigma = 0.5$ dla funkcji Rastrigina oraz $\sigma = 2.0$ dla funkcji Griewanka, prawdopodobieństwo mutacji = 0.4, prawdopodobieństwo krzyżowania = 0.6, liczba sukcesji elitarnych = 1.

3.1. Rozmiar populacji



(a) f-cja Rastrigina

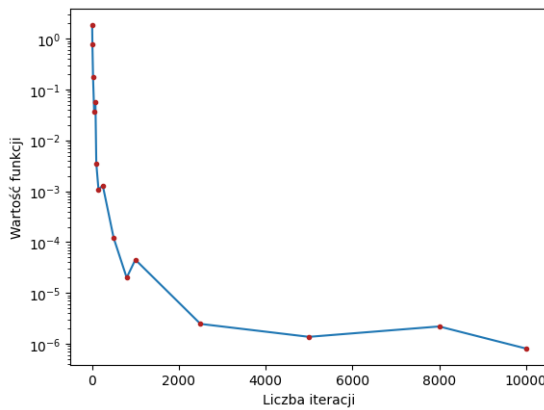


(b) f-cja Griewanka

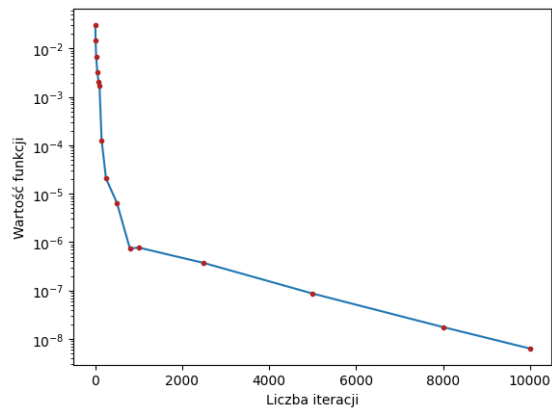
rozmiar populacji	1	2	5	10	50	100	200
Rastrigin	39,37	3,64	$7,59 \cdot 10^{-2}$	$2,02 \cdot 10^{-3}$	$8,79 \cdot 10^{-5}$	$1,00 \cdot 10^{-4}$	$4,46 \cdot 10^{-5}$
Griewank	0,76	$1,60 \cdot 10^{-2}$	$5,97 \cdot 10^{-3}$	$3,46 \cdot 10^{-3}$	$1,03 \cdot 10^{-5}$	$8,73 \cdot 10^{-7}$	$3,798 \cdot 10^{-7}$

Tab. 1: Błąd absolutny wartości funkcji dopasowania dla różnych rozmiarów populacji

3.2. Liczba iteracji



(a) f-cja Rastrigina

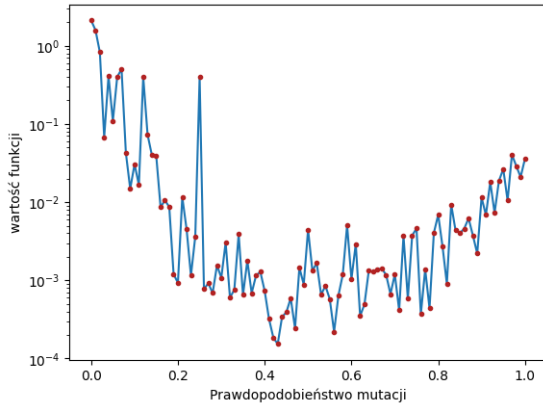


(b) f-cja Griewanka

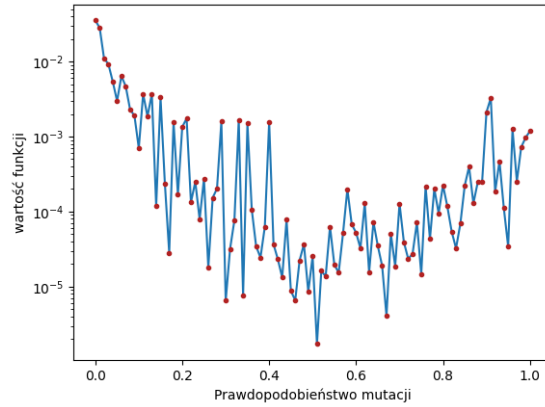
liczba iteracji	5	50	250	500	1000	5000	10 000
Rastrigin	1,86	$3,69 \cdot 10^{-2}$	$1,29 \cdot 10^{-3}$	$1,21 \cdot 10^{-4}$	$4,51 \cdot 10^{-5}$	$1,37 \cdot 10^{-6}$	$8,08 \cdot 10^{-7}$
Griewank	0,03	$3,28 \cdot 10^{-3}$	$2,06 \cdot 10^{-5}$	$6,53 \cdot 10^{-6}$	$7,74 \cdot 10^{-7}$	$8,69 \cdot 10^{-8}$	$6,40 \cdot 10^{-9}$

Tab. 2: Błąd absolutny wartości funkcji dopasowania dla różnych rozmiarów populacji

3.3. Prawdopodobieństwo mutacji



(a) f-cja Rastrigina

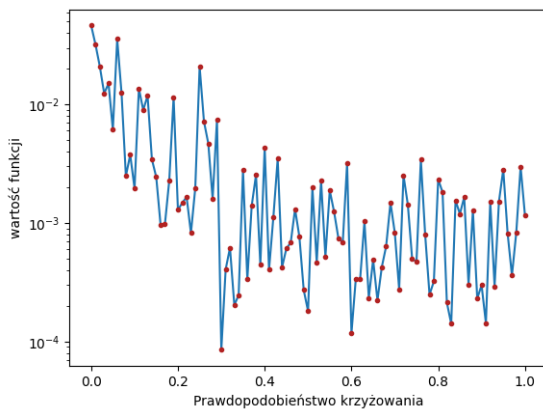


(b) f-cja Griewanka

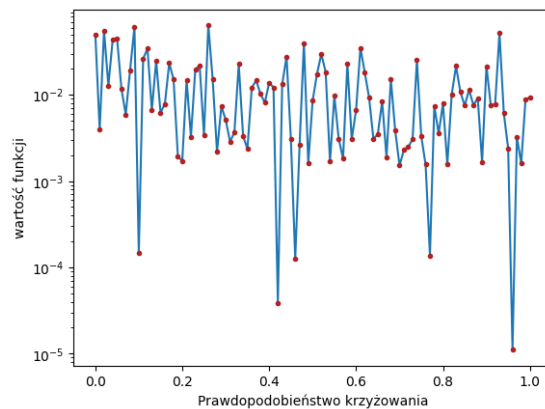
p-stwo mutacji [%]	0	1	20	40	60	80	100
Rastrigin	2,10	1,58	$9,29 \cdot 10^{-4}$	$7,31 \cdot 10^{-4}$	$1,04 \cdot 10^{-3}$	$6,90 \cdot 10^{-3}$	$3,58 \cdot 10^{-2}$
Griewank	$3,52 \cdot 10^{-2}$	$2,81 \cdot 10^{-2}$	$1,34 \cdot 10^{-3}$	$1,58 \cdot 10^{-3}$	$5,24 \cdot 10^{-5}$	$2,25 \cdot 10^{-4}$	$9,64 \cdot 10^{-4}$

Tab. 3: Błąd absolutny wartości funkcji dopasowania dla różnych rozmiarów populacji

3.4. Prawdopodobieństwo krzyżowania



(a) f-cja Rastrigina

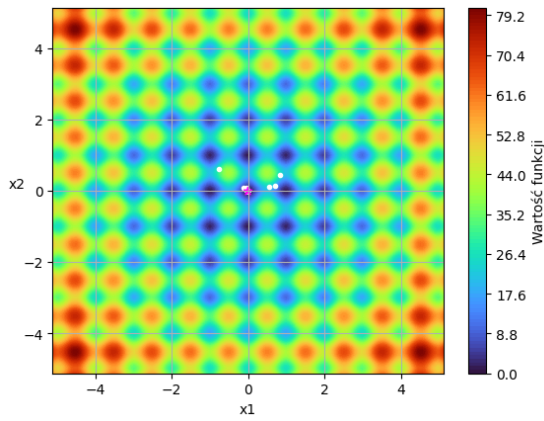


(b) f-cja Griewanka

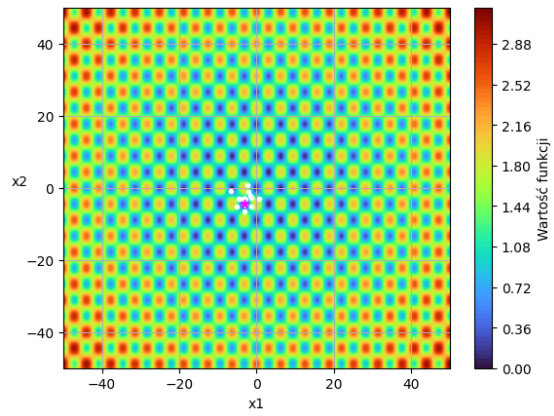
p-stwo krzyżowania [%]	0	1	20	40	60	80	100
Rastrigin	$4,63 \cdot 10^{-2}$	$3,19 \cdot 10^{-2}$	$1,29 \cdot 10^{-3}$	$4,34 \cdot 10^{-3}$	$1,17 \cdot 10^{-4}$	$2,33 \cdot 10^{-3}$	$1,16 \cdot 10^{-3}$
Griewank	$4,98 \cdot 10^{-2}$	$3,97 \cdot 10^{-3}$	$1,72 \cdot 10^{-3}$	$1,38 \cdot 10^{-2}$	$6,66 \cdot 10^{-3}$	$7,93 \cdot 10^{-3}$	$9,24 \cdot 10^{-3}$

Tab. 4: Błąd absolutny wartości funkcji dopasowania dla różnych rozmiarów populacji

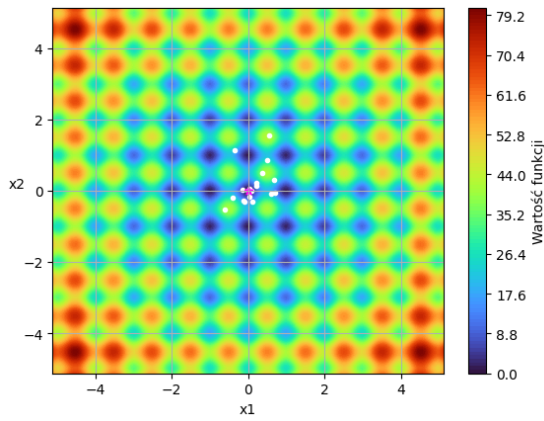
3.5. Wizualizacje



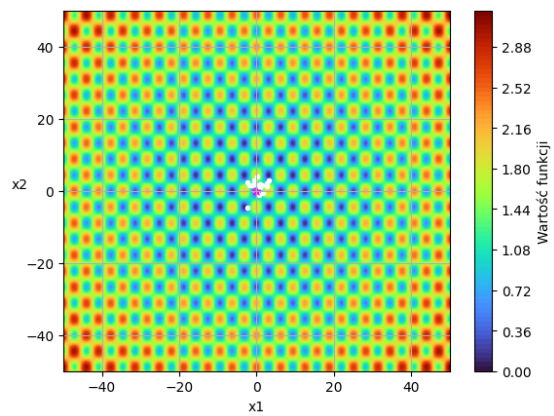
(a) p-stwo mutacji = 0,1



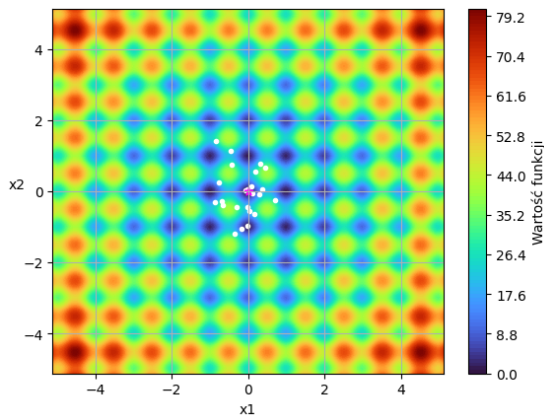
(d) p-stwo mutacji = 0,1



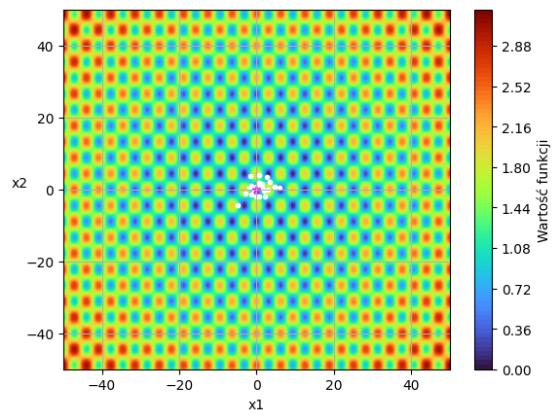
(b) p-stwo mutacji = 0,5



(e) p-stwo mutacji = 0,5



(c) p-stwo mutacji = 0,9



(f) p-stwo mutacji = 0,9

Widać, że wraz z zwiększającym się prawdopodobieństwem mutacji, rośnie rozproszenie osobników wokół optimum.

4. Podsumowanie

Z wykresów widać, że im większa jest liczba iteracji tym dokładniejsze rozwiązanie jesteśmy w stanie otrzymać. Taka sama zależność jest dla rozmiaru populacji. Dlatego warto przyjmować dłuższe czasy trwania algorytmu i większe populacje. Wiąże się to jednak z wydłużonym czasem działania programu. Przy czym zazwyczaj mamy ograniczony budżet - FES, liczbę ewaluacji - dlatego nie możemy dowolnie zwiększać tych dwóch parametrów.

Optymalne wartości prawdopodobieństwa mutacji oraz krzyżowania znajdują się w okolicy środka przedziału - między 0,3 a 0,8. Nie można jednak wskazać konkretnej najlepszej wartości, ale wybór z powyższego przedziału powinien okazać się wystarczająco skuteczny, aby otrzymać satysfakcjonujące wyniki.

W porównaniu do metody gradientu prostego, zauważyć można gigantyczną przewagę metod ewolucyjnych (jeżeli parametry są ustawione w odpowiedni sposób) - w znacznej większości uruchomień są one w stanie znaleźć optimum globalne. Przy czym często zajmuje to więcej czasu, a sama dokładność jest znacznie mniejsza.

Dokładność mogłoby poprawić wprowadzenie adaptacyjnej zmiany wartości σ oraz prawdopodobieństwa mutacji. Na początku trwania algorytmu wartości te mogłyby być duże, co sprawiłoby, że algorytm zajął by się przede wszystkim eksploracją. Z kolejnymi iteracjami wartości te powinny maleć, żeby algorytm przeszedł w eksploatację znalezionego optimum. Takie podejście pozwala znaleźć optimum z większą dokładnością, jednak narażone jest na wpadanie do optimum lokalnych.