

Metody Inteligencji Obliczeniowej
Laboratorium 7

Algorytmy genetyczne

Kamil Pyla

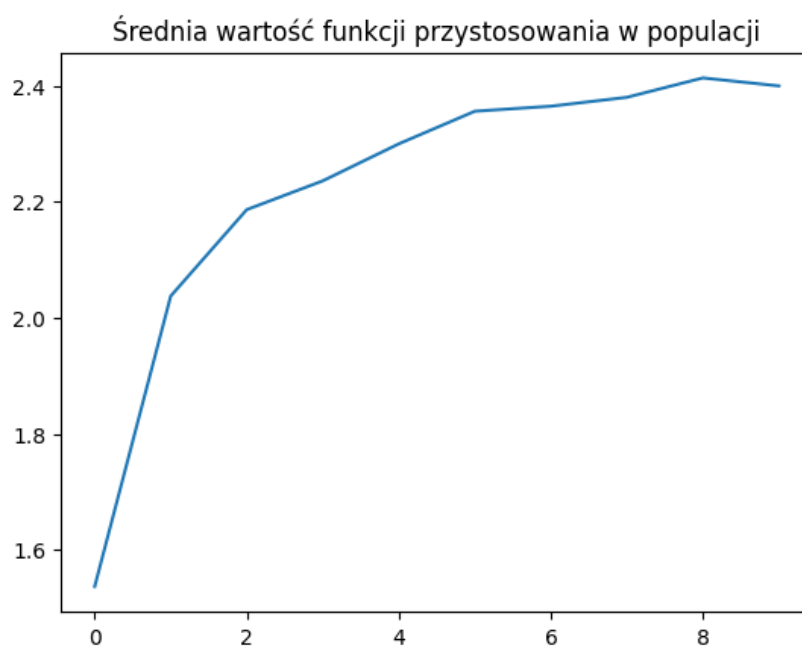


Wyniki zadania 1:

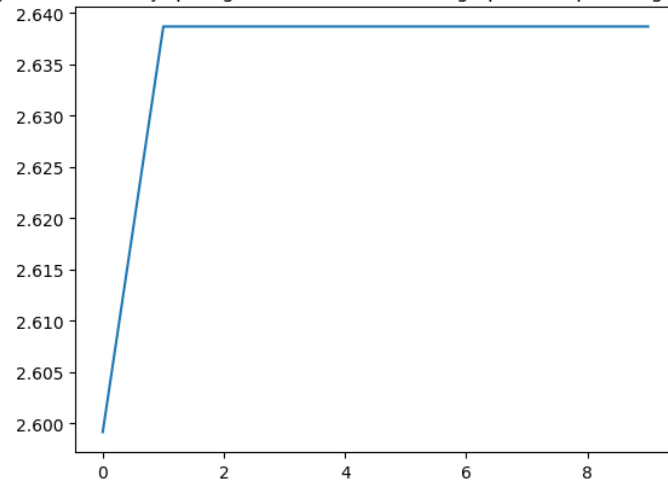
prawdopodobieństwo mutacji	sposób kodowania	znaleziony wynik	błąd względny [%]
0.0	binarne	2.638696798747 9553	0.185
0,1	binarne	2.641334399527 739	0.063
0,5	binarne	2.643538787104 327	0.00148
1	binarne	2.6435611360171 13	0.00064
0	gray	2.638696798747 9553	0.185
0,1	gray	2.641825476501 2477	0.066
0,5	gray	2.643565989519 7006	0.00045

1	gray	2.6435611360171 13	0.00064
---	------	-----------------------	---------

1. Wyniki graficzne dla prawdopodobieństwa mutacji: 0, kodowanie: binarne

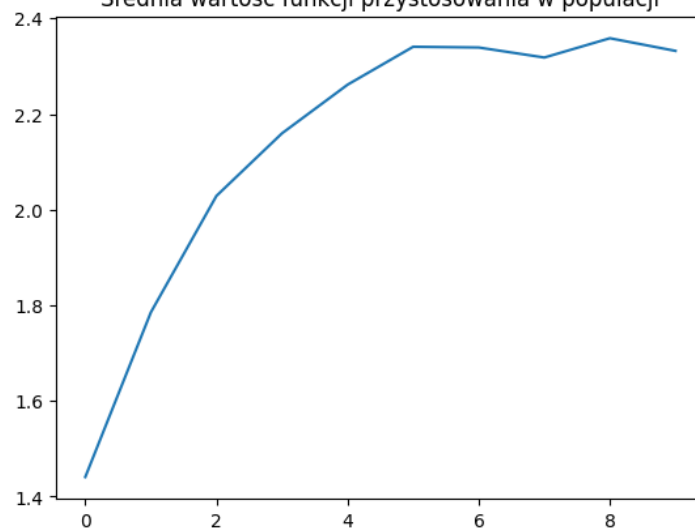


Przystosowanie najlepszego osobnikazonego podczas przebiegu algorytmu

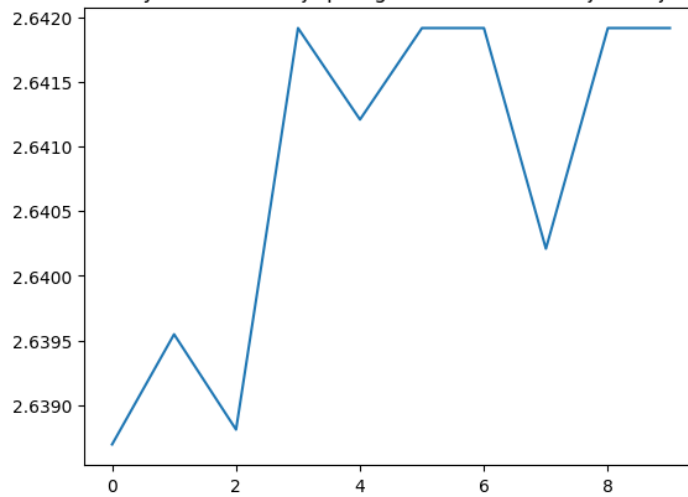


2. Wyniki graficzne dla prawdopodobieństwa mutacji: 0.1, kodowanie: binarne

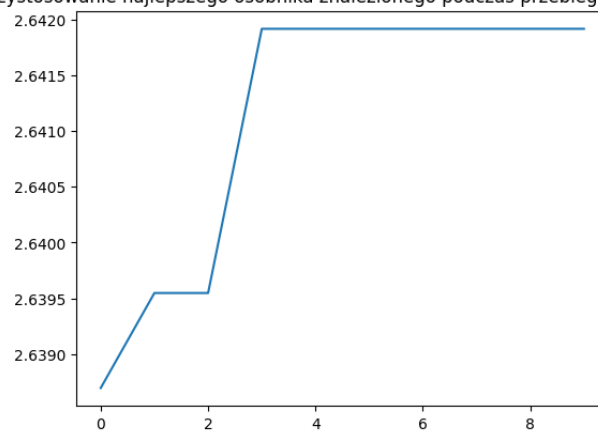
Średnia wartość funkcji przystosowania w populacji



Przystosowanie najlepszego osobnika w każdej iteracji

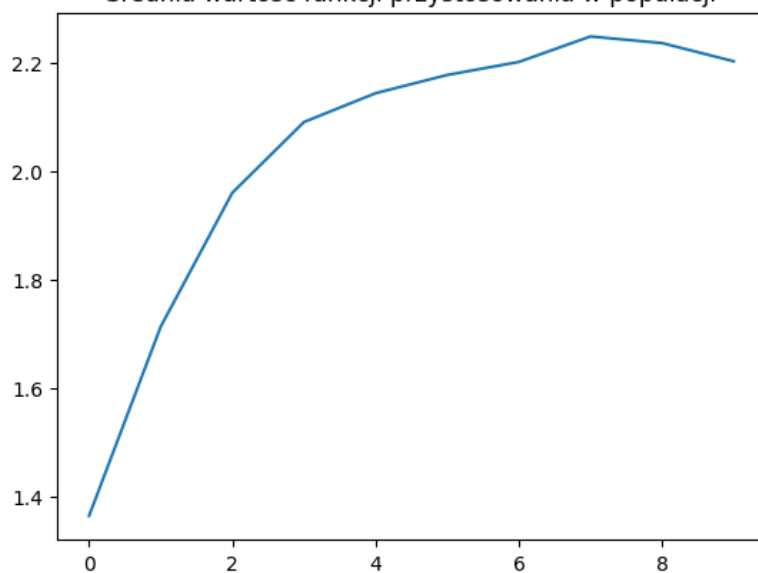


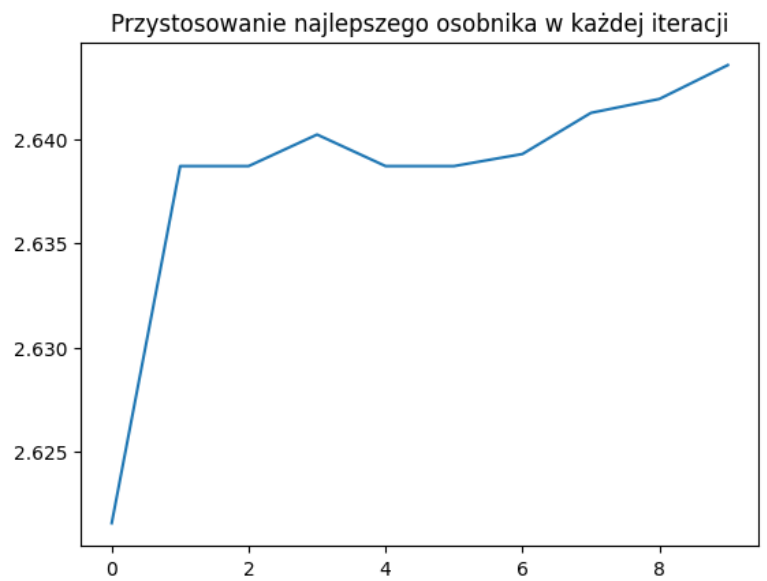
Przystosowanie najlepszego osobnika znalezione podczas przebiegu algorytmu



3. Wyniki graficzne dla prawdopodobieństwa mutacji: 0,5 kodowanie:
binarne

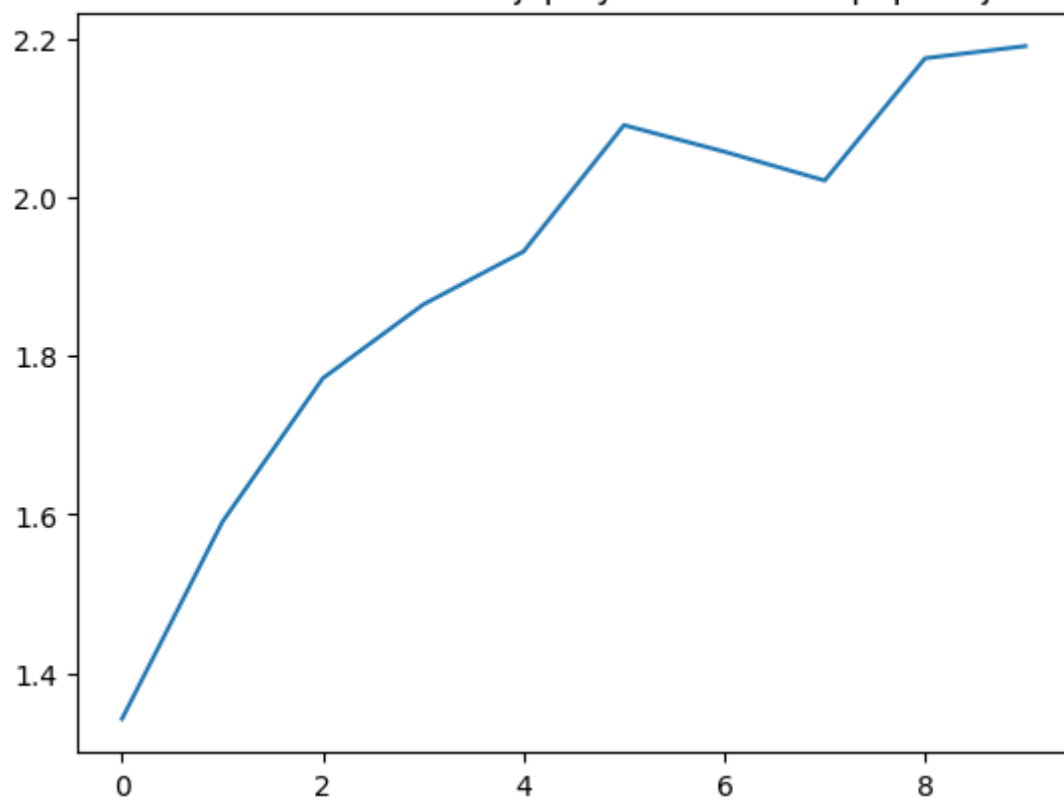
Średnia wartość funkcji przystosowania w populacji





4. Wyniki graficzne dla prawdopodobieństwa mutacji: 0 kodowanie: graya:

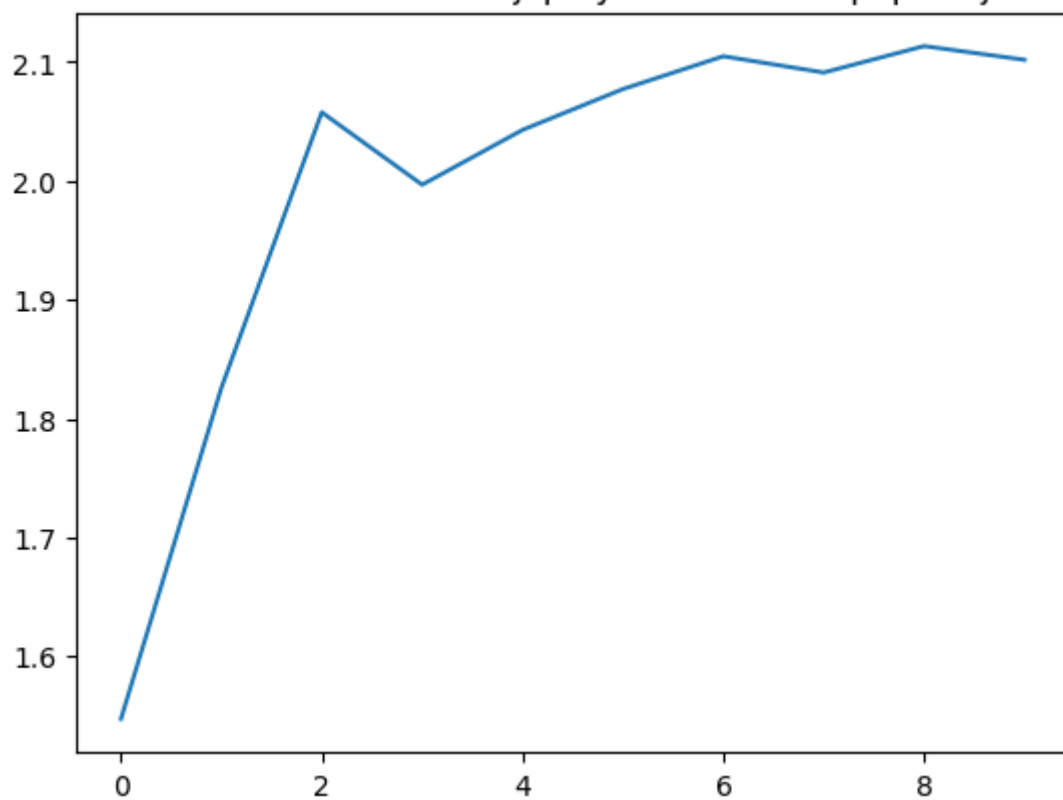
Średnia wartość funkcji przystosowania w populacji





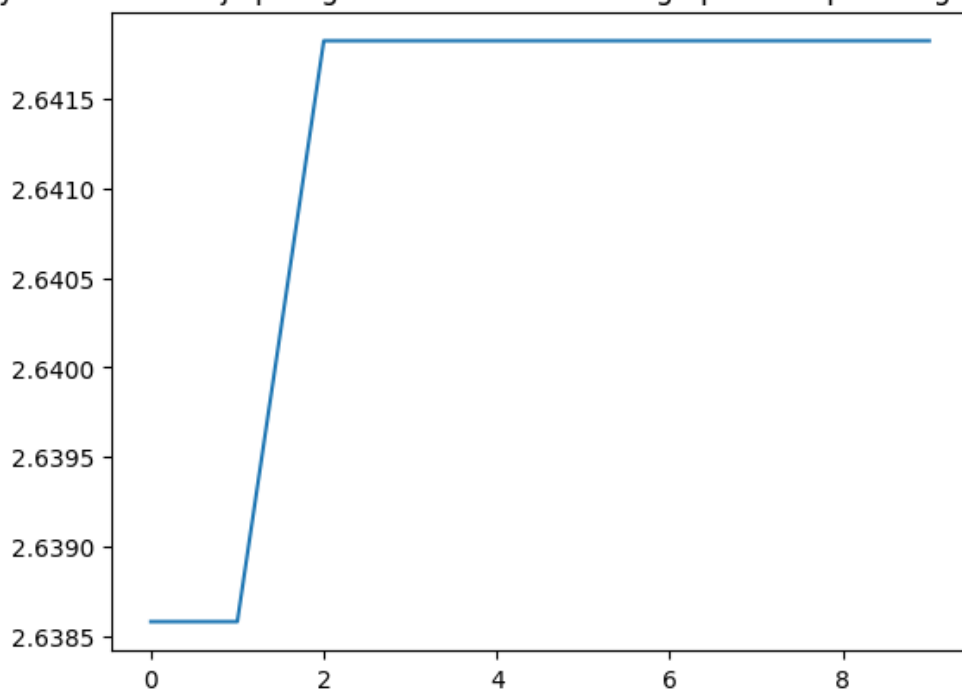
5. Wyniki graficzne dla prawdopodobieństwa mutacji: 0,1 kodowanie:
graya:

Średnia wartość funkcji przystosowania w populacji



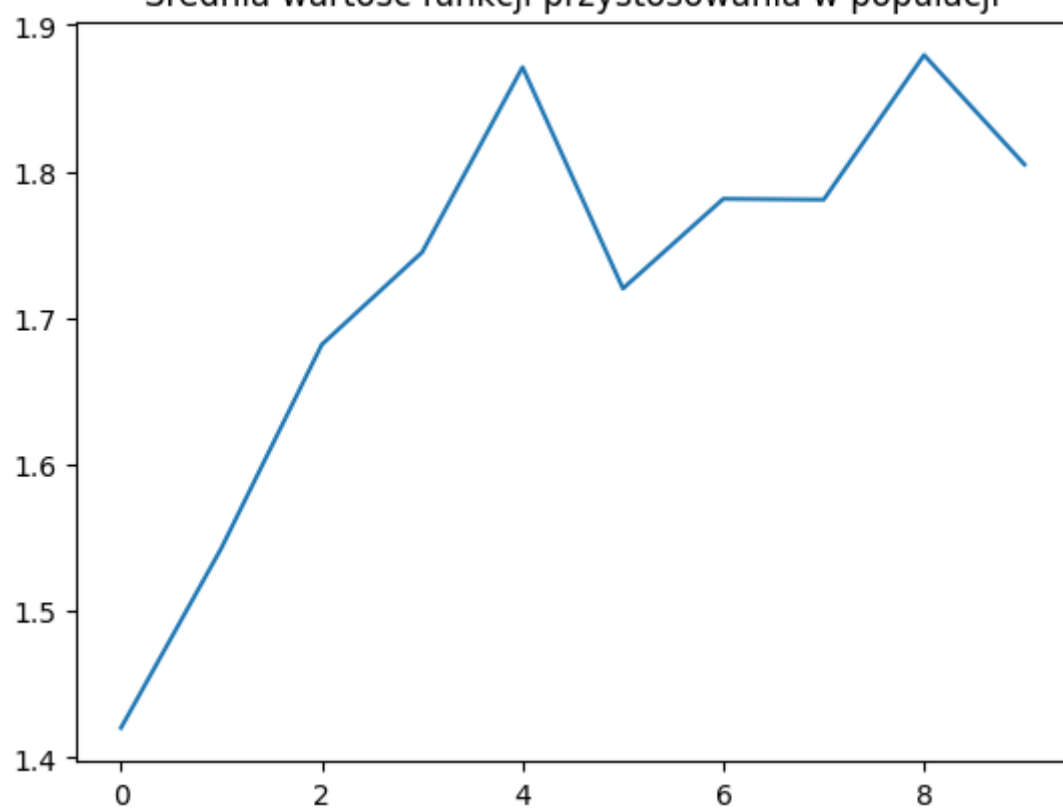


Przystosowanie najlepszego osobnika znajzonego podczas przebiegu algorytmu



6. Wyniki graficzne dla prawdopodobieństwa mutacji: 0,5 kodowanie:
graya:

Średnia wartość funkcji przystosowania w populacji





7. Wyniki graficzne dla prawdopodobieństwa mutacji: 1 kodowanie: graya:

Wnioski: Mutacje w algorytmach genetycznych mogą pomóc w osiągnięciu lepszego wyniku, zauważam również podobieństwo do algorytmów

herudystycznych takich jak algorytm symulowanego wyżarzania, w tamtym jednak wypadku, przyjmujemy gorsze rozwiązanie z pewnym prawdopodobieństwem, co z kolei może doprowadzić do lepszego wyniku. W mutacji przyjmujemy pewną losowość, która może doprowadzić do wyniku, niemożliwego do osiągnięcia przy replikacji bez zmian. Korzyści z mutacji wynikają bezpośrednio z tabeli.

Wyniki zadania 2:

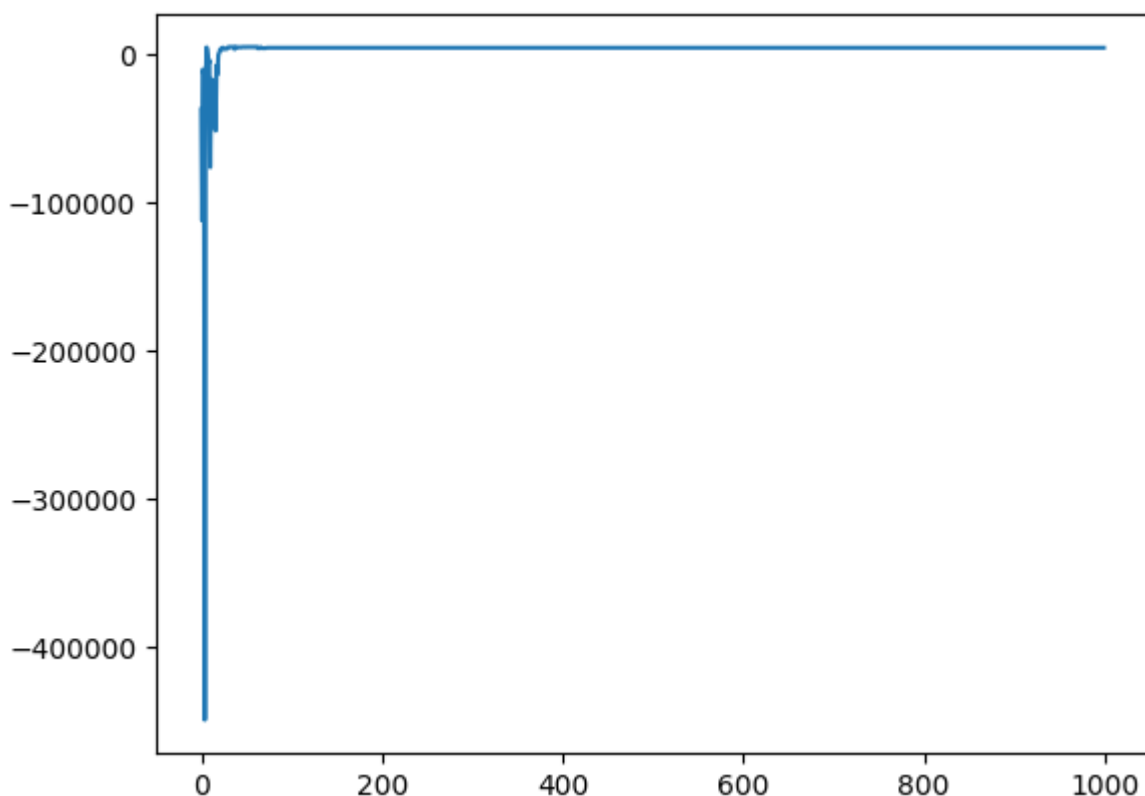
Jako chromosom użyłem wektora złożonego z 4 wartości typu int, w metodzie crossover każdą z wartości zamieniam na liczbę binarną, następnie łączę liczbę binarną, przy mutacji podobne działanie. Metoda doboru rodziców: koło ruletki

Wyniki dla wielkości populacji 100, licznie iteracji 1000:

Mutation probability: 30%

Best solution: [30, 35, 208, 48]

Best value: 5048.0

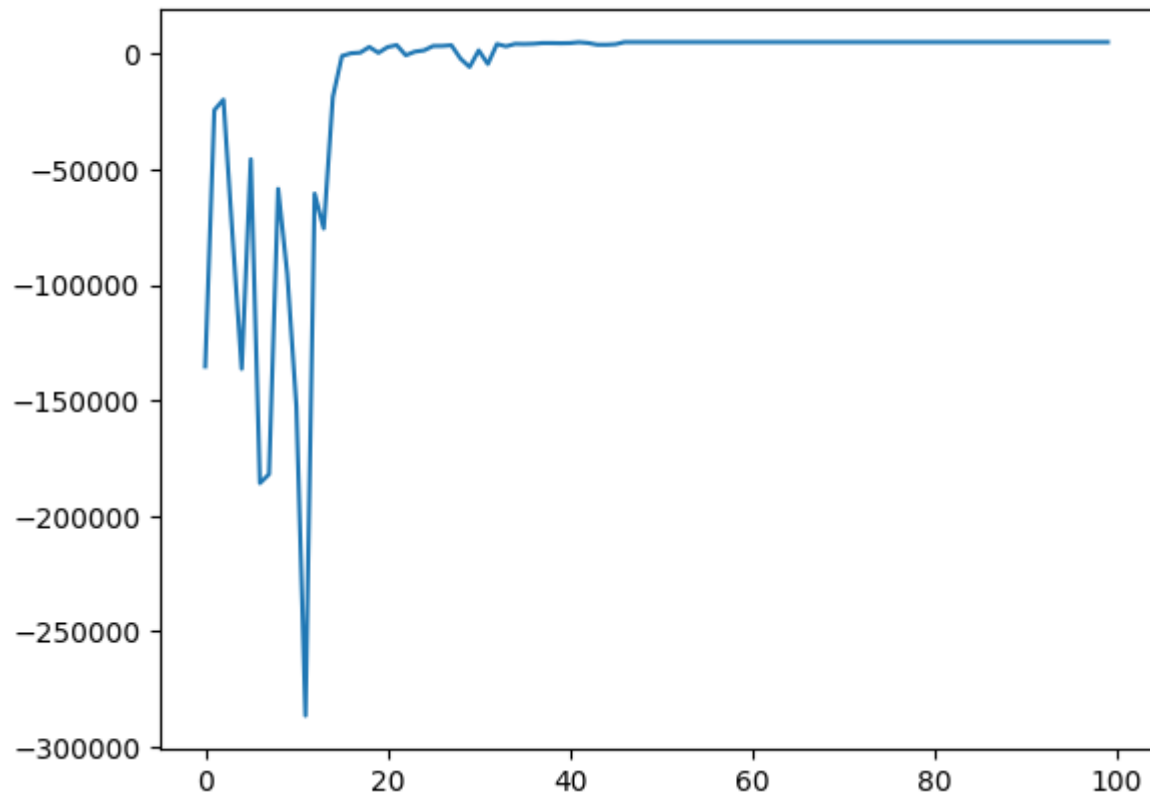


Mutation probability: 30%

Best solution: [20, 26, 223, 48]

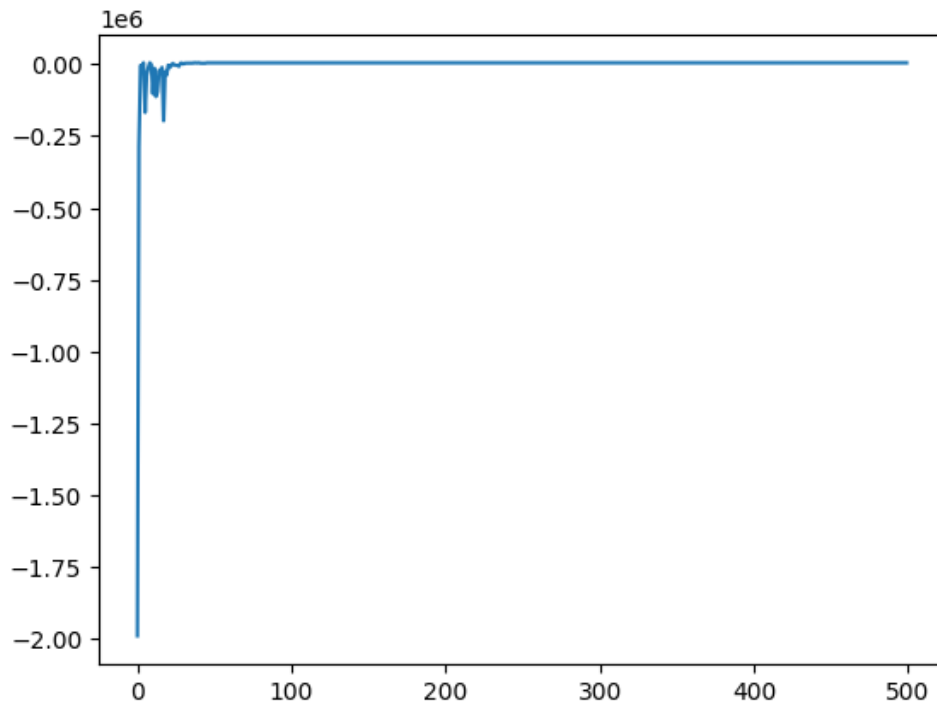
Best value: 5048.0

Iterations 100:



Best solution: [50, 35, 205, 49]

Best value: 5049.0



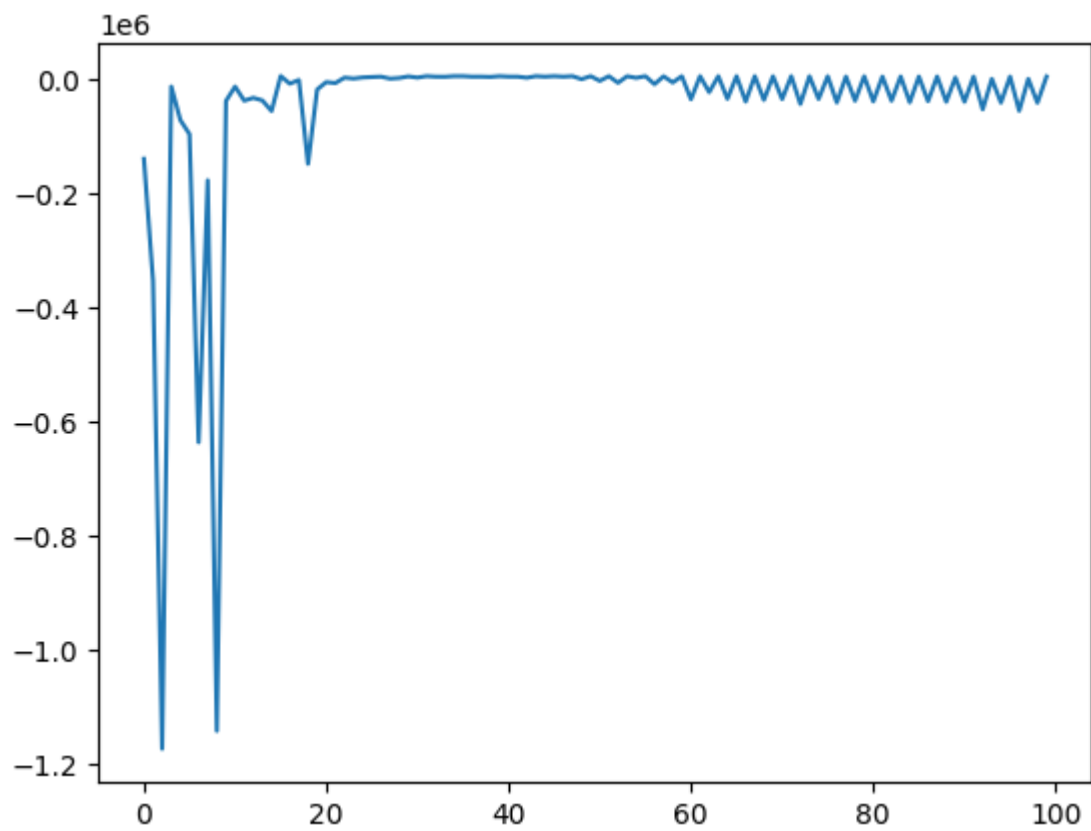
Wynik obliczony za pomocą wolfram alpha:

maximize	function	$5000 - 600 \times \frac{(x-20)^2 (y-35)^2}{z} - (x-50)^2 (w-48)^2 + w$
	domain	$x > 1$ and $x < 255$ and $y > 1$ and $y < 255$ and $z > 1$ and $z < 255$ and $w > 1$ and $w < 255$
Global maxima		
(no global maxima found)		
Local maxima		
$\max \left\{ 5000 - \frac{600((x-20)^2 (y-35)^2)}{z} - (x-50)^2 (w-48)^2 + w \mid \right.$ $\left. \{x > 1, x < 255, y > 1, y < 255, z > 1, z < 255, w > 1, w < 255\} \right\} \approx 5048.$ <p>at $(w, x, y, z) \approx (48.0006, 20., 36.9152, 7.87654)$</p>		
$\max \left\{ 5000 - \frac{600((x-20)^2 (y-35)^2)}{z} - (x-50)^2 (w-48)^2 + w \mid \right.$ $\left. \{x > 1, x < 255, y > 1, y < 255, z > 1, z < 255, w > 1, w < 255\} \right\} \approx 5048.$ <p>at $(w, x, y, z) \approx (48.0006, 20., 143.806, 128.354)$</p>		
$\max \left\{ 5000 - \frac{600((x-20)^2 (y-35)^2)}{z} - (x-50)^2 (w-48)^2 + w \mid \right.$ $\left. \{x > 1, x < 255, y > 1, y < 255, z > 1, z < 255, w > 1, w < 255\} \right\} \approx 5048.$ <p>at $(w, x, y, z) \approx (48.0006, 20., 128.512, 131.731)$</p>		
$\max \left\{ 5000 - \frac{600((x-20)^2 (y-35)^2)}{z} - (x-50)^2 (w-48)^2 + w \mid \right.$ $\left. \{x > 1, x < 255, y > 1, y < 255, z > 1, z < 255, w > 1, w < 255\} \right\} \approx 5255.$ <p>at $(w, x, y, z) \approx (255, 50., 35., 180.337)$</p>		

Ten sam algorytm przy kodowaniu Graya:

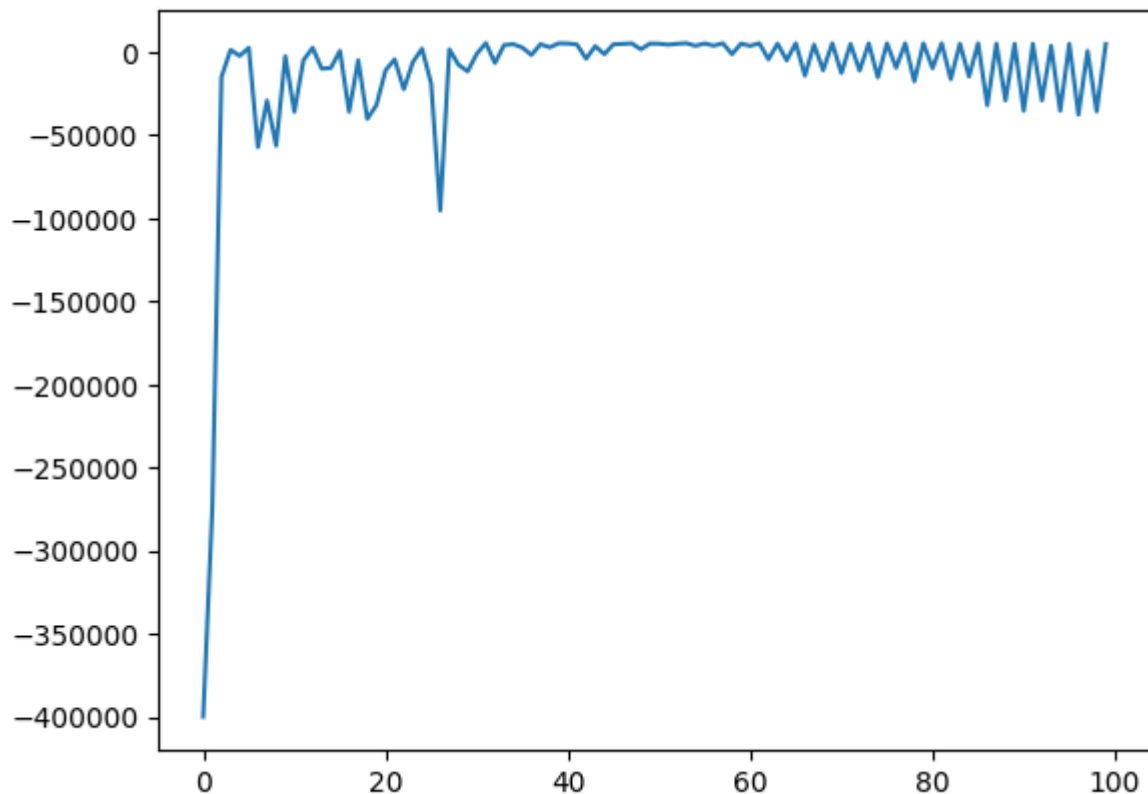
Best solution: [50, 35, 203, 244]

Best value: 5244.0



Best solution: [50, 35, 150, 148]

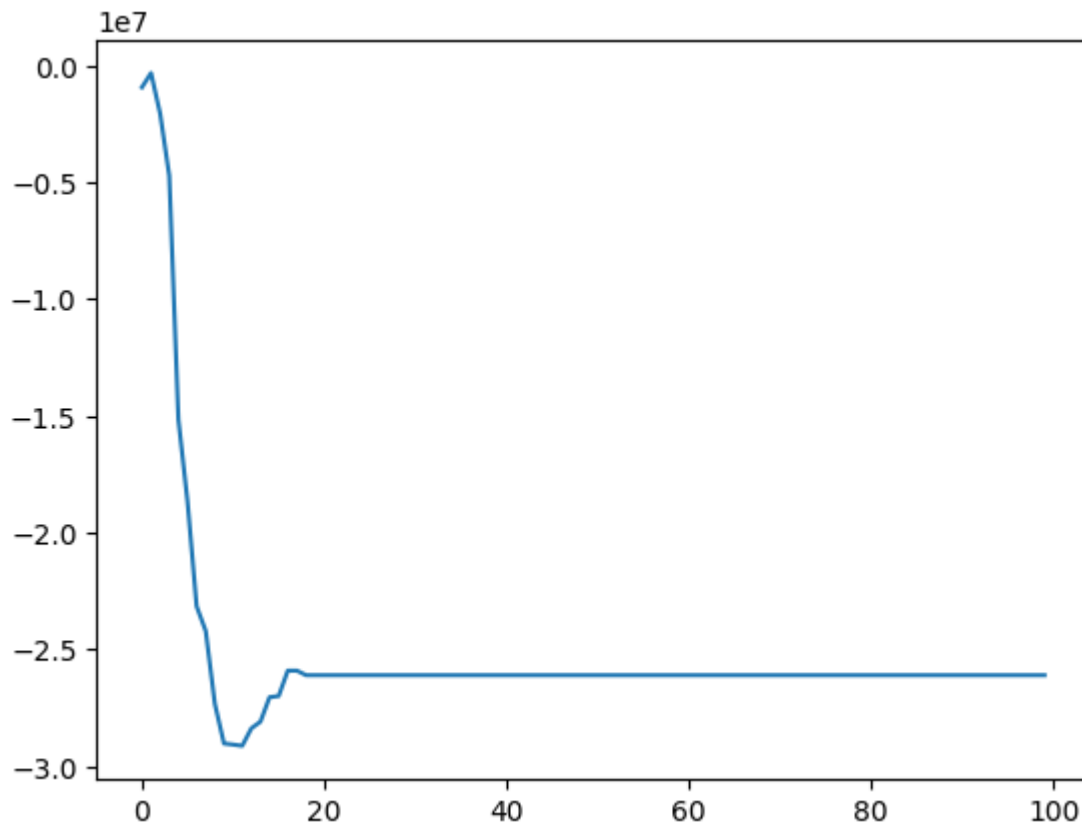
Best value: 5148.0



Wnioski: Przy doborze algorytmu genetycznego, najważniejsze znaczenie ma sposób doboru rodziców i sam sposób tworzenia potomków. Przed zmianą na kodowanie binarne lub kodowanie gray'a tworzyłem podomków przez wyliczenie średniej arytmetycznej z poszczególnych wartości rodziców, przynosiło to fatalne wyniki:

Best solution: [18, 182, 168, 53]

Best value: -329247.0



Kodowanie kodem Graya przynosi lepsze wyniki udało się osiągnąć bardzo dobry wynik funkcji maximum.

Link do repozytorium z kodem:

https://github.com/KamilPyla/MIO_2023/tree/master/lab_07