

POLITECHNIKA WROCŁAWSKA

LABORATORIUM

INTELIGENCJA OBliczeniowa i JEJ ZASTOSOWANIA

Algorytmy ewolucyjne i hybrydowe

Authors:

Rafał PIENIĄŻEK
Jakub POMYKAŁA

Supervisor:

prof. dr inż. Olgierd UNOLD

8 maja 2018

Spis treści

1 Wstęp	4
2 Zastosowany algorytm optymalizacji	4
3 Funkcja Shuberta	5
3.1 Wzór analityczny	5
3.2 Wykres w ustalonym przedziale zmiennych	5
3.3 Optymalizacja poszukiwania ekstremum globalnego	6
3.3.1 Modyfikacja parametru elitarności	6
3.3.2 Modyfikacja parametru mutacji	8
3.3.3 Modyfikacja parametru krzyżowania	11
3.3.4 Modyfikacja parametru liczby iteracji	14
3.3.5 Modyfikacja parametru rozmiaru populacji	17
4 Funkcja Bohachevsky'ego	20
4.1 Wzór analityczny	20
4.2 Wykres funkcji	20
4.3 Optymalizacja	21
4.3.1 Modyfikacja parametru elitarności	21
4.3.2 Modyfikacja parametru mutacji	24
4.3.3 Modyfikacja parametru krzyżowania	27
4.3.4 Modyfikacja parametru liczby iteracji	30
4.3.5 Modyfikacja parametru rozmiaru populacji	33
5 Funkcja Branina	36
5.1 Wzór analityczny	36
5.2 Wykres w ustalonym przedziale zmiennych	36
5.3 Optymalizacja	37
5.3.1 Modyfikacja parametru elitarności	37
5.3.2 Modyfikacja parametru mutacji	40
5.3.3 Modyfikacja parametru krzyżowania	43
5.3.4 Modyfikacja parametru liczby iteracji	46
5.3.5 Modyfikacja parametru rozmiaru populacji	49
6 Wnioski	51

Spis rysunków

1	Wzór analityczny funkcji Schuberta	5
2	Minimum globalne dla funkcji Schuberta	5
3	Wykres funkcji Schuberta	5
4	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0	6
5	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.25	6
6	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.5	7
7	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.75	7
8	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e1	8
9	Test optymalizacji GA Schubert p50 i100 c0.8 m0 e0.05	8
10	Test optymalizacji GA Schubert p50 i100 c0.8 m0.25 e0.05	9
11	Test optymalizacji GA Schubert p50 i100 c0.8 m0.5 e0.05	9
12	Test optymalizacji GA Schubert p50 i100 c0.8 m0.75 e0.05	10
13	Test optymalizacji GA Schubert p50 i100 c0.8 m1 e0.05	10
14	Test optymalizacji GA Schubert p50 i100 c0 m0.1 e0.05	11
15	Test optymalizacji GA Schubert p50 i100 c0.25 m0.1 e0.05	11
16	Test optymalizacji GA Schubert p50 i100 c0.5 m0.1 e0.05	12
17	Test optymalizacji GA Schubert p50 i100 c0.75 m0.1 e0.05	12
18	Test optymalizacji GA Schubert p50 i100 c1 m0.1 e0.05	13
19	Test optymalizacji GA Schubert p50 i50 c0.8 m0.1 e0.05	14
20	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.05	14
21	Test optymalizacji GA Schubert p50 i150 c0.8 m0.1 e0.05	15
22	Test optymalizacji GA Schubert p50 i200 c0.8 m0.1 e0.05	15
23	Test optymalizacji GA Schubert p50 i250 c0.8 m0.1 e0.05	16
24	Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.05	17
25	Test optymalizacji GA Schubert p100 i100 c0.8 m0.1 e0.05	17
26	Test optymalizacji GA Schubert p150 i100 c0.8 m0.1 e0.05	18
27	Test optymalizacji GA Schubert p200 i100 c0.8 m0.1 e0.05	18
28	Test optymalizacji GA Schubert p250 i100 c0.8 m0.1 e0.05	19
29	Wzór analityczny funkcji Bochachevsky'ego	20
30	Minimum globalne funkcji Bochachevsky'ego	20
31	Wzór analityczny funkcji Bochachevsky'ego	20
32	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0	21
33	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.25	21
34	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.5	22
35	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.75	22
36	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e1	23
37	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0 e0.05	24
38	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.25 e0.05	24
39	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.5 e0.05	25
40	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.75 e0.05	25
41	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m1 e0.05	26
42	Test optymalizacji GA Bohachevsky1 p50 i100 c0 m0.1 e0.05	27
43	Test optymalizacji GA Bohachevsky1 p50 i100 c0.25 m0.1 e0.05	27
44	Test optymalizacji GA Bohachevsky1 p50 i100 c0.5 m0.1 e0.05	28
45	Test optymalizacji GA Bohachevsky1 p50 i100 c0.75 m0.1 e0.05	28
46	Test optymalizacji GA Bohachevsky1 p50 i100 c1 m0.1 e0.05	29
47	Test optymalizacji GA Bohachevsky1 p50 i50 c0.8 m0.1 e0.05	30
48	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.05	30
49	Test optymalizacji GA Bohachevsky1 p50 i150 c0.8 m0.1 e0.05	31

50	Test optymalizacji GA Bohachevsky1 p50 i200 c0.8 m0.1 e0.05	31
51	Test optymalizacji GA Bohachevsky1 p50 i250 c0.8 m0.1 e0.05	32
52	Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.05	33
53	Test optymalizacji GA Bohachevsky1 p100 i100 c0.8 m0.1 e0.05	33
54	Test optymalizacji GA Bohachevsky1 p150 i100 c0.8 m0.1 e0.05	34
55	Test optymalizacji GA Bohachevsky1 p200 i100 c0.8 m0.1 e0.05	34
56	Test optymalizacji GA Bohachevsky1 p250 i100 c0.8 m0.1 e0.05	35
57	Wzór analityczny funkcji Branina	36
58	Minimum globalne funkcji Branina	36
59	Wzór analityczny funkcji Branina	36
60	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0	37
61	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.25	37
62	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.5	38
63	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.75	38
64	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e1	39
65	Test optymalizacji GA Branin p50 i100 c0.8 m0 e0.05	40
66	Test optymalizacji GA Branin p50 i100 c0.8 m0.25 e0.05	40
67	Test optymalizacji GA Branin p50 i100 c0.8 m0.5 e0.05	41
68	Test optymalizacji GA Branin p50 i100 c0.8 m0.75 e0.05	41
69	Test optymalizacji GA Branin p50 i100 c0.8 m1 e0.05	42
70	Test optymalizacji GA Branin p50 i100 c0 m0.1 e0.05	43
71	Test optymalizacji GA Branin p50 i100 c0.25 m0.1 e0.05	43
72	Test optymalizacji GA Branin p50 i100 c0.5 m0.1 e0.05	44
73	Test optymalizacji GA Branin p50 i100 c0.75 m0.1 e0.05	44
74	Test optymalizacji GA Branin p50 i100 c1 m0.1 e0.05	45
75	Test optymalizacji GA Branin p50 i50 c0.8 m0.1 e0.05	46
76	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.05	46
77	Test optymalizacji GA Branin p50 i150 c0.8 m0.1 e0.05	47
78	Test optymalizacji GA Branin p50 i200 c0.8 m0.1 e0.05	47
79	Test optymalizacji GA Branin p50 i250 c0.8 m0.1 e0.05	48
80	Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.05	49
81	Test optymalizacji GA Branin p100 i100 c0.8 m0.1 e0.05	49
82	Test optymalizacji GA Branin p150 i100 c0.8 m0.1 e0.05	50
83	Test optymalizacji GA Branin p200 i100 c0.8 m0.1 e0.05	50
84	Test optymalizacji GA Branin p250 i100 c0.8 m0.1 e0.05	51

1 Wstęp

Celem laboratorium było przeprowadzenie optymalizacji globalnej dla wybranych funkcji z pakietu globalOptTests.

2 Zastosowany algorytm optymalizacji

W laboratorium zastosowano algorytmy genetyczne będące klasą algorytmów ewolucyjnych. Algorytmy ewolucyjne stanowią kierunek sztucznej inteligencji, która wykorzystuje i symuluje ewolucję biologiczną. Wszystkie algorytmy tej klasy symulują podstawowe zachowania w teorii ewolucji biologicznej - procesy selekcji, mutacji i reprodukcji. Zachowanie jednostek zależy od środowiska. Zbiór jednostek nazywa się populacją. Taka populacja ewoluje zgodnie z regułami selekcji zgodnie z funkcją celu przypisaną do środowiska. Propagowane do kolejnych pokoleń są tylko najbardziej dopasowane osobniki.

3 Funkcja Schuberta

3.1 Wzór analityczny

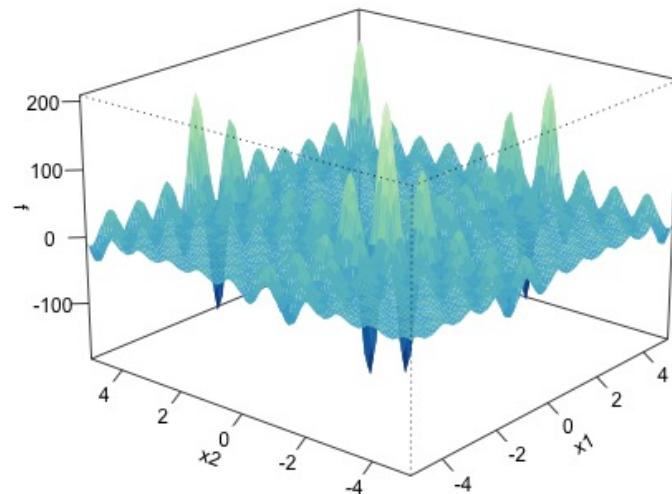
$$f(\mathbf{x}) = \left(\sum_{i=1}^5 i \cos((i+1)x_1 + i) \right) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right)$$

Rysunek 1: Wzór analityczny funkcji Schuberta

$$f(\mathbf{x}^*) = -186.7309$$

Rysunek 2: Minimum globalne dla funkcji Schuberta

3.2 Wykres w ustalonym przedziale zmiennych

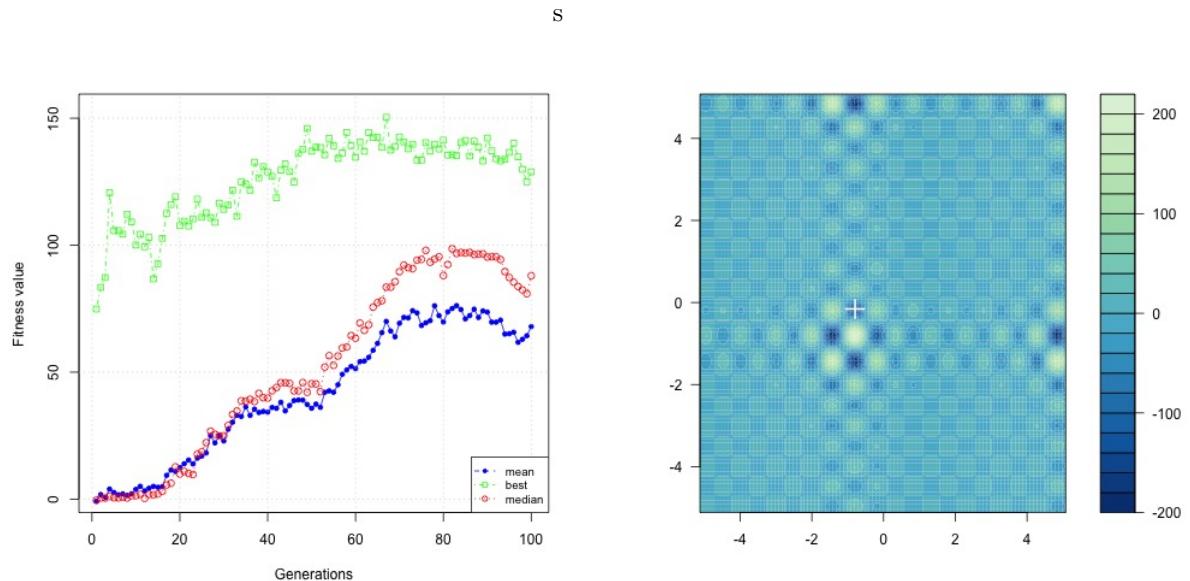


Rysunek 3: Wykres funkcji Schuberta

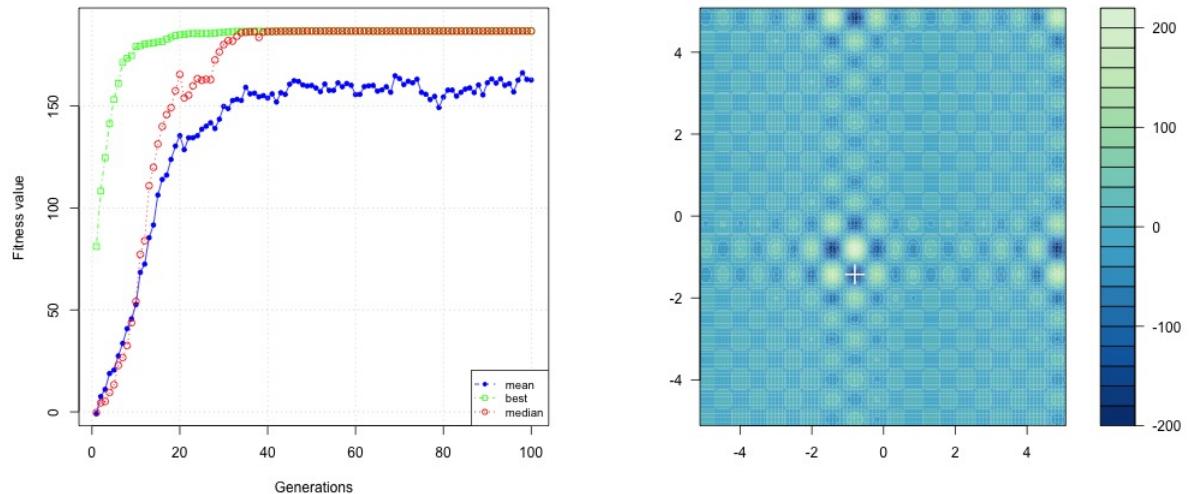
3.3 Optymalizacja poszukiwania ekstremum globalnego

3.3.1 Modyfikacja parametru elitarności

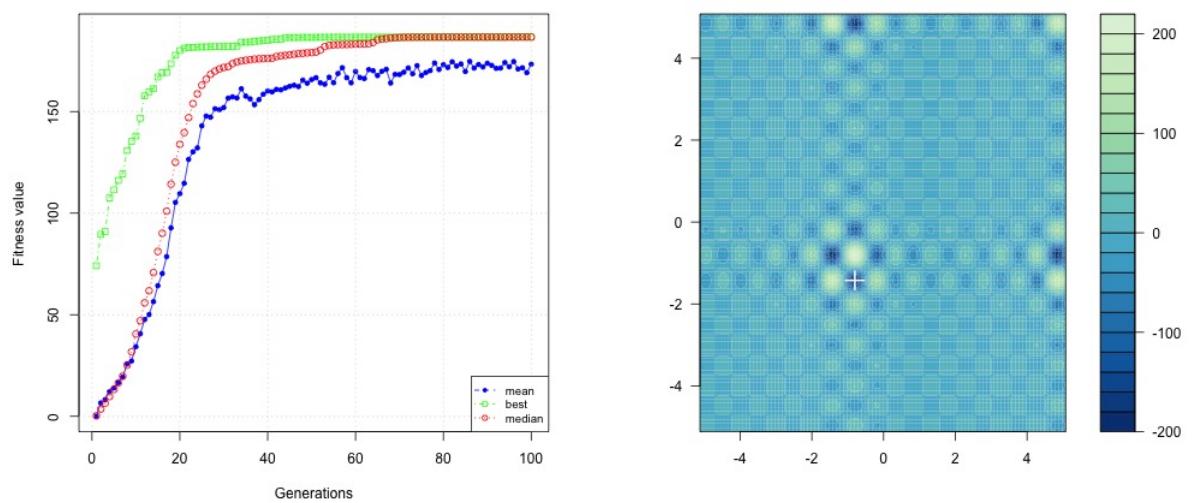
W przypadku zwiększenia wartości elitarności populacji średnie wartości odchyleń zmniejszają się. W przypadku większej elitarności algorytm szybciej znajduje optymalne rozwiązanie. Jeżeli elitarność populacji jest równa 1, to algorytm stochastycznie wylosował jeden zestaw rozwiązań, który z powodu braku ewolucji nie zbliżył się do rozwiązania optymalnego.



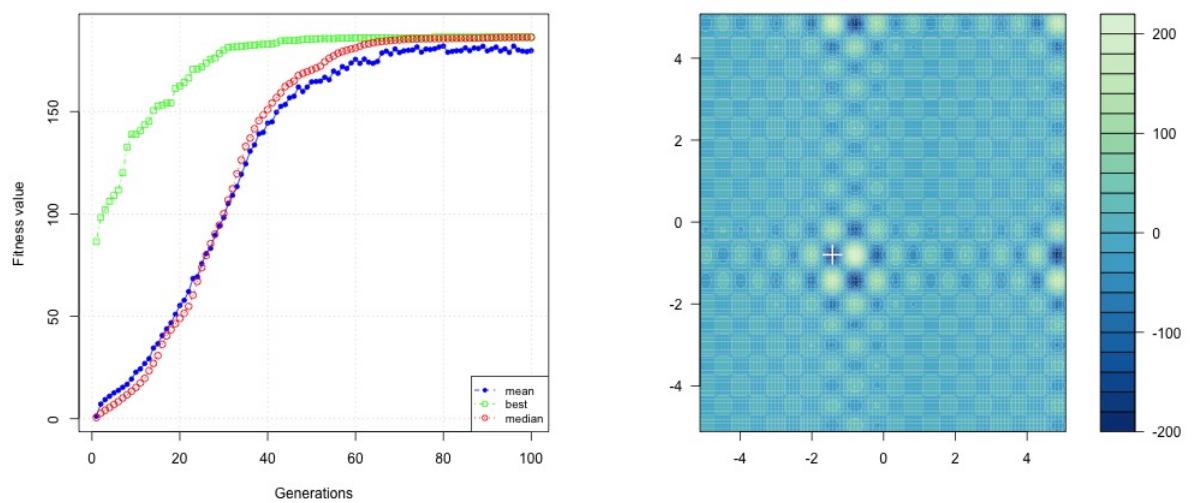
Rysunek 4: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0



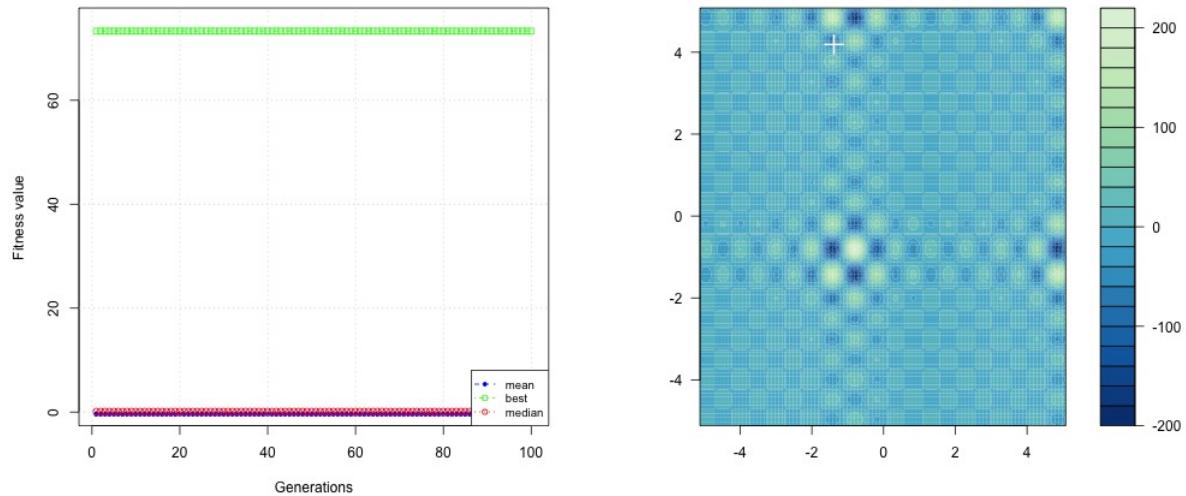
Rysunek 5: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.25



Rysunek 6: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.5



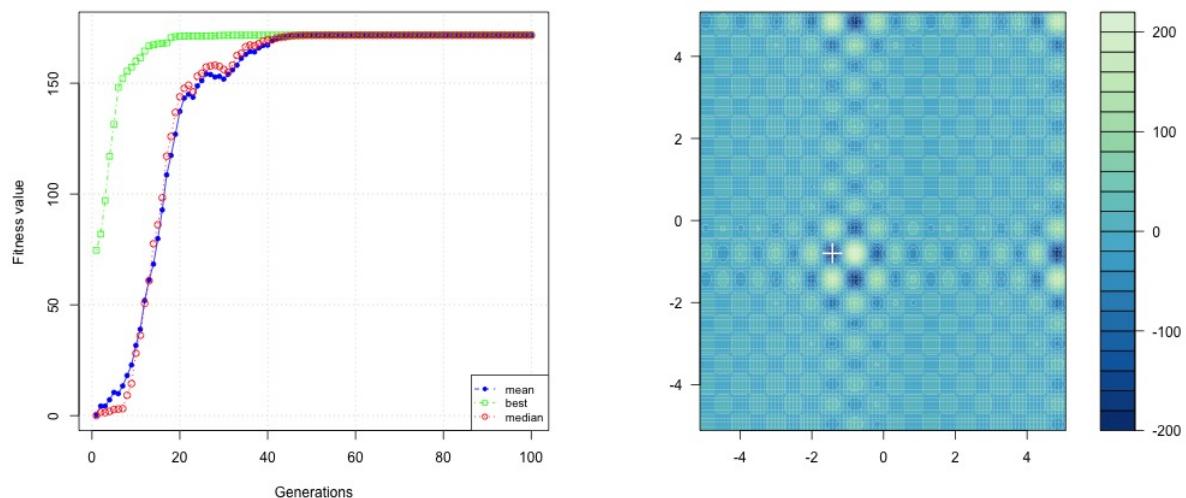
Rysunek 7: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.75



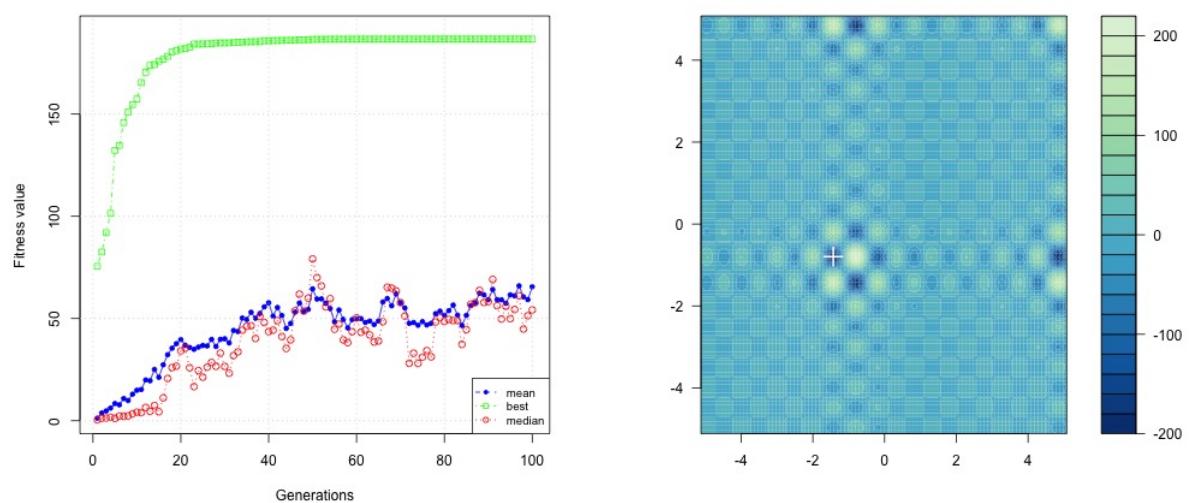
Rysunek 8: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e1

3.3.2 Modyfikacja parametru mutacji

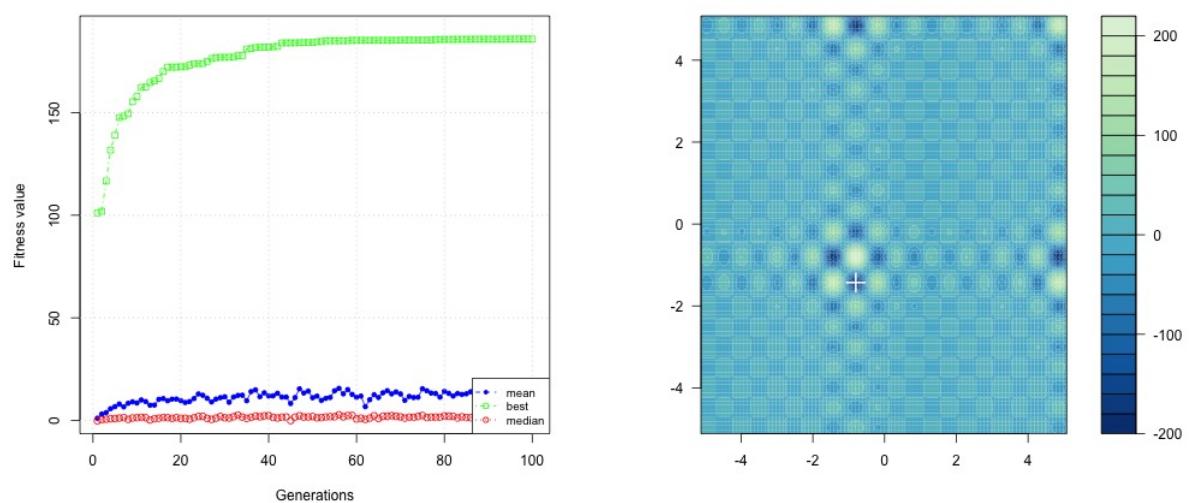
W przypadku zwiększenia wartości mutacji zarówno średnia jak i mediana zmniejszają się się. Algorytm zawsze znajduje po podobnej liczbie populacji rozwiązanie optymalne.



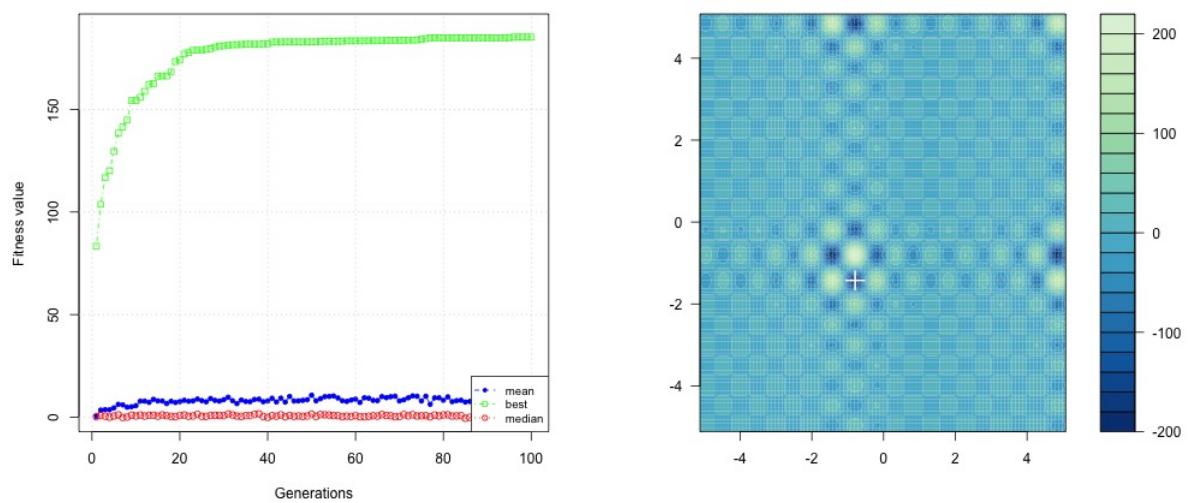
Rysunek 9: Test optymalizacji GA Schubert p50 i100 c0.8 m0.05



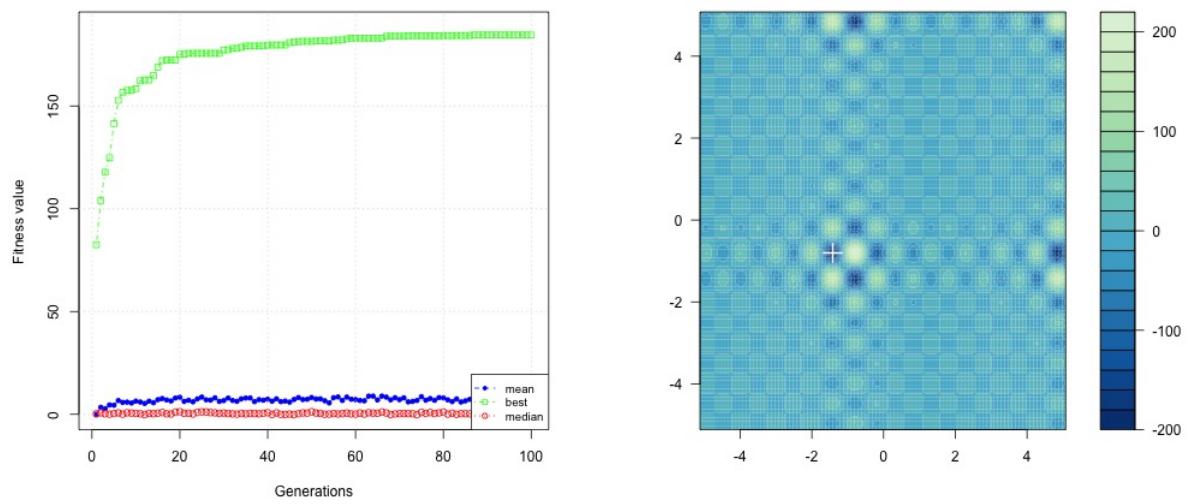
Rysunek 10: Test optymalizacji GA Schubert p50 i100 c0.8 m0.25 e0.05



Rysunek 11: Test optymalizacji GA Schubert p50 i100 c0.8 m0.5 e0.05



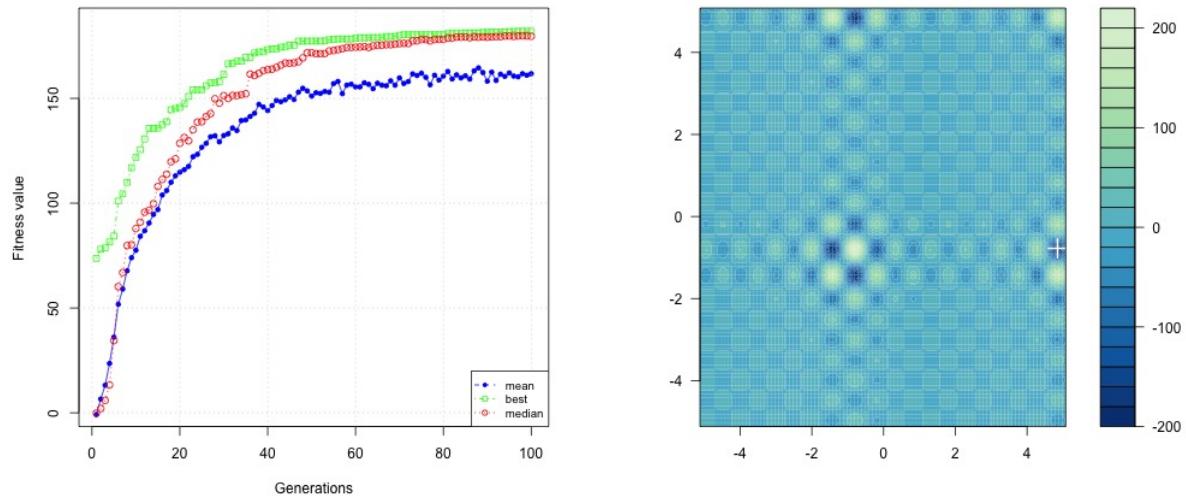
Rysunek 12: Test optymalizacji GA Schubert p50 i100 c0.8 m0.75 e0.05



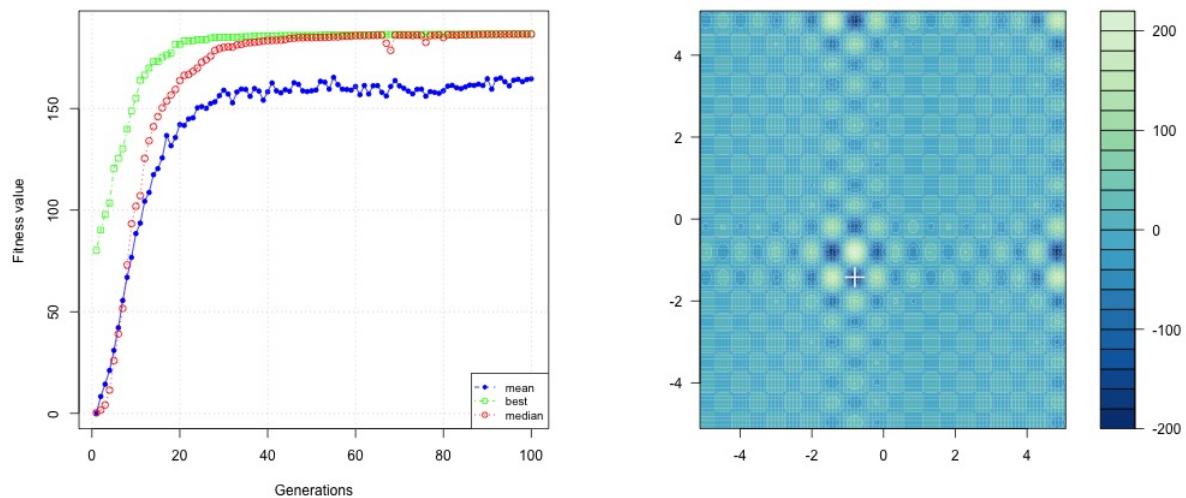
Rysunek 13: Test optymalizacji GA Schubert p50 i100 c0.8 m1 e0.05

3.3.3 Modyfikacja parametru krzyżowania

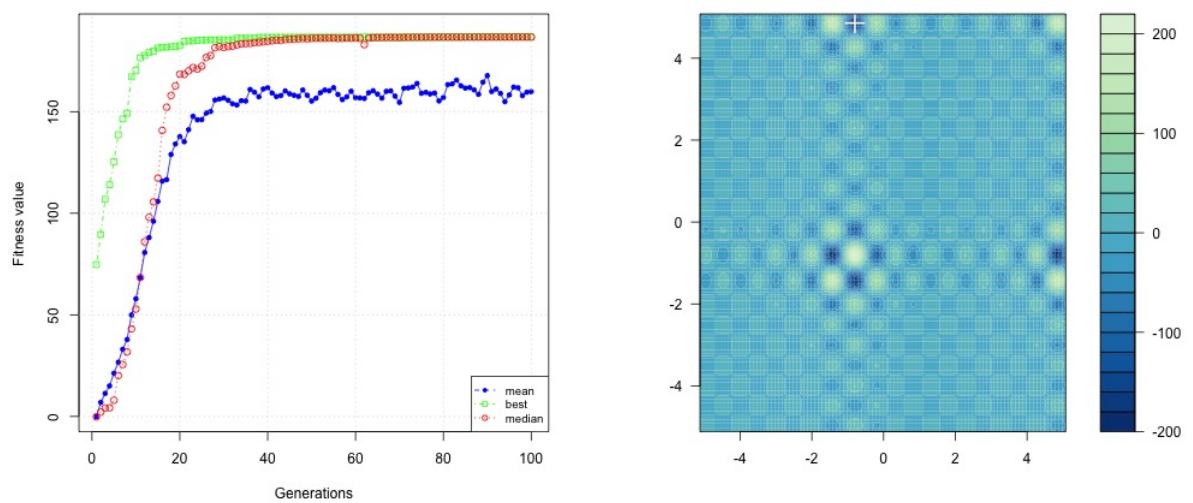
W przypadku zwiększenia parametru krzyżowania algorytm szybciej znajduje lokalne minimum, ale nie zawsze jest to ekstremum globalne. Wartości średniej i mediany pozostają na podobnym poziomie niezależnie od parametrów.



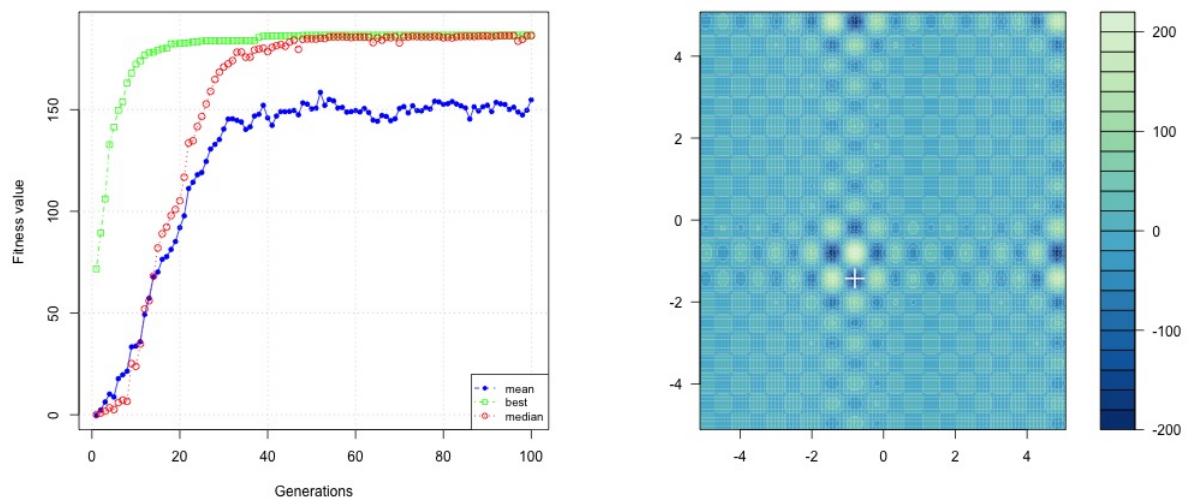
Rysunek 14: Test optymalizacji GA Schubert p50 i100 c0 m0.1 e0.05



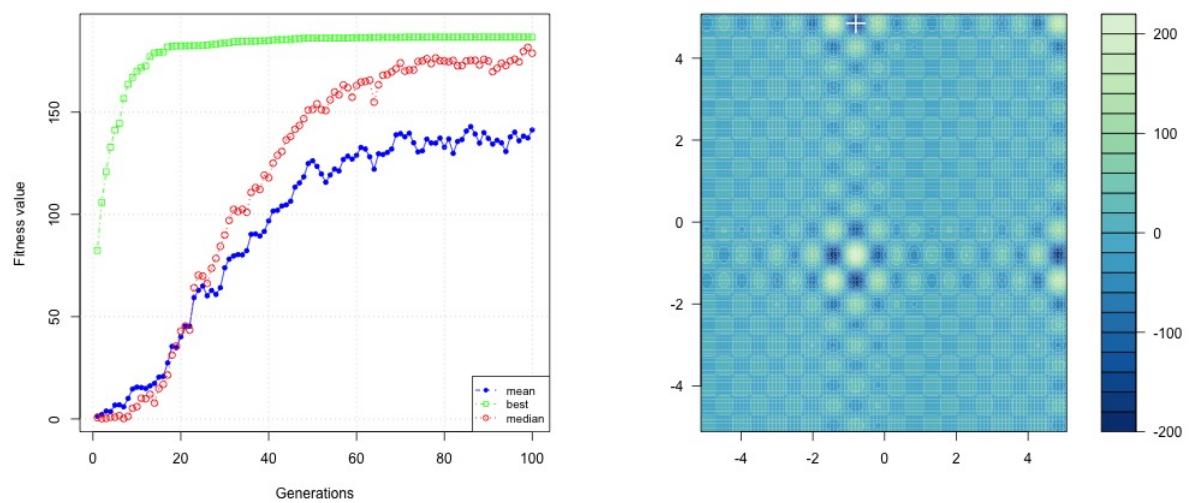
Rysunek 15: Test optymalizacji GA Schubert p50 i100 c0.25 m0.1 e0.05



Rysunek 16: Test optymalizacji GA Schubert p50 i100 c0.5 m0.1 e0.05



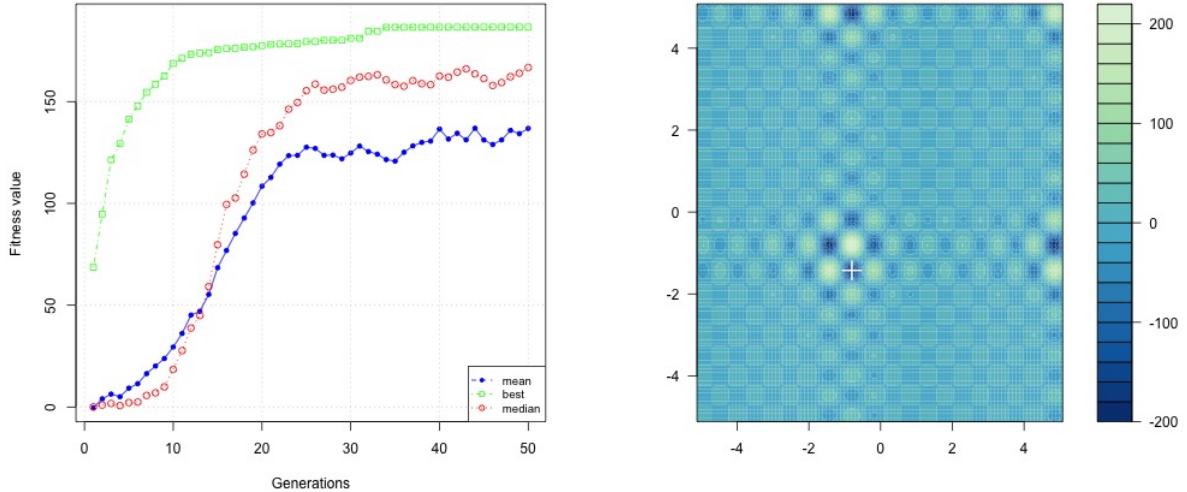
Rysunek 17: Test optymalizacji GA Schubert p50 i100 c0.75 m0.1 e0.05



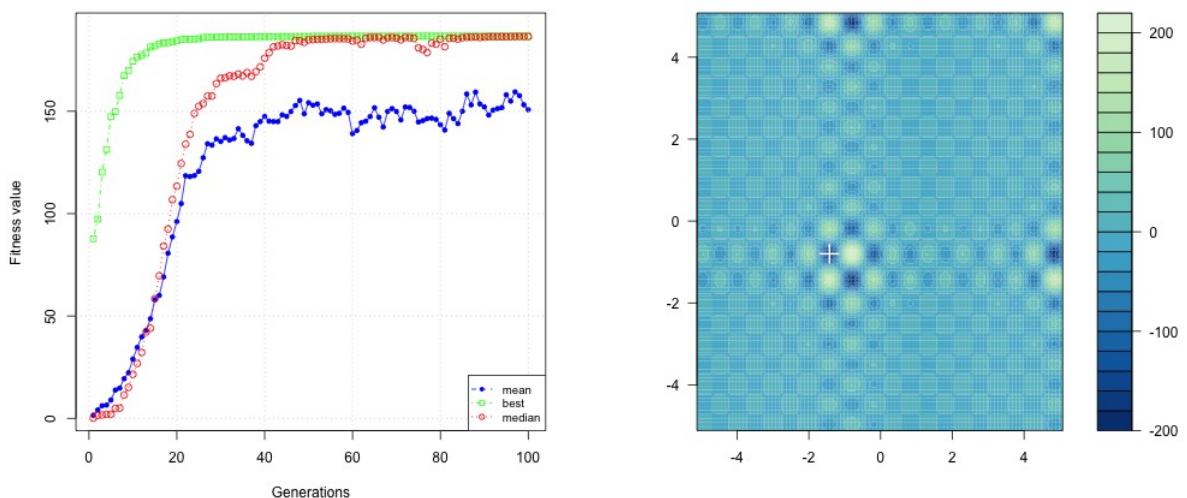
Rysunek 18: Test optymalizacji GA Schubert p50 i100 c1 m0.1 e0.05

3.3.4 Modyfikacja parametru liczby iteracji

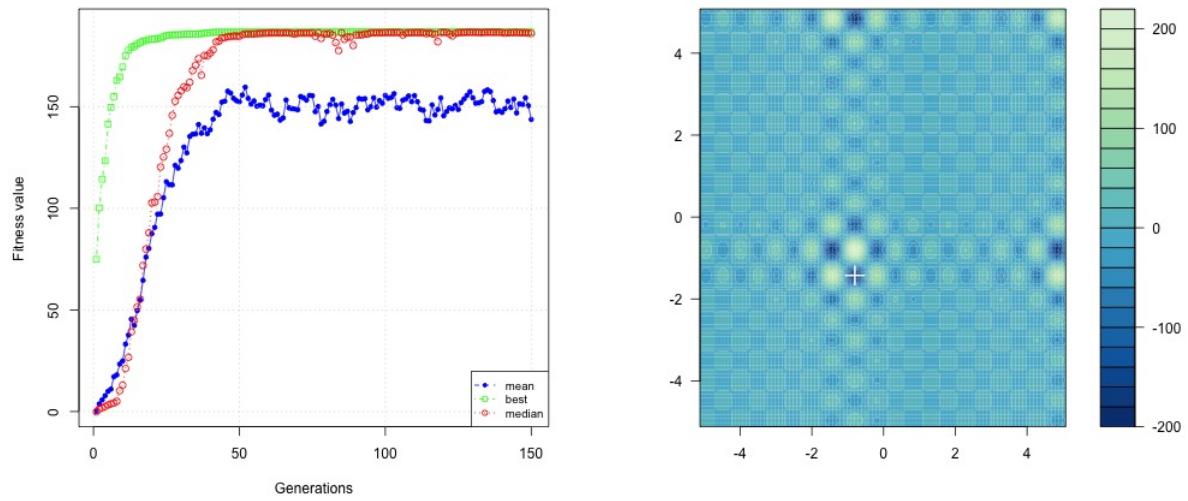
W przypadku zwiększenia ilości iteracji zarówno średnia jak i mediana pozostają na podobnym poziomie. Zmiana parametru liczby iteracji nie wpływa również znacząco na wyniki.



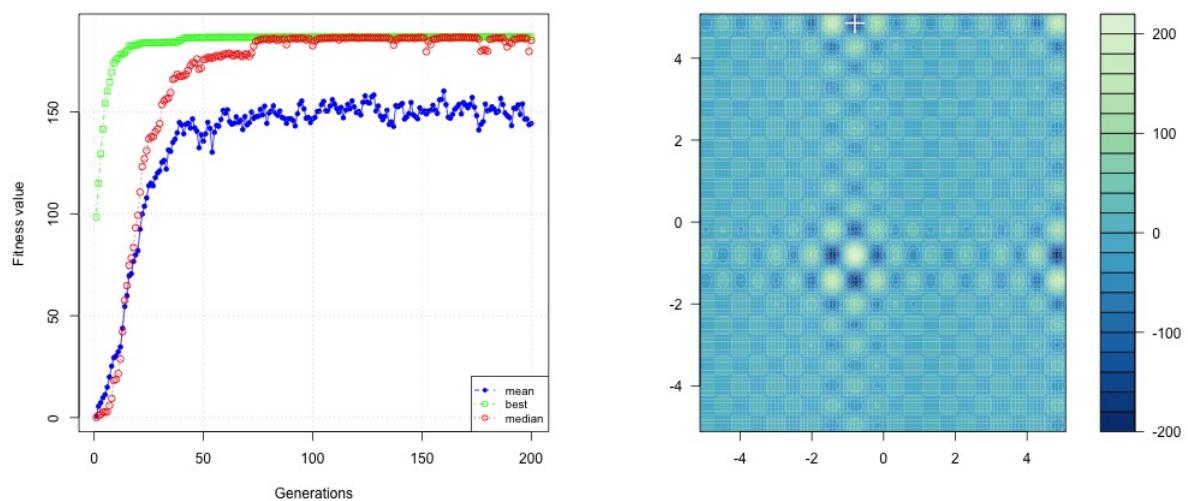
Rysunek 19: Test optymalizacji GA Schubert p50 i50 c0.8 m0.1 e0.05



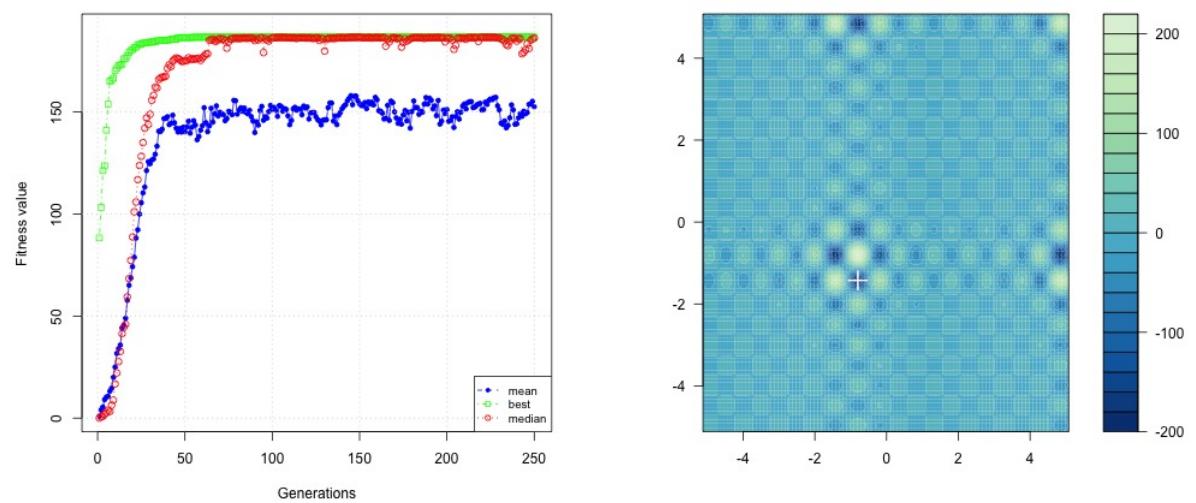
Rysunek 20: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.05



Rysunek 21: Test optymalizacji GA Schubert p50 i150 c0.8 m0.1 e0.05



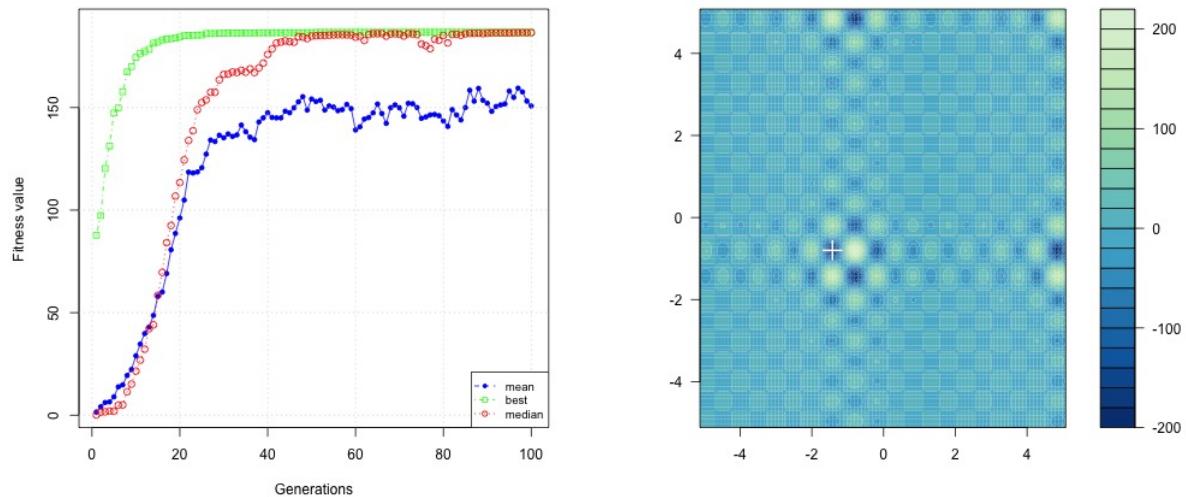
Rysunek 22: Test optymalizacji GA Schubert p50 i200 c0.8 m0.1 e0.05



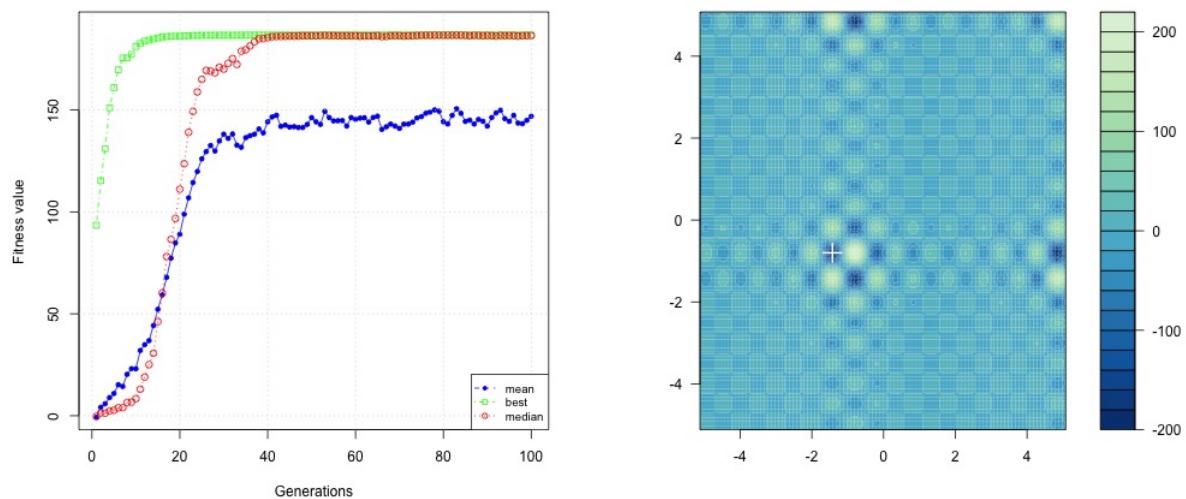
Rysunek 23: Test optymalizacji GA Schubert p50 i250 c0.8 m0.1 e0.05

3.3.5 Modyfikacja parametru rozmiaru populacji

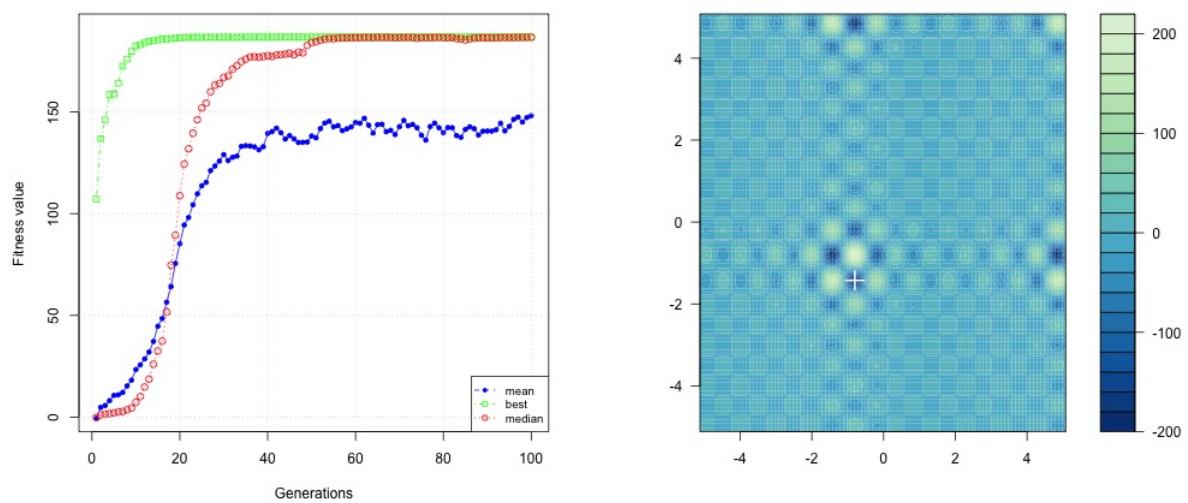
Gdy parametr rozmiaru populacji jest ustawiony na wartość populacji 250 można zaobserwować małą anomalię średniej i mediany w okolicach 50-60 pokolenia. Anomalie średniej i mediany są skorelowane ze sobą, jednak nie wpływają na znalezione rozwiązanie. Niezależnie od rozmiaru parametru algorytm znajduje minimum globalne przy podobnej ilości pokoleń.



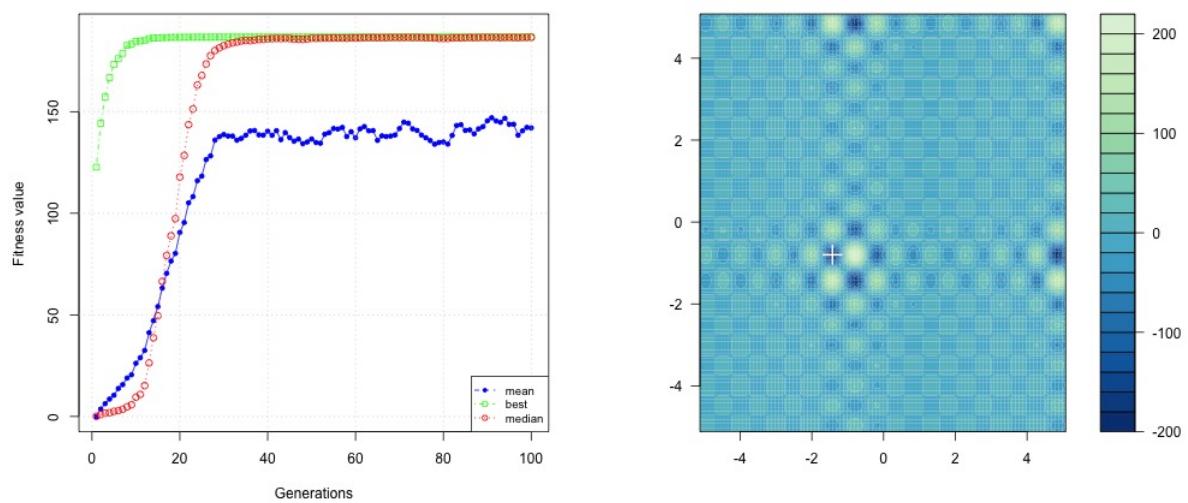
Rysunek 24: Test optymalizacji GA Schubert p50 i100 c0.8 m0.1 e0.05



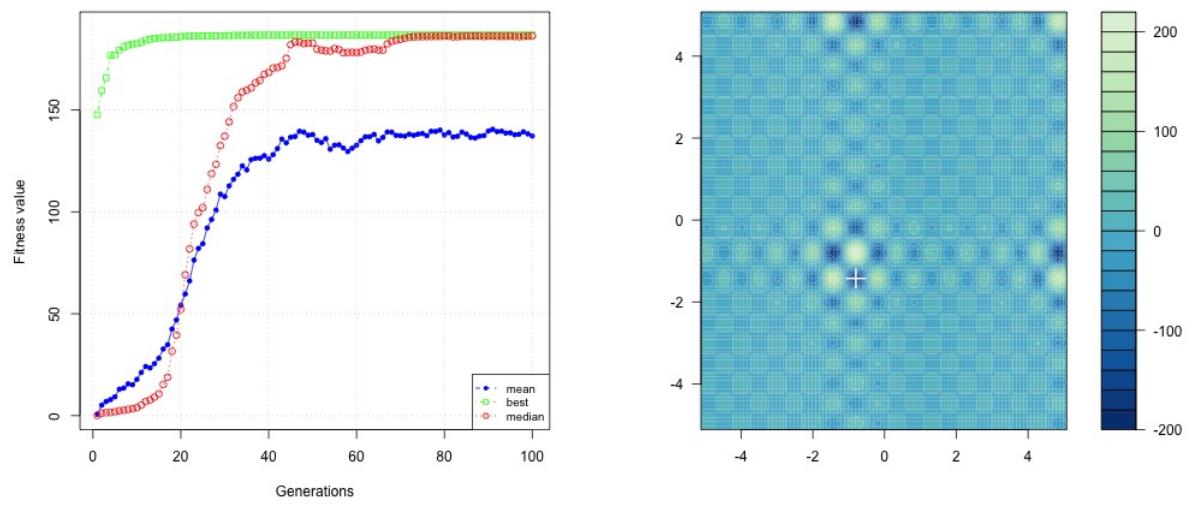
Rysunek 25: Test optymalizacji GA Schubert p100 i100 c0.8 m0.1 e0.05



Rysunek 26: Test optymalizacji GA Schubert p150 i100 c0.8 m0.1 e0.05



Rysunek 27: Test optymalizacji GA Schubert p200 i100 c0.8 m0.1 e0.05



Rysunek 28: Test optymalizacji GA Schubert p250 i100 c0.8 m0.1 e0.05

4 Funkcja Bohachevsky'ego

4.1 Wzór analityczny

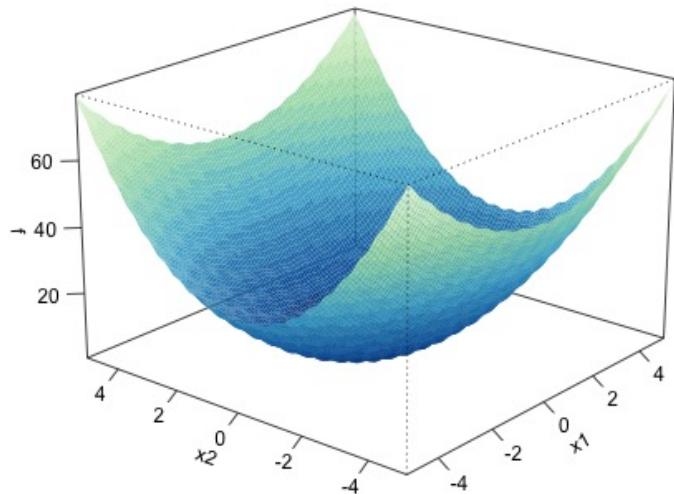
$$f(x, y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$$

Rysunek 29: Wzór analityczny funkcji Bochachevsky'ego

$$f_j(\mathbf{x}^*) = 0, \text{ at } \mathbf{x}^* = (0, 0), \text{ for all } j = 1, 2, 3$$

Rysunek 30: Minimum globalne funkcji Bochachevsky'ego

4.2 Wykres funkcji

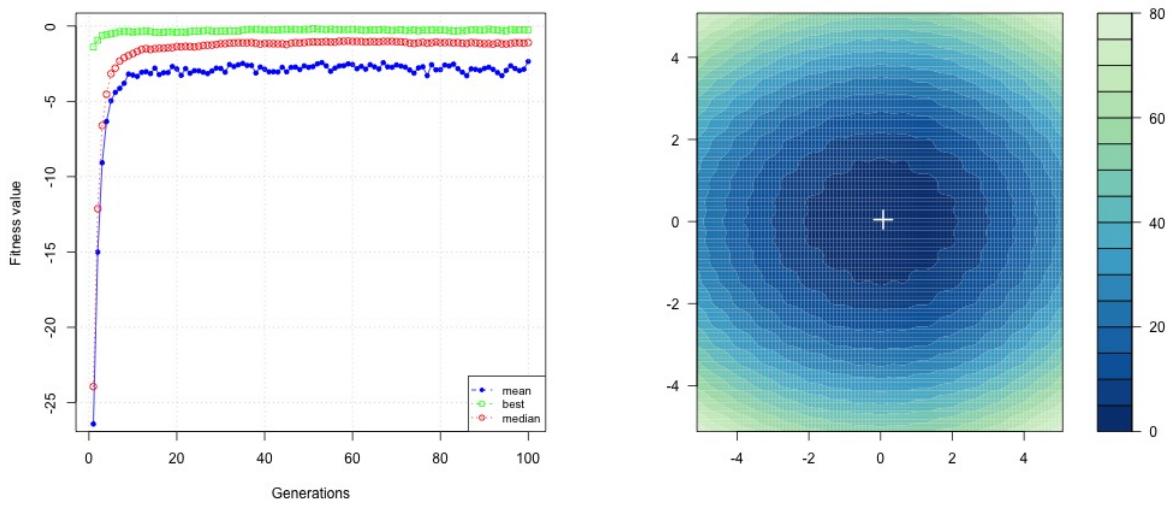


Rysunek 31: Wzór analityczny funkcji Bochachevsky'ego

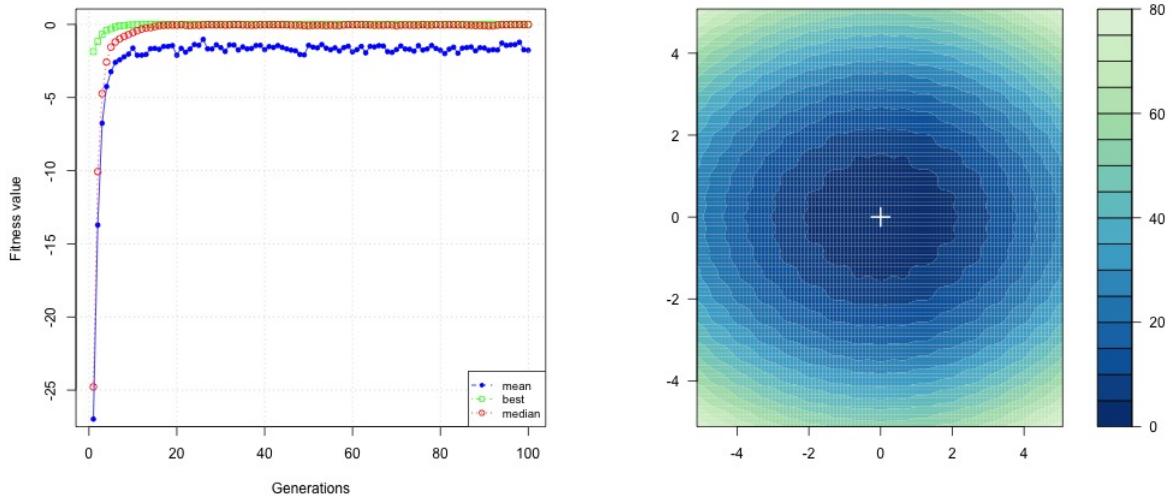
4.3 Optymalizacja

4.3.1 Modyfikacja parametru elitarności

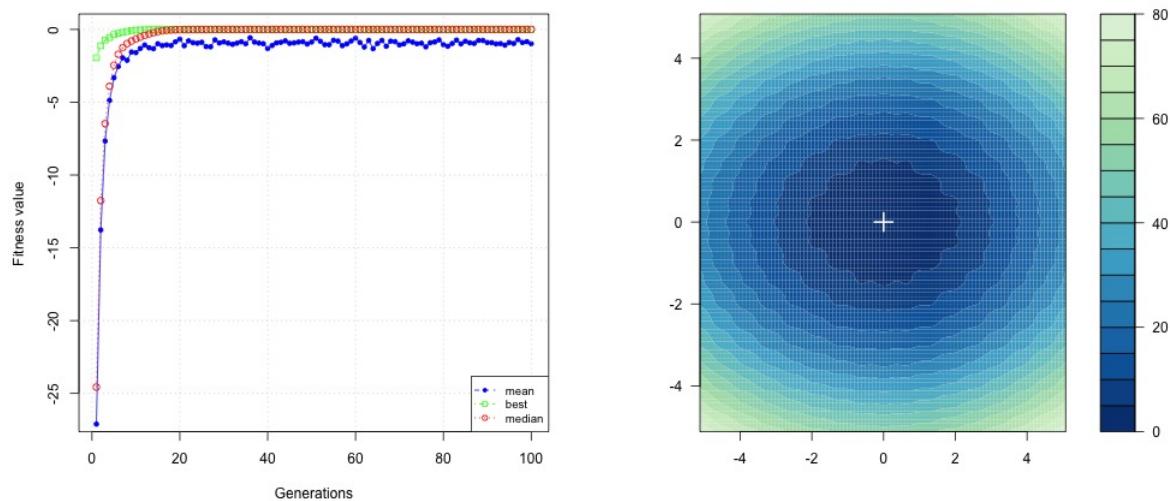
W przypadku zwiększenia wartości elitarności populacji średnia i mediana pozostają na podobnym poziomie. W przypadku większej elitarności algorytm szybciej znajduje optymalne rozwiązanie. Jeżeli elitarność populacji jest równa 1, to algorytm stochastycznie wlosował jeden zestaw rozwiązań, który z powodu braku ewolucji nie zbliżył się do rozwiązania optymalnego.



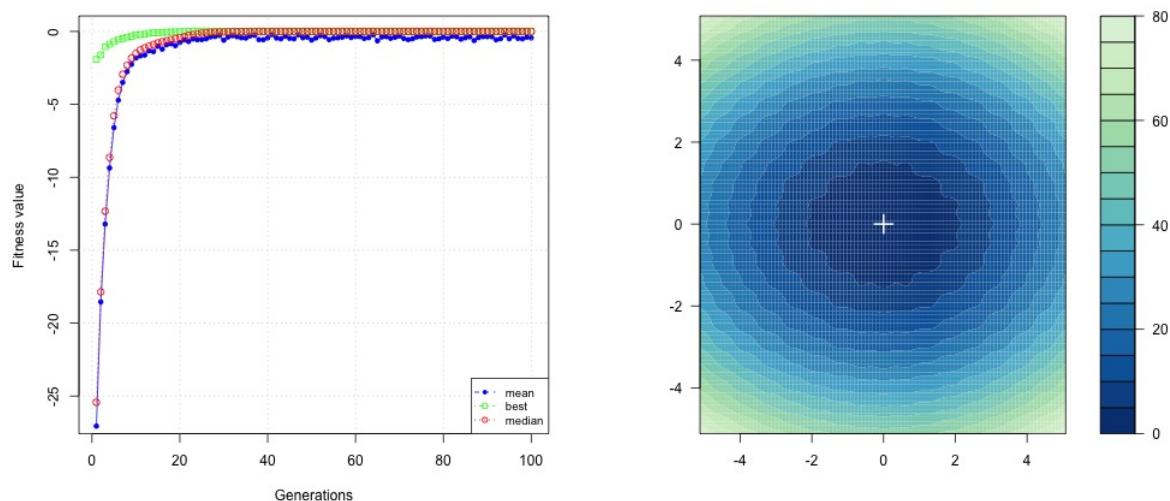
Rysunek 32: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0



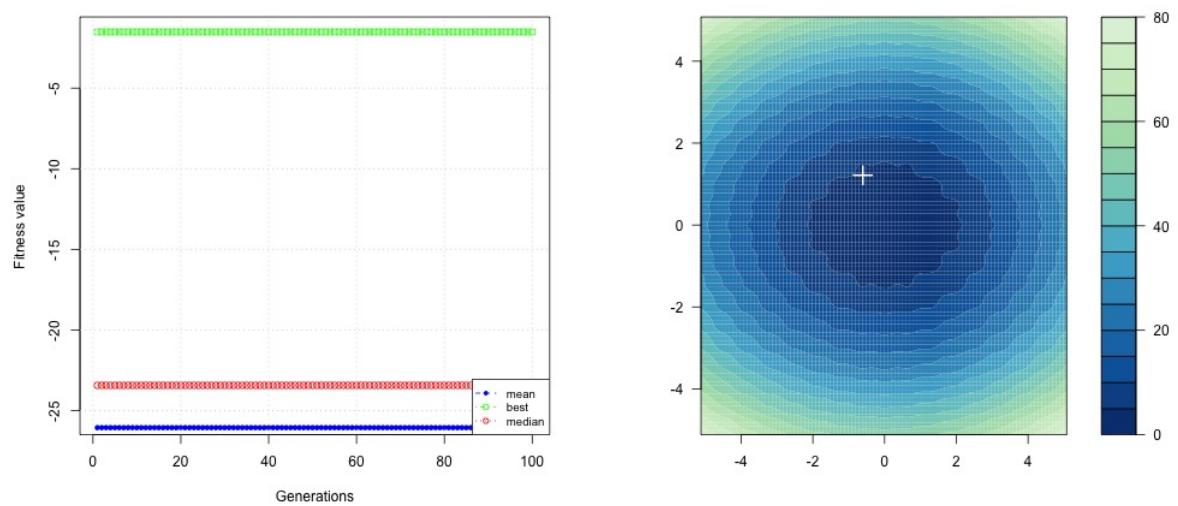
Rysunek 33: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.25



Rysunek 34: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.5



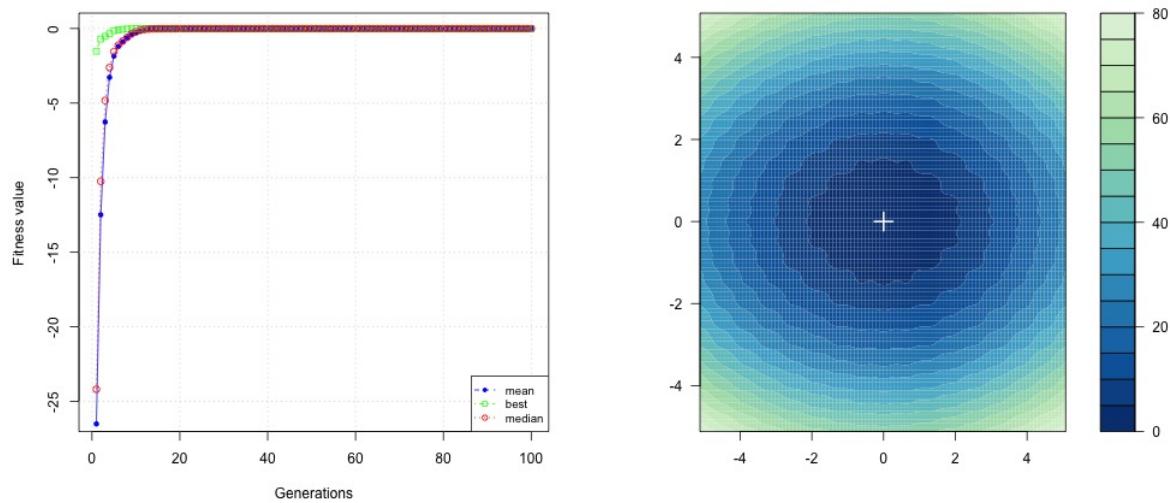
Rysunek 35: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.75



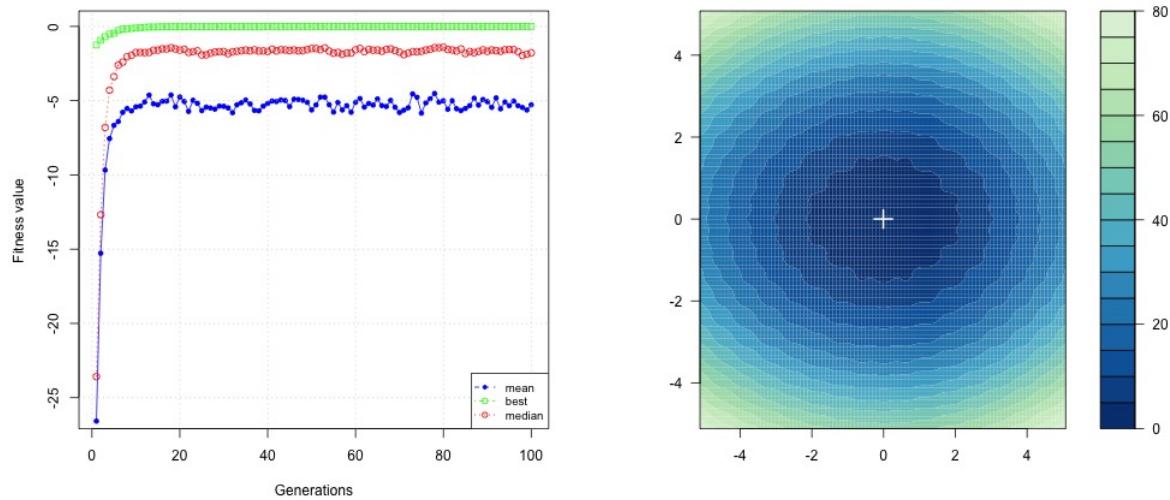
Rysunek 36: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e1

4.3.2 Modyfikacja parametru mutacji

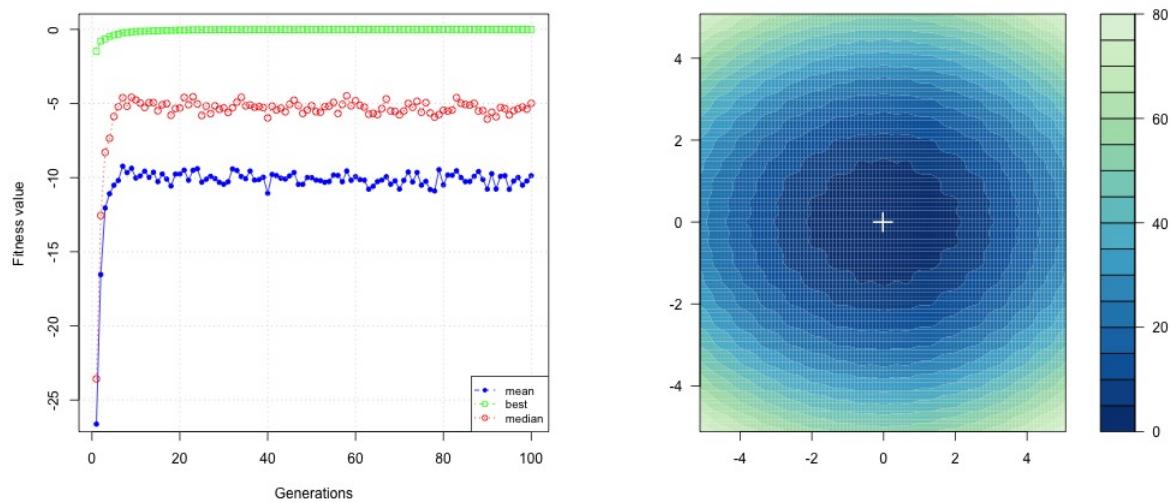
Zwiększenie parametru mutacji sprawia, że zarówno średnia jak i mediana zwiększają się wraz ze wzrostem parametru mutacji. Niezależnie od tego algorytm znajduje optymalne rozwiązanie stosunkowo szybko i po podobnej ilości pokoleń.



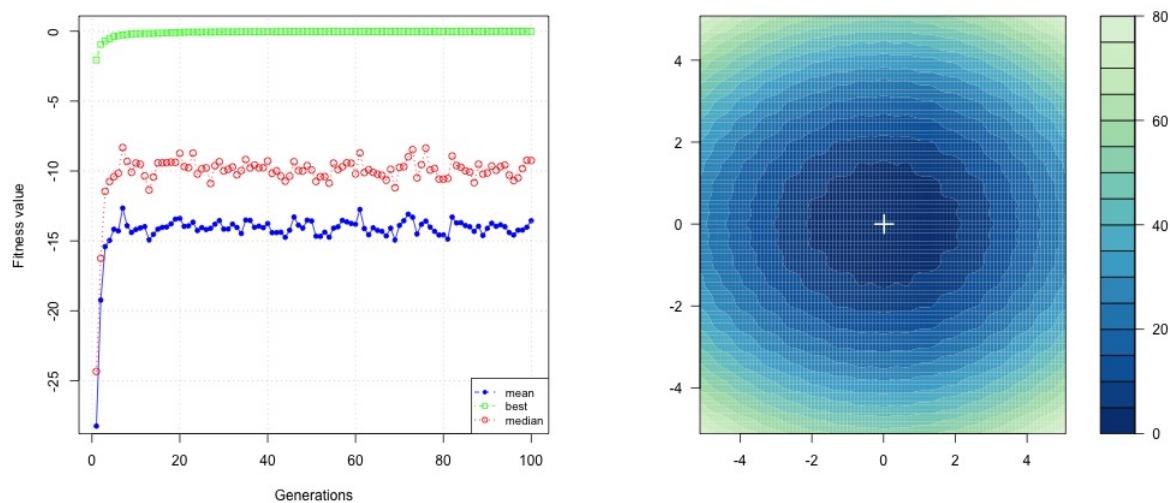
Rysunek 37: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0 e0.05



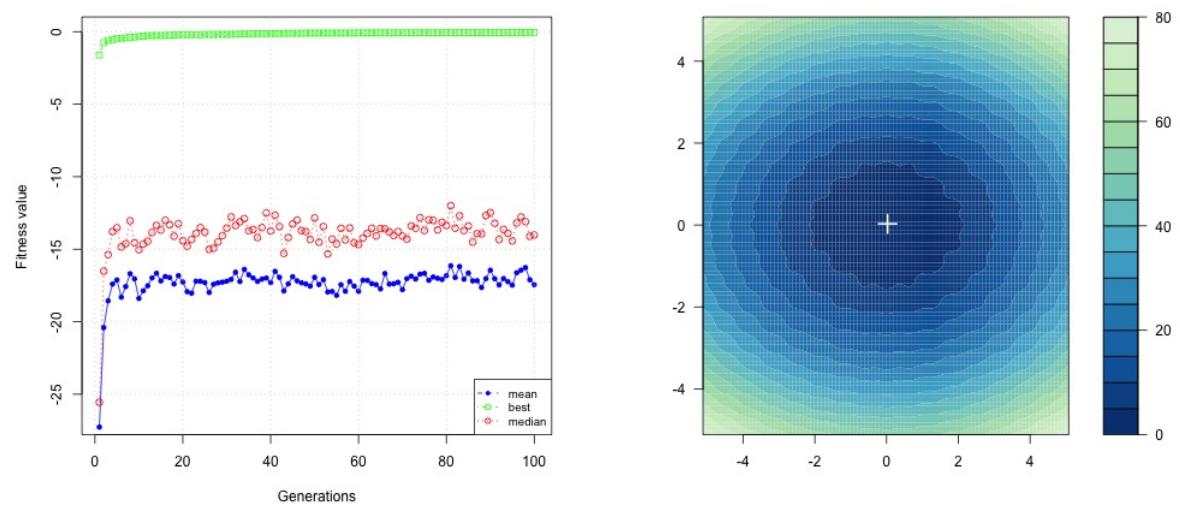
Rysunek 38: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.25 e0.05



Rysunek 39: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.5 e0.05



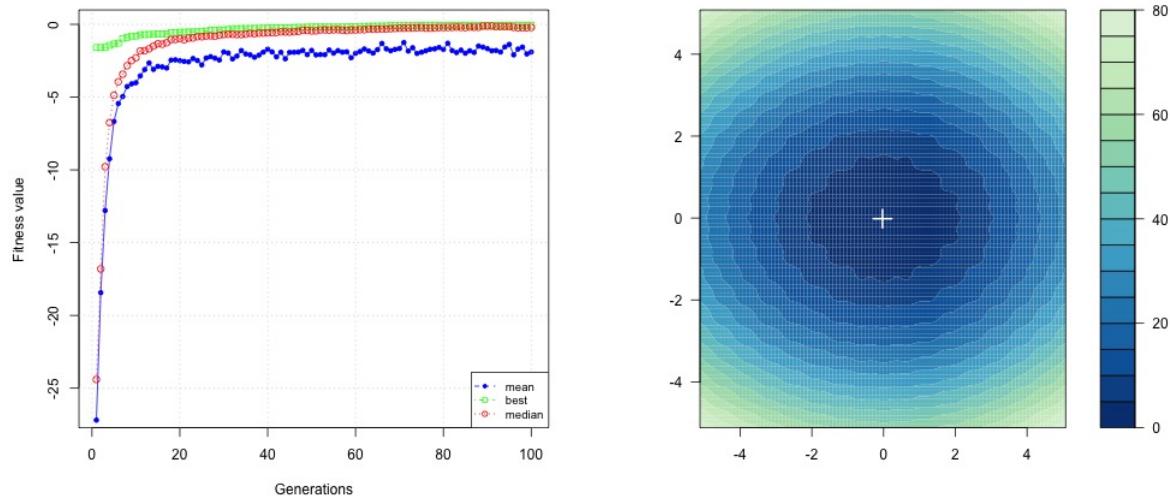
Rysunek 40: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.75 e0.05



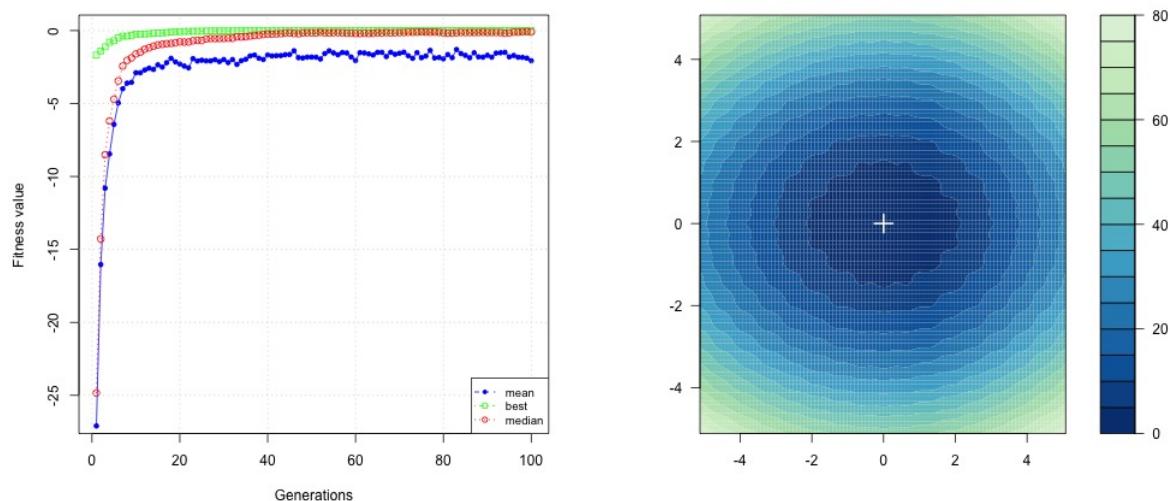
Rysunek 41: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m1 e0.05

4.3.3 Modyfikacja parametru krzyżowania

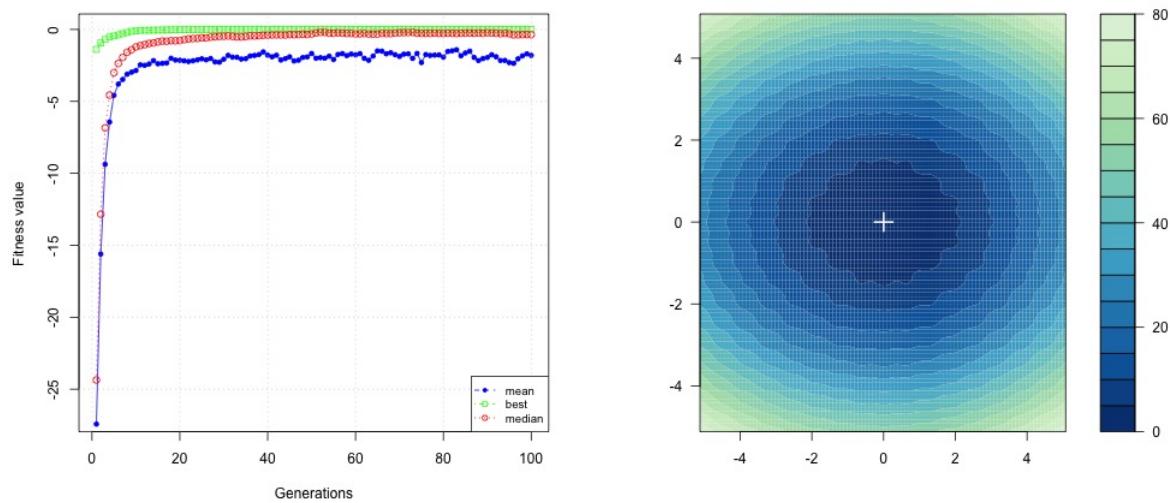
Modyfikacja parametru krzyżowania nie wpływa na rozwiązanie, ani na wartości średniej i mediany.



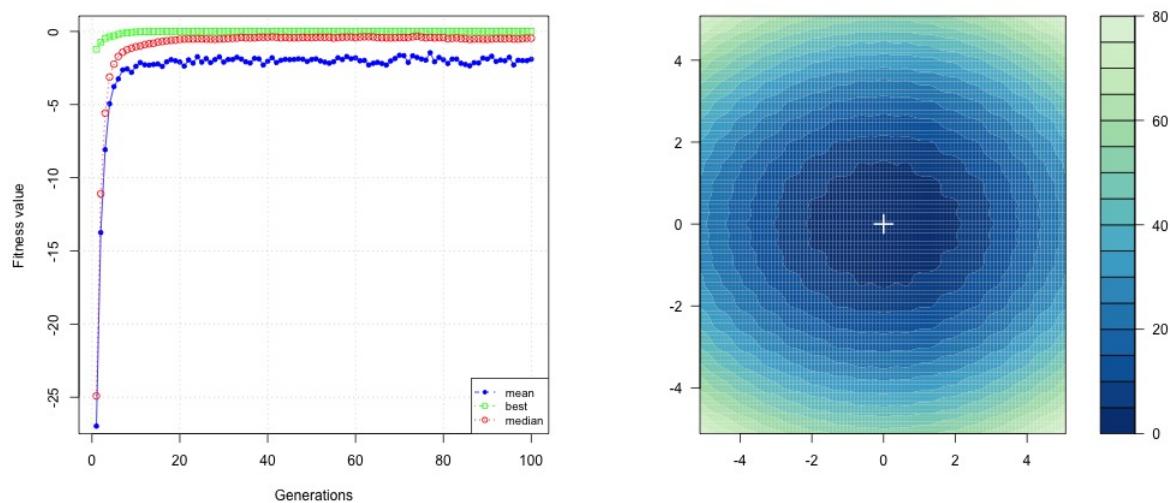
Rysunek 42: Test optymalizacji GA Bohachevsky1 p50 i100 c0 m0.1 e0.05



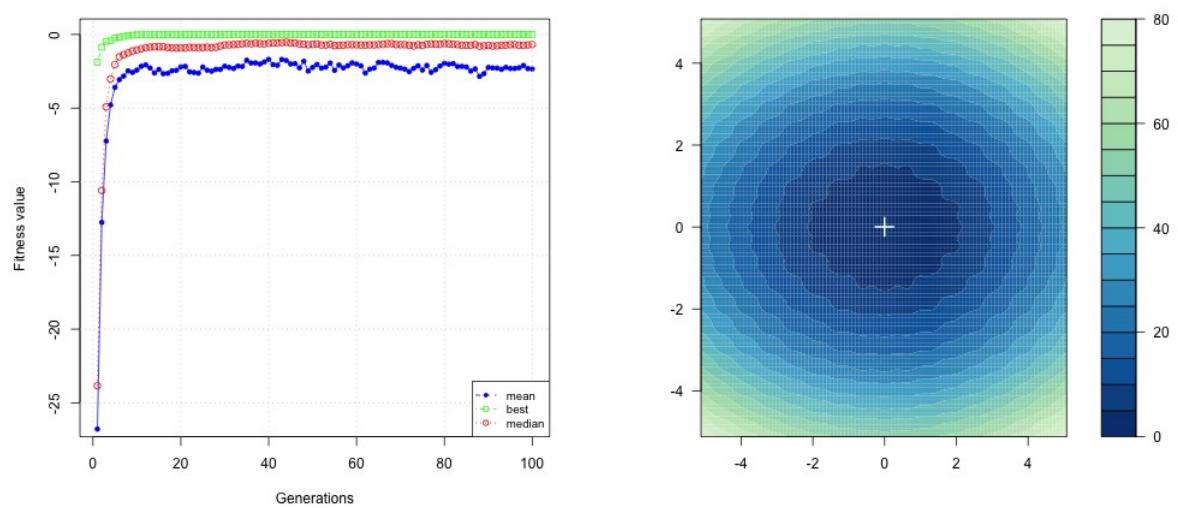
Rysunek 43: Test optymalizacji GA Bohachevsky1 p50 i100 c0.25 m0.1 e0.05



Rysunek 44: Test optymalizacji GA Bohachevsky1 p50 i100 c0.5 m0.1 e0.05



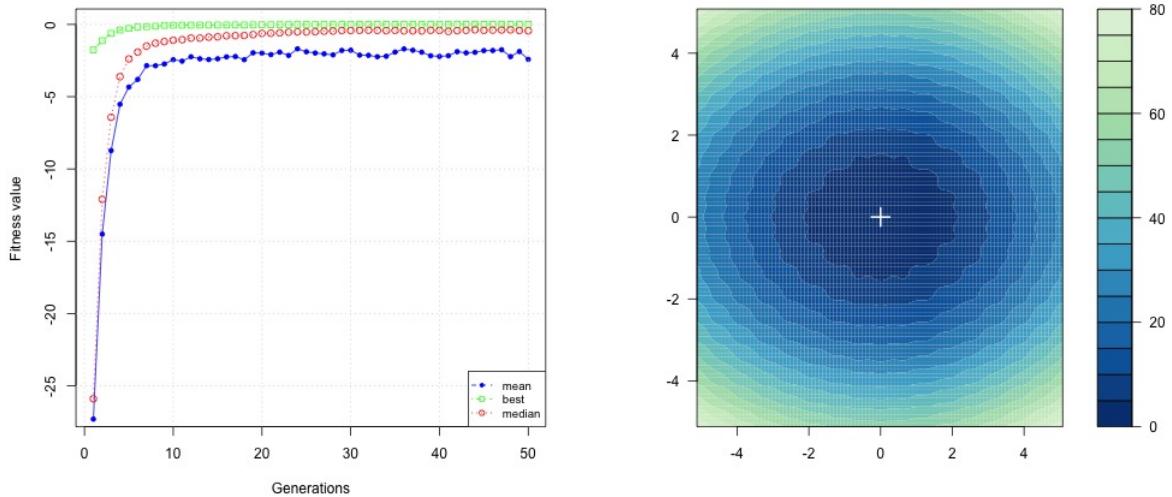
Rysunek 45: Test optymalizacji GA Bohachevsky1 p50 i100 c0.75 m0.1 e0.05



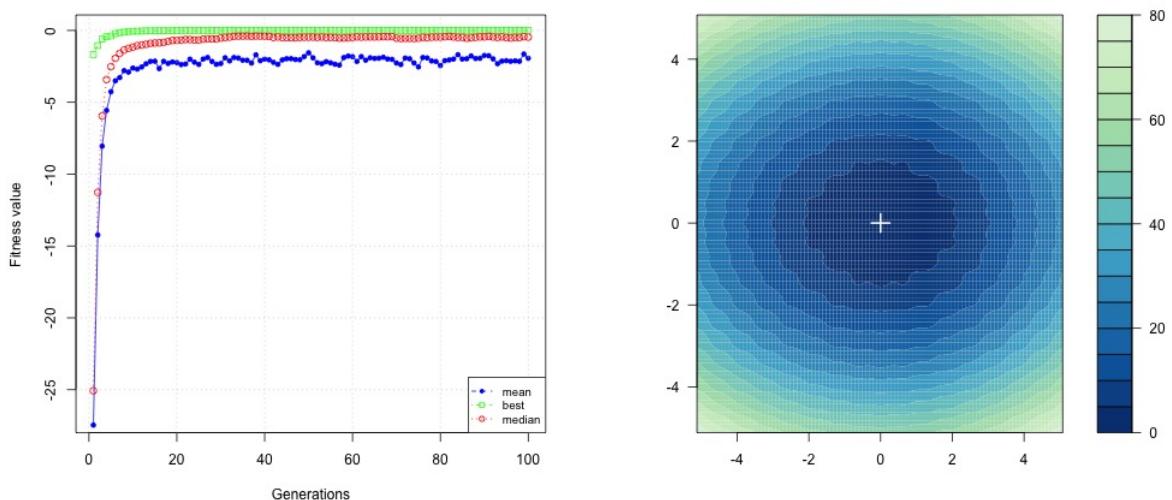
Rysunek 46: Test optymalizacji GA Bohachevsky1 p50 i100 c1 m0.1 e0.05

4.3.4 Modyfikacja parametru liczby iteracji

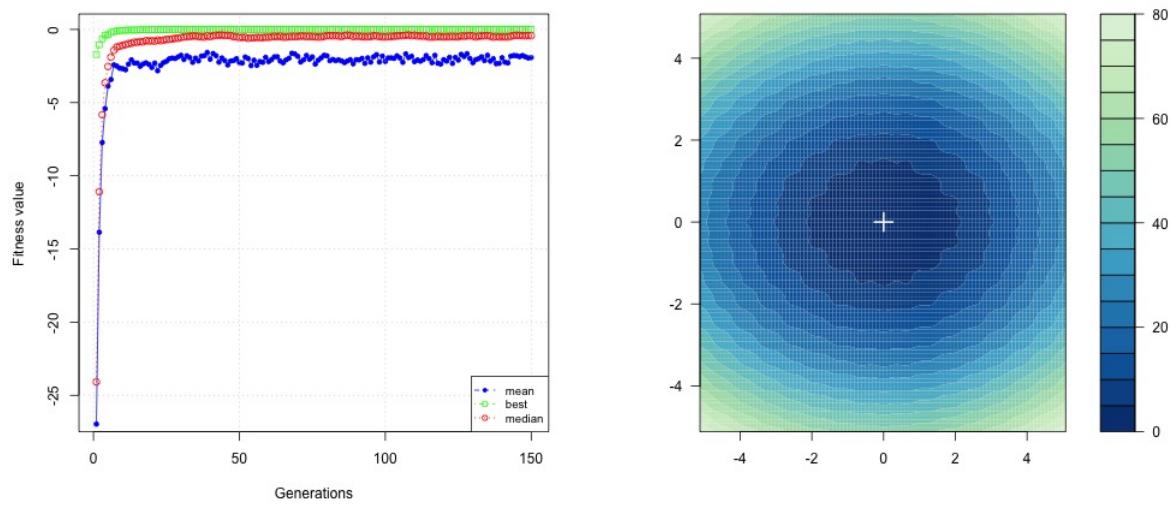
Modyfikacja parametru rozmiaru populacji nie wpływa na rozwiązanie, ani na wartości średniej i mediany.



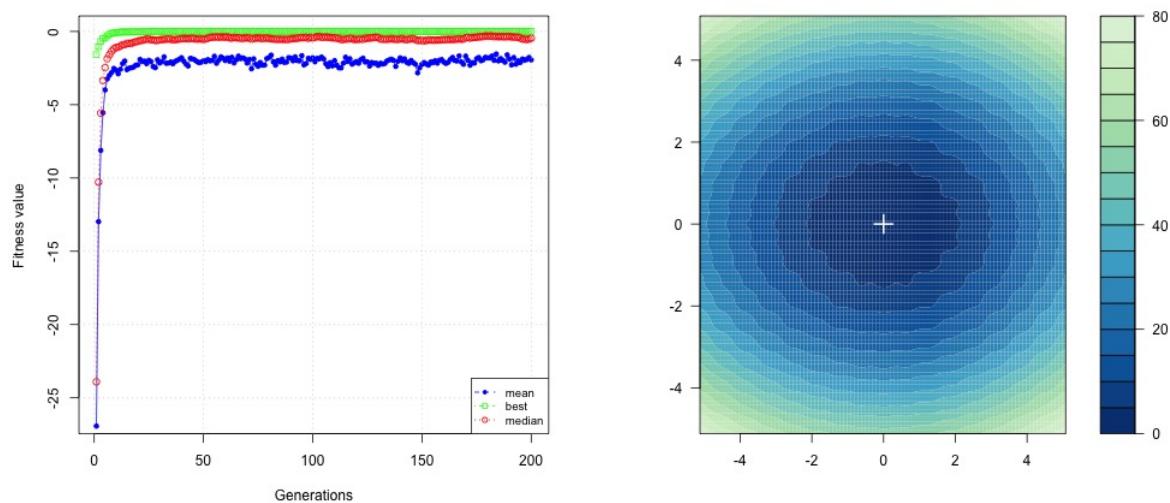
Rysunek 47: Test optymalizacji GA Bohachevsky1 p50 i50 c0.8 m0.1 e0.05



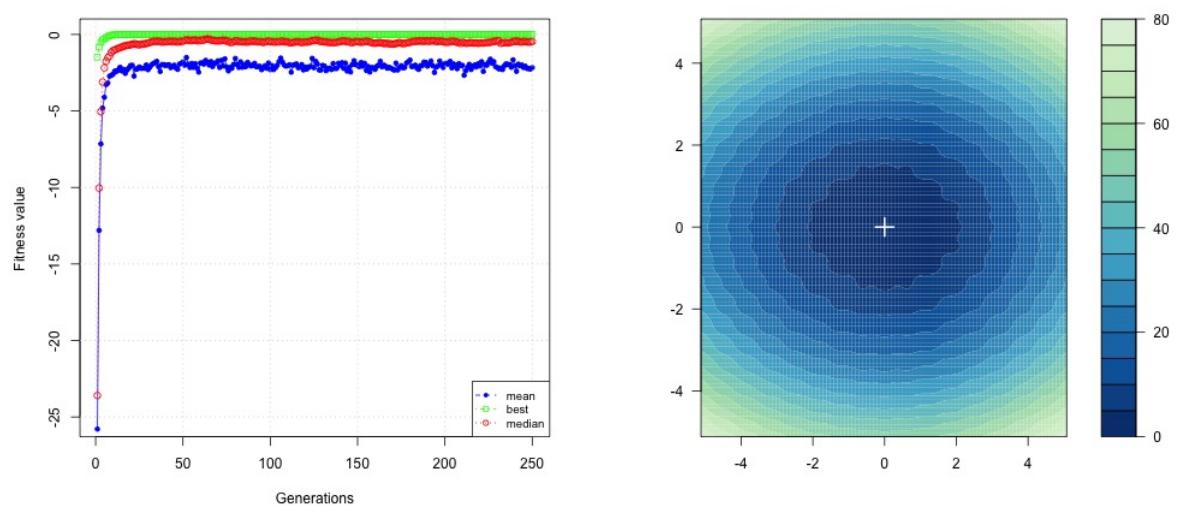
Rysunek 48: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.05



Rysunek 49: Test optymalizacji GA Bohachevsky1 p50 i150 c0.8 m0.1 e0.05



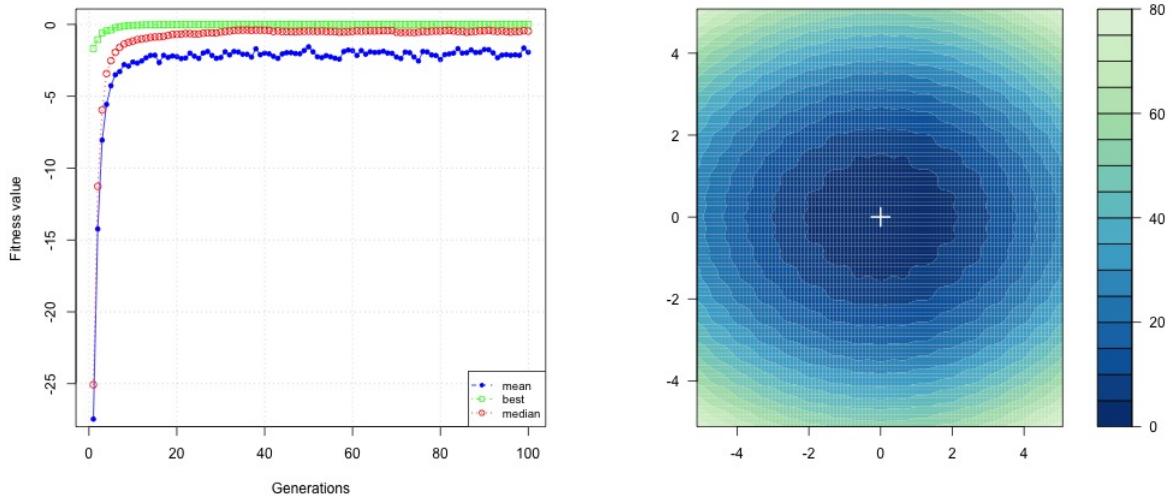
Rysunek 50: Test optymalizacji GA Bohachevsky1 p50 i200 c0.8 m0.1 e0.05



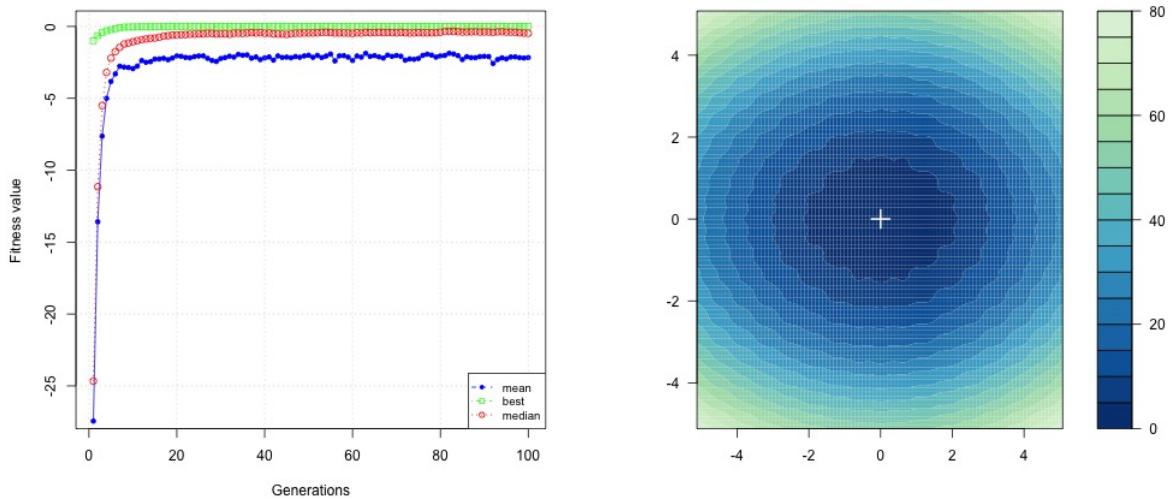
Rysunek 51: Test optymalizacji GA Bohachevsky1 p50 i250 c0.8 m0.1 e0.05

4.3.5 Modyfikacja parametru rozmiaru populacji

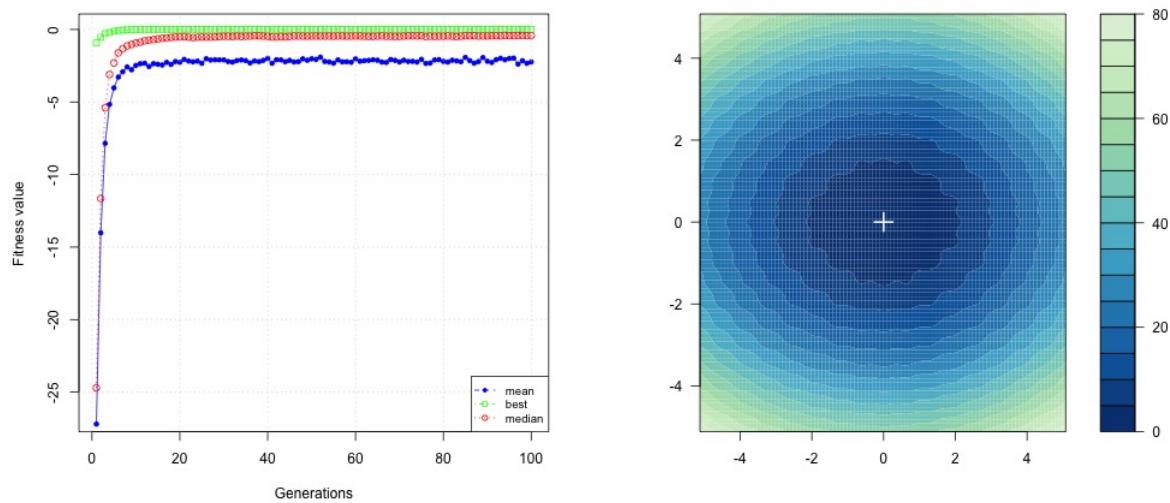
Modyfikacja parametru rozmiaru krzyżowania nie wpływa na rozwiązanie, ani na wartości średniej i mediany.



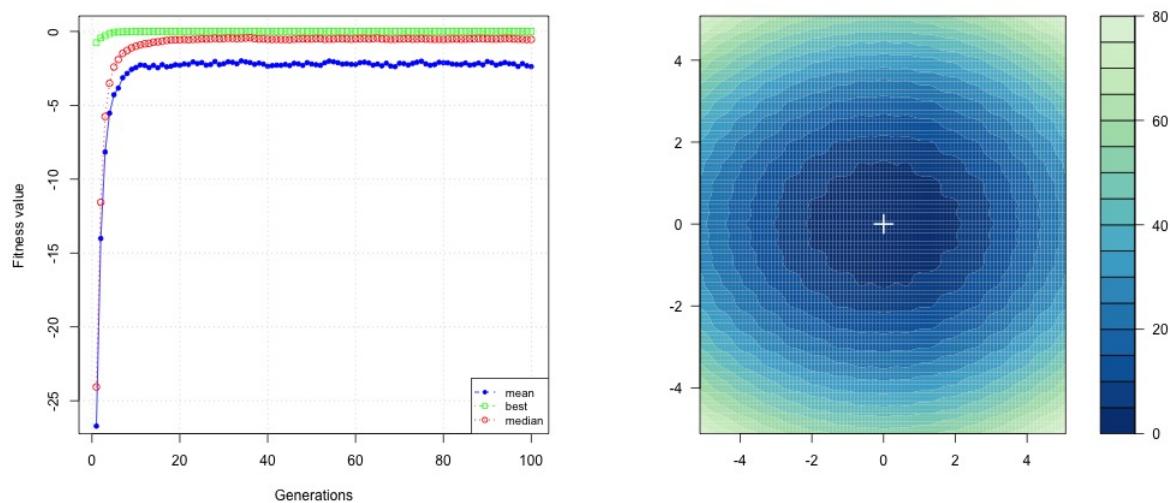
Rysunek 52: Test optymalizacji GA Bohachevsky1 p50 i100 c0.8 m0.1 e0.05



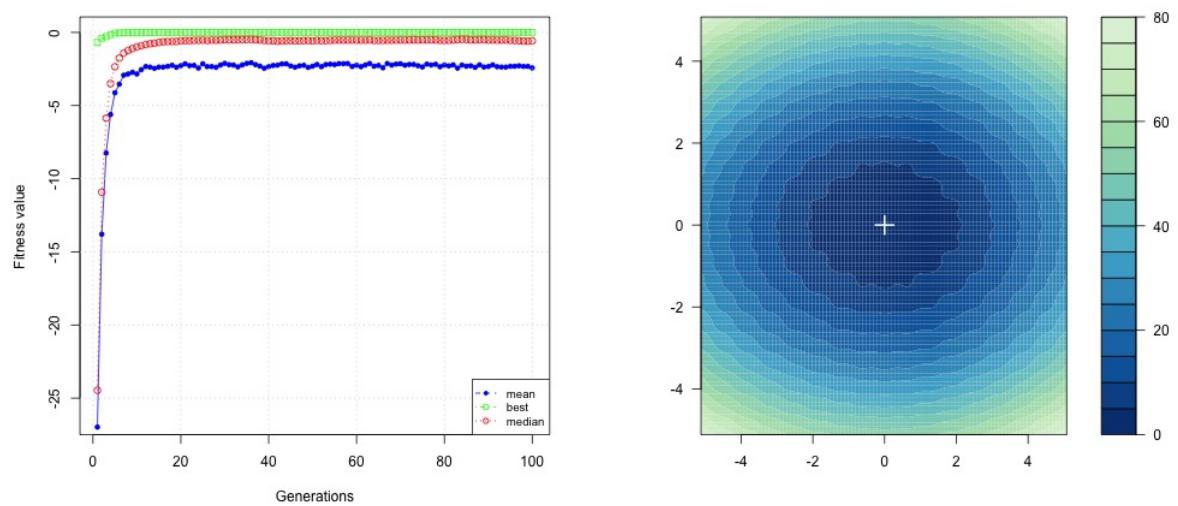
Rysunek 53: Test optymalizacji GA Bohachevsky1 p100 i100 c0.8 m0.1 e0.05



Rysunek 54: Test optymalizacji GA Bohachevsky1 p150 i100 c0.8 m0.1 e0.05



Rysunek 55: Test optymalizacji GA Bohachevsky1 p200 i100 c0.8 m0.1 e0.05



Rysunek 56: Test optymalizacji GA Bohachevsky1 p250 i100 c0.8 m0.1 e0.05

5 Funkcja Branina

5.1 Wzór analityczny

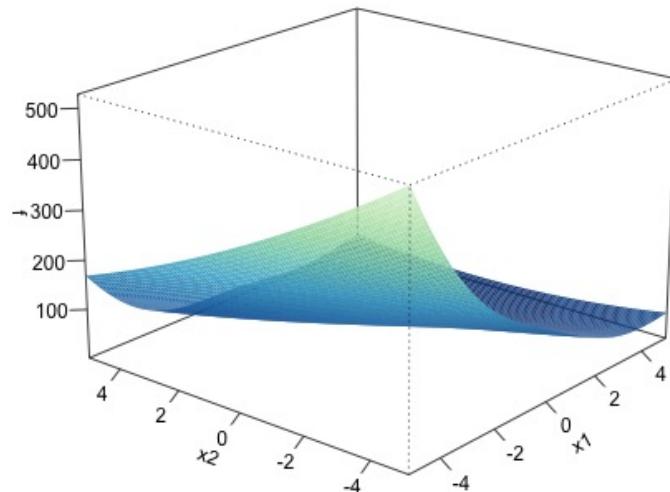
$$f(\mathbf{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t)\cos(x_1) + s$$

Rysunek 57: Wzór analityczny funkcji Branina

$$f(\mathbf{x}^*) = 0.397887, \text{ at } \mathbf{x}^* = (-\pi, 12.275), (\pi, 2.275) \text{ and } (9.42478, 2.475)$$

Rysunek 58: Minimum globalne funkcji Branina

5.2 Wykres w ustalonym przedziale zmiennych

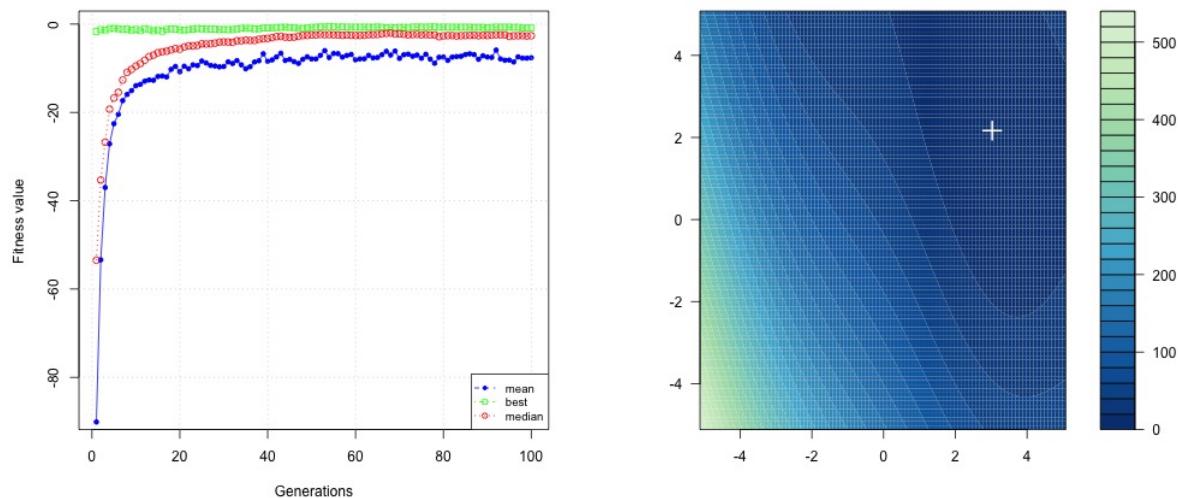


Rysunek 59: Wzór analityczny funkcji Branina

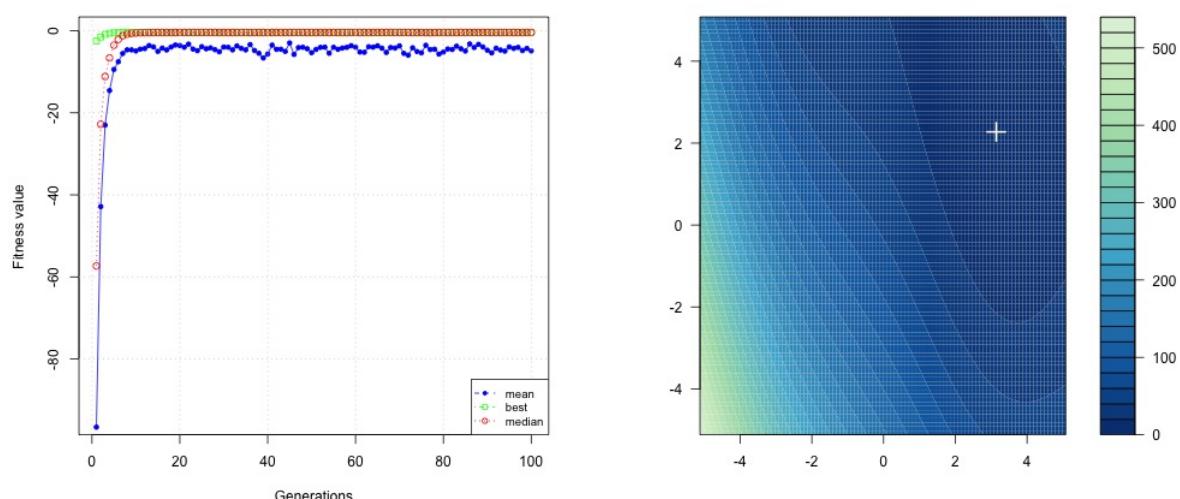
5.3 Optymalizacja

5.3.1 Modyfikacja parametru elitarności

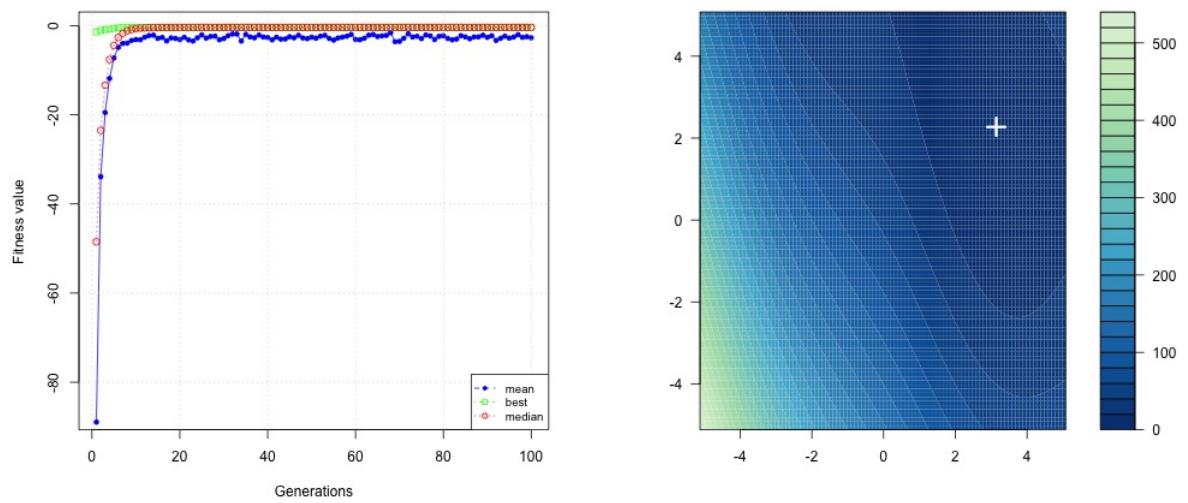
W przypadku braku populacji elitarnej algorytm potrzebował więcej czasu aby odnaleźć poprawny wynik, wskazuje na to wykres mediany. W przypadku wartości 0.25, 0.5 oraz 0.75 algorytm szybciej odnajdywał poprawne rozwiązanie. W momencie gdy cała populacja była oznaczana jako elitarna algorytm nie znajdował poprawnego rozwiązania, nie występowała również zmiana między kolejnymi generacjami.



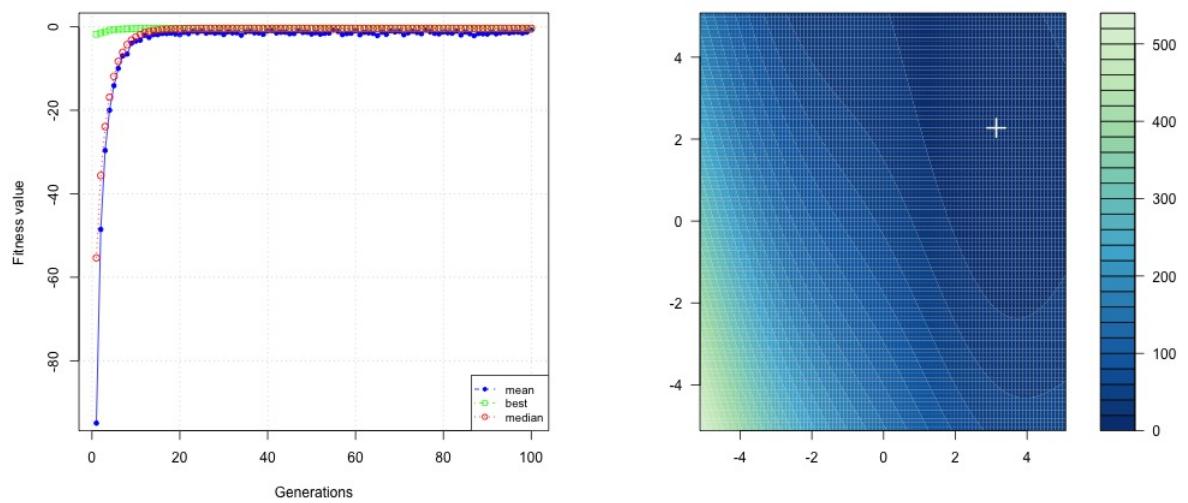
Rysunek 60: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0



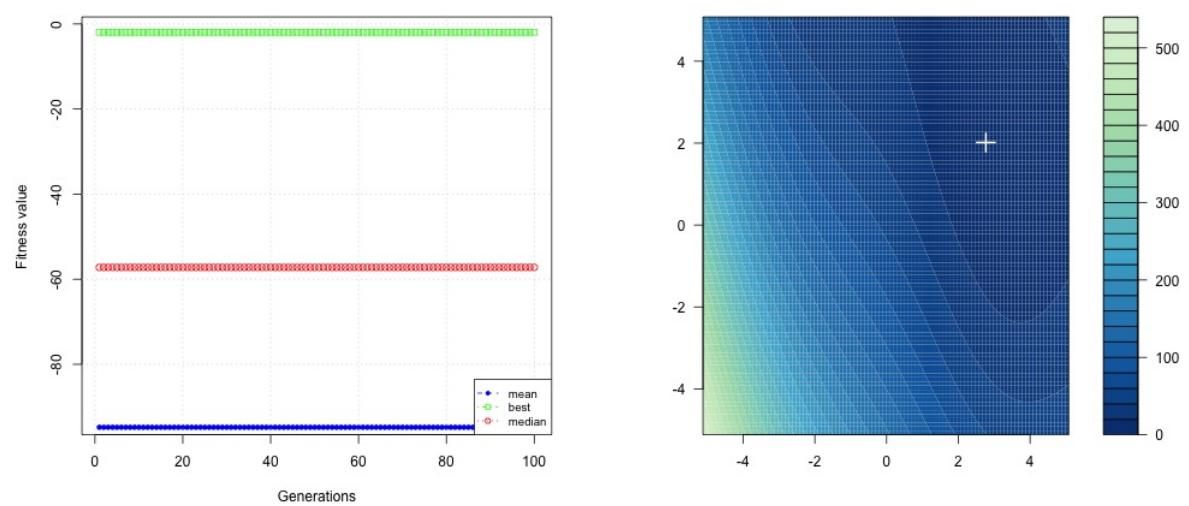
Rysunek 61: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.25



Rysunek 62: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.5



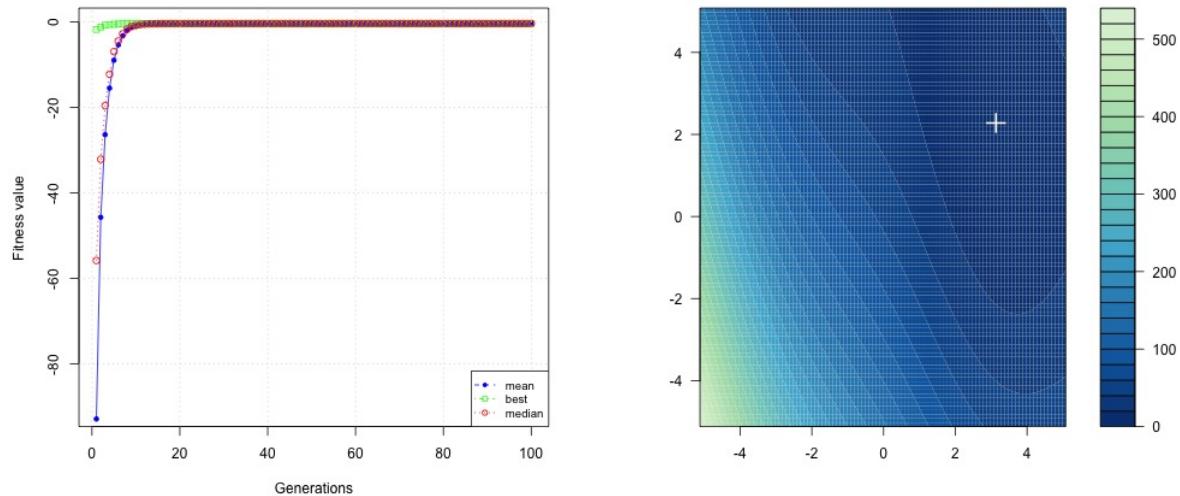
Rysunek 63: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.75



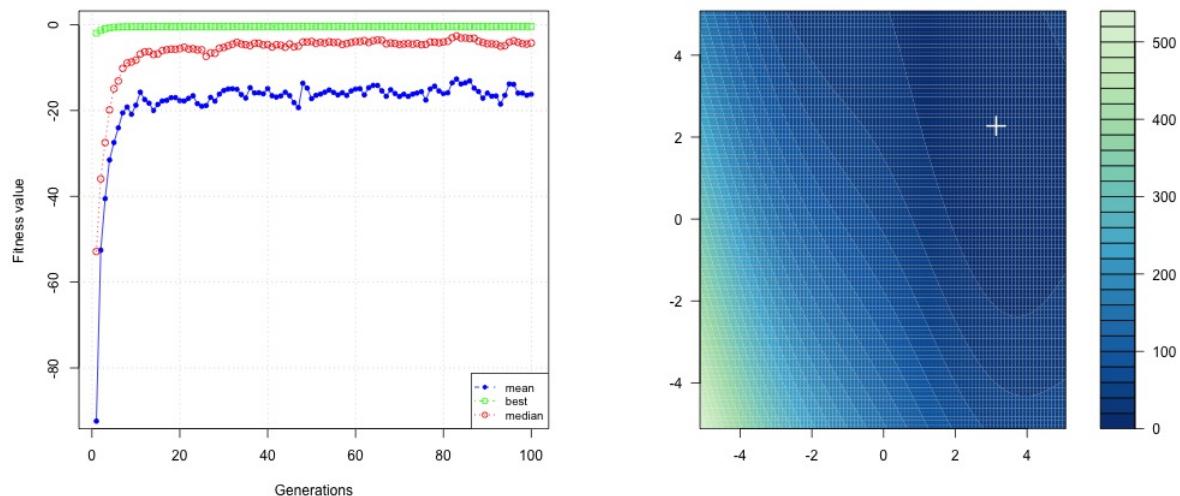
Rysunek 64: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e1

5.3.2 Modyfikacja parametru mutacji

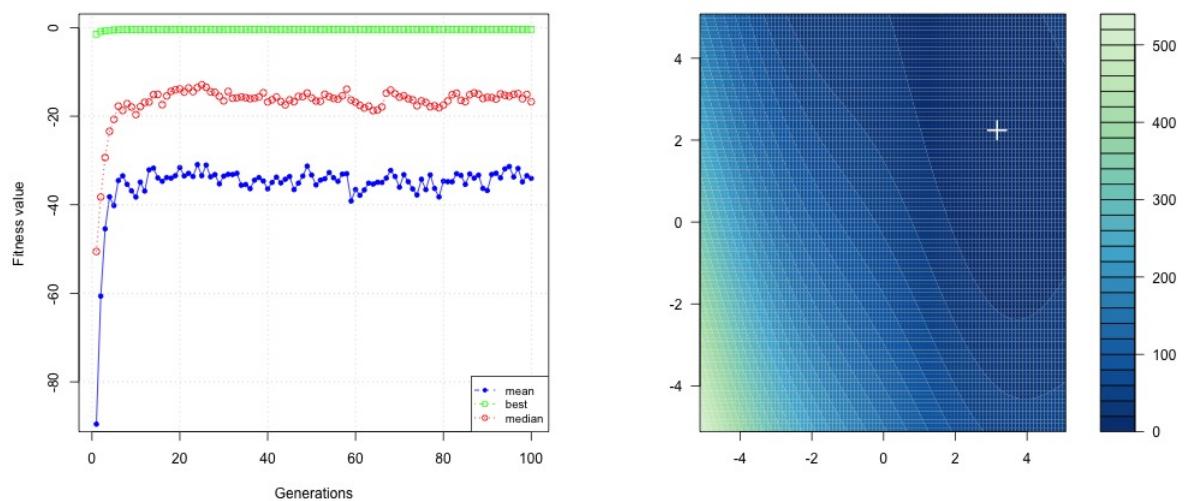
Algorytm bez wzgledu na zmianę parametru mutacji zawsze odnajdował poprawne rozwiązanie, im większa wartość parametru tym wykres mediany oraz średniej wskazywał na coraz większe odchylenia i niższe wyniki.



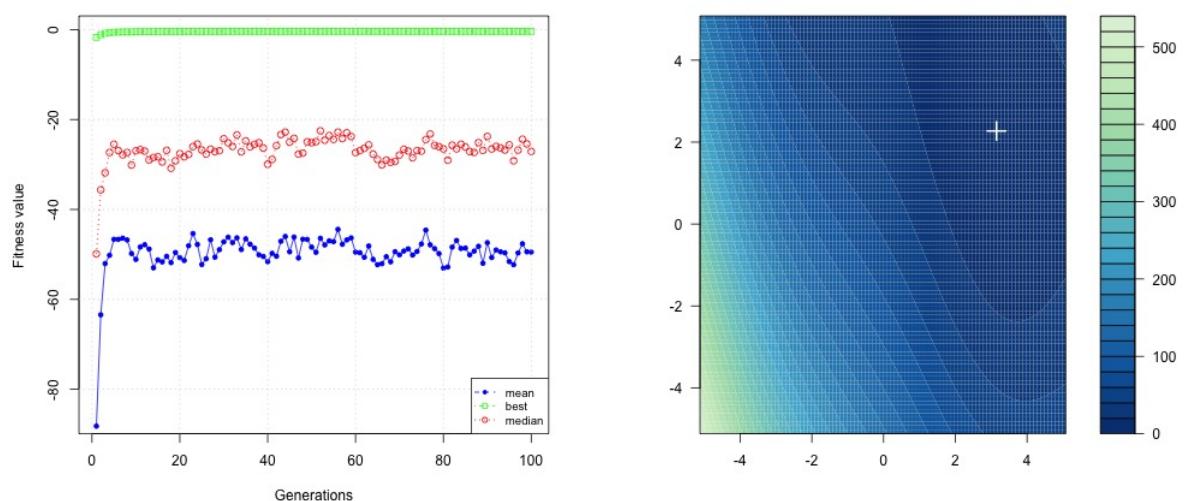
Rysunek 65: Test optymalizacji GA Branin p50 i100 c0.8 m0 e0.05



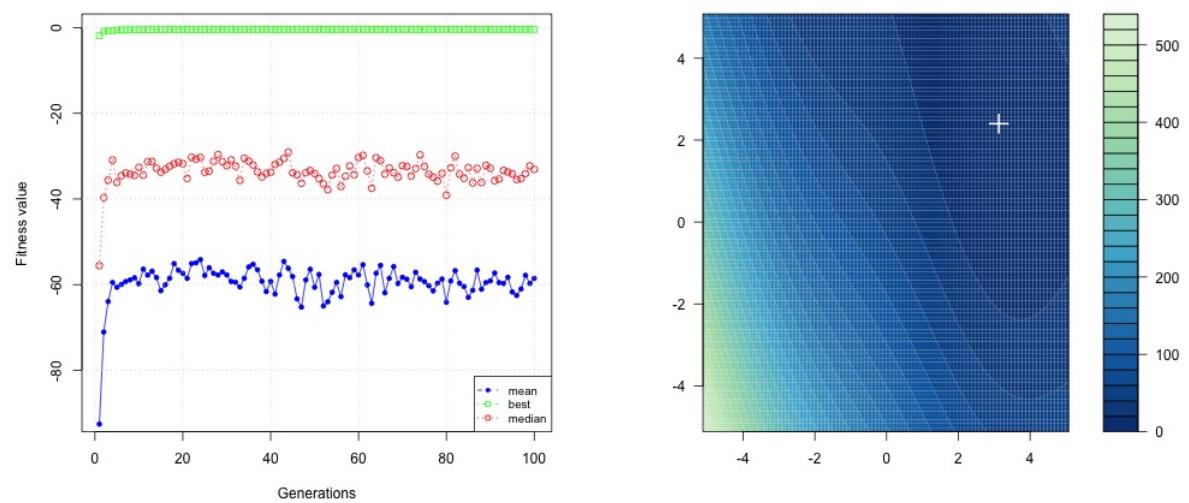
Rysunek 66: Test optymalizacji GA Branin p50 i100 c0.8 m0.25 e0.05



Rysunek 67: Test optymalizacji GA Branin p50 i100 c0.8 m0.5 e0.05



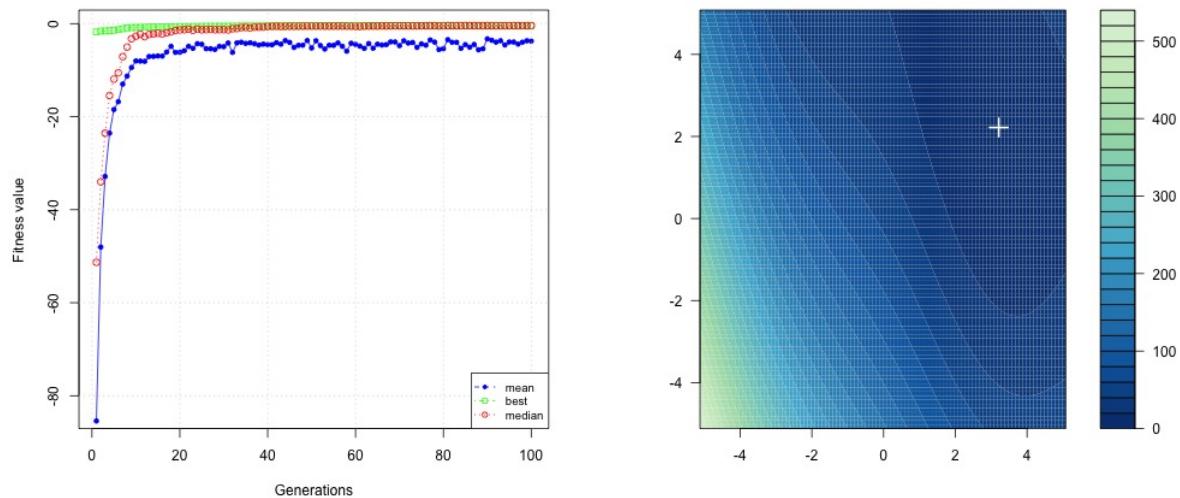
Rysunek 68: Test optymalizacji GA Branin p50 i100 c0.8 m0.75 e0.05



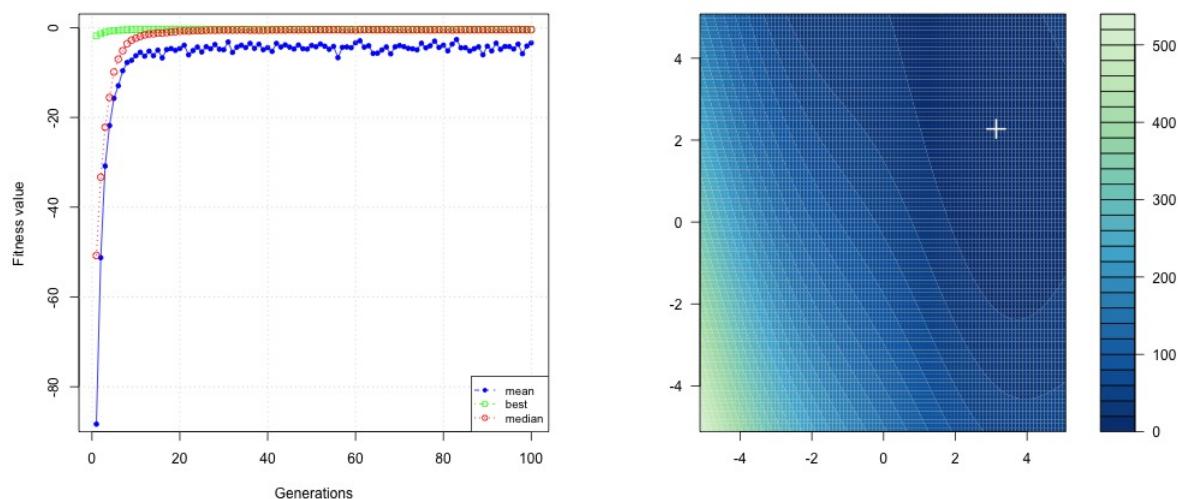
Rysunek 69: Test optymalizacji GA Branin p50 i100 c0.8 m1 e0.05

5.3.3 Modyfikacja parametru krzyżowania

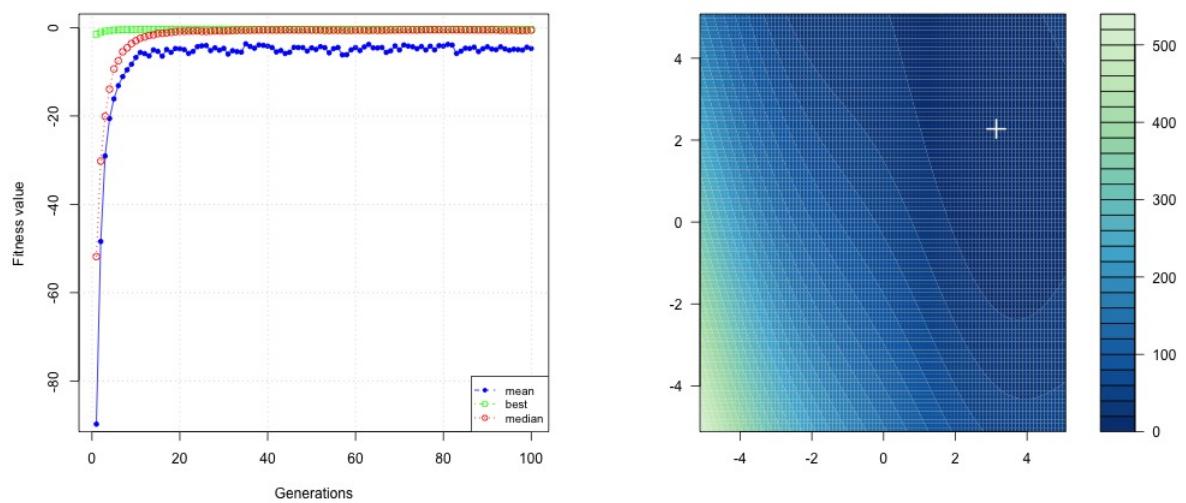
W przypadku braku krzyżowania funkcja potrzebuje większej ilości iteracji aby znaleźć poprawne rozwiązanie. Wyższy wskaźnik ma w tym przypadku pozytywny wpływ na szybkość odnajdywanego rozwiązania.



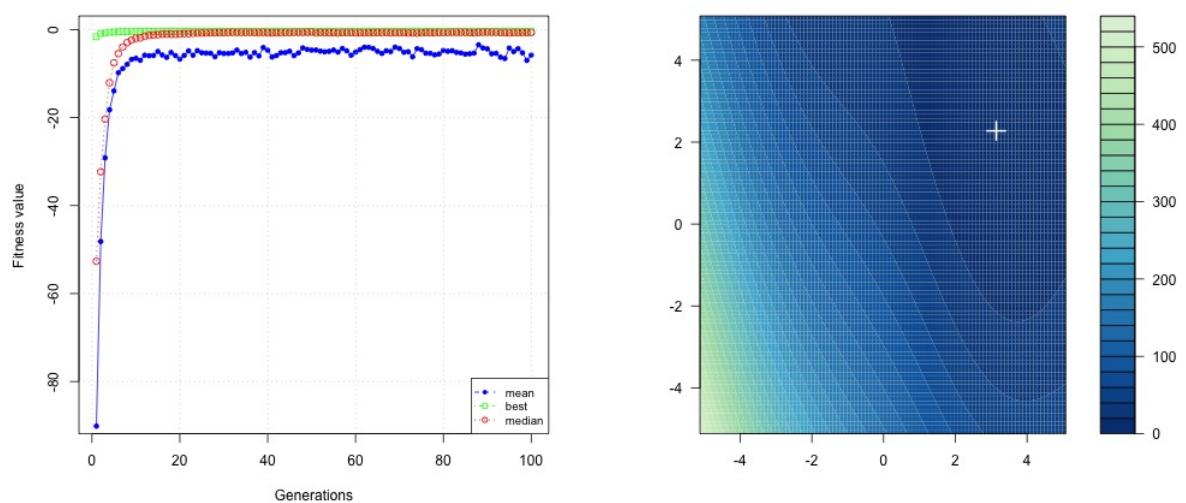
Rysunek 70: Test optymalizacji GA Branin p50 i100 c0 m0.1 e0.05



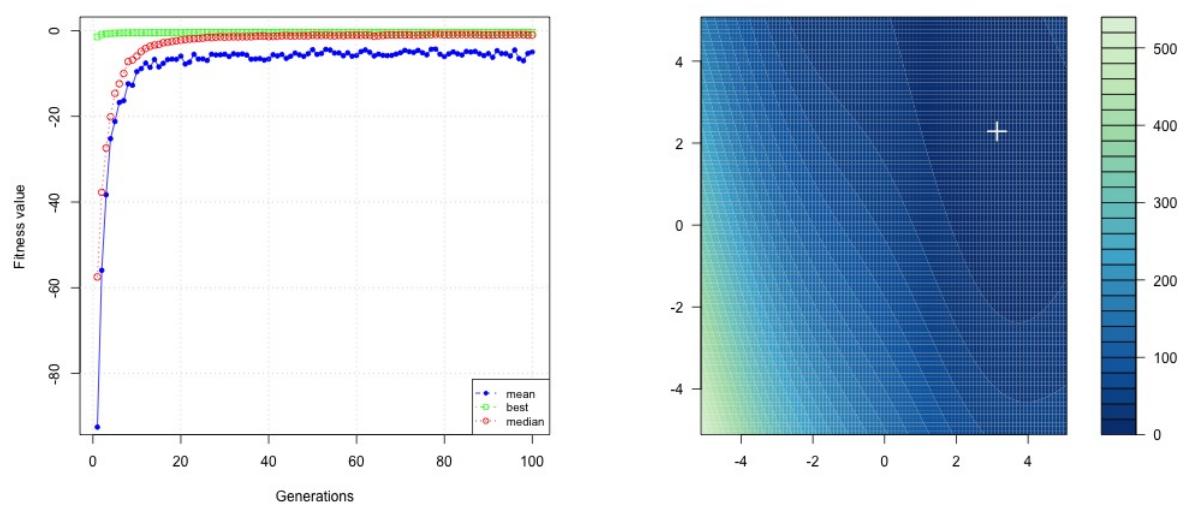
Rysunek 71: Test optymalizacji GA Branin p50 i100 c0.25 m0.1 e0.05



Rysunek 72: Test optymalizacji GA Branin p50 i100 c0.5 m0.1 e0.05



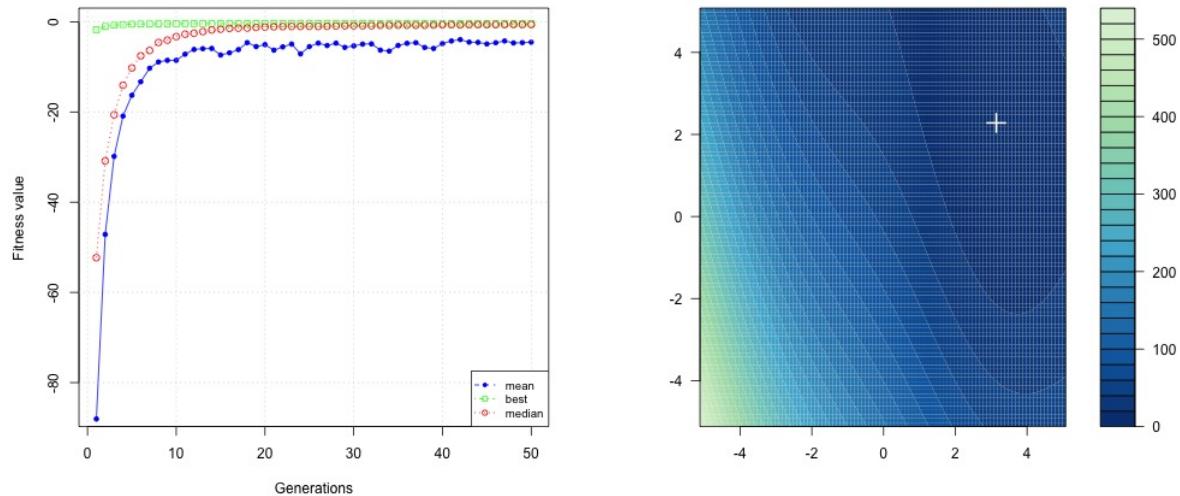
Rysunek 73: Test optymalizacji GA Branin p50 i100 c0.75 m0.1 e0.05



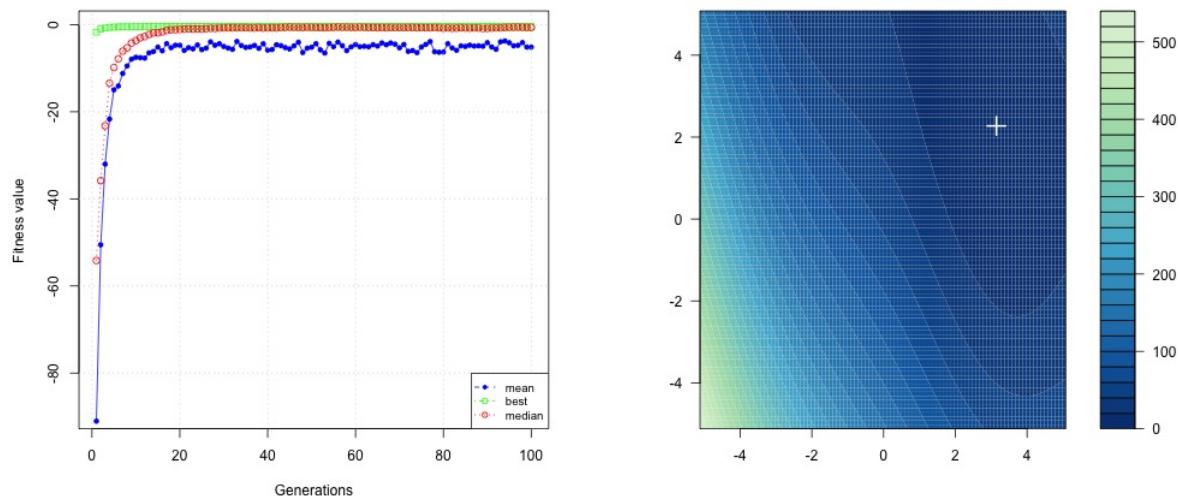
Rysunek 74: Test optymalizacji GA Branin p50 i100 c1 m0.1 e0.05

5.3.4 Modyfikacja parametru liczby iteracji

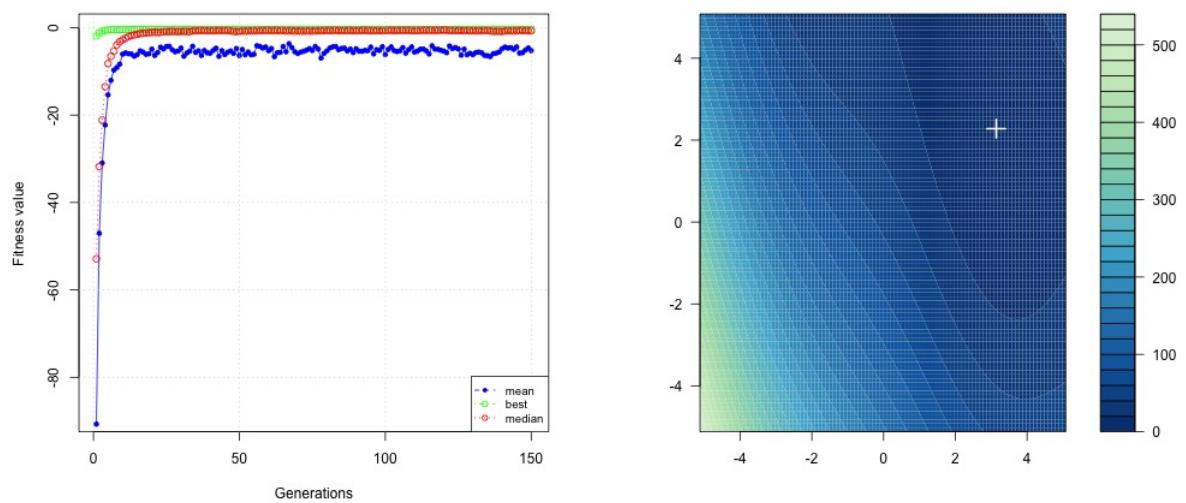
Zmiana liczby iteracji w przypadku funkcji Branin nie wpływa na jakość rozwiązania. Nie występują żadne anomalie między zmianami parametru. Funkcja w każdym przypadku zachowuje się identycznie.



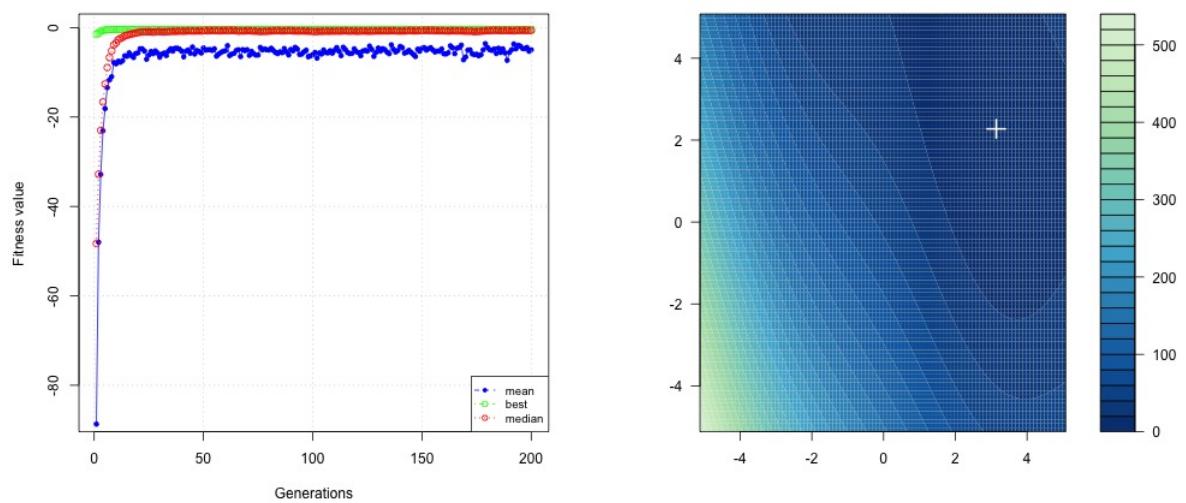
Rysunek 75: Test optymalizacji GA Branin p50 i50 c0.8 m0.1 e0.05



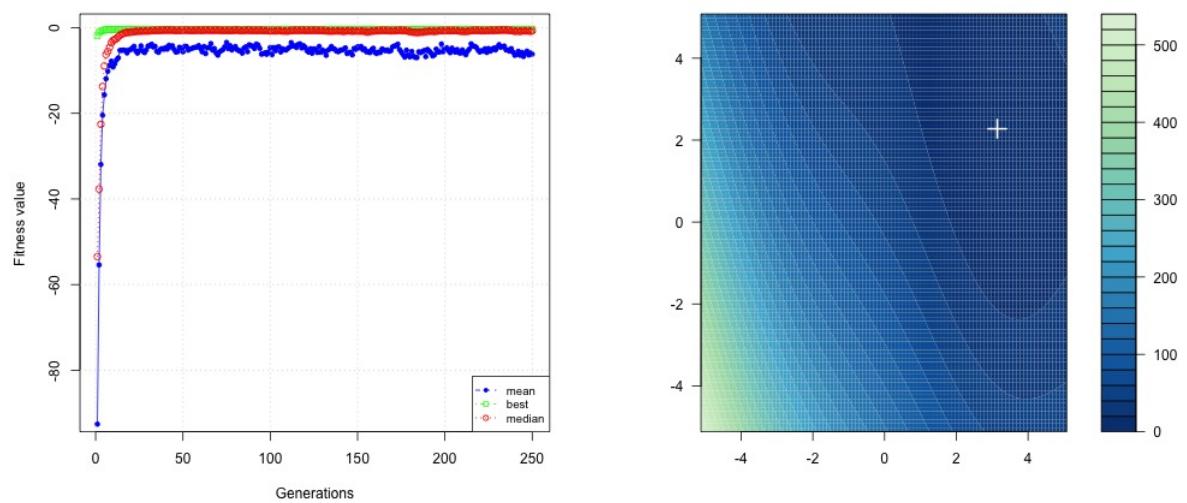
Rysunek 76: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.05



Rysunek 77: Test optymalizacji GA Branin p50 i150 c0.8 m0.1 e0.05



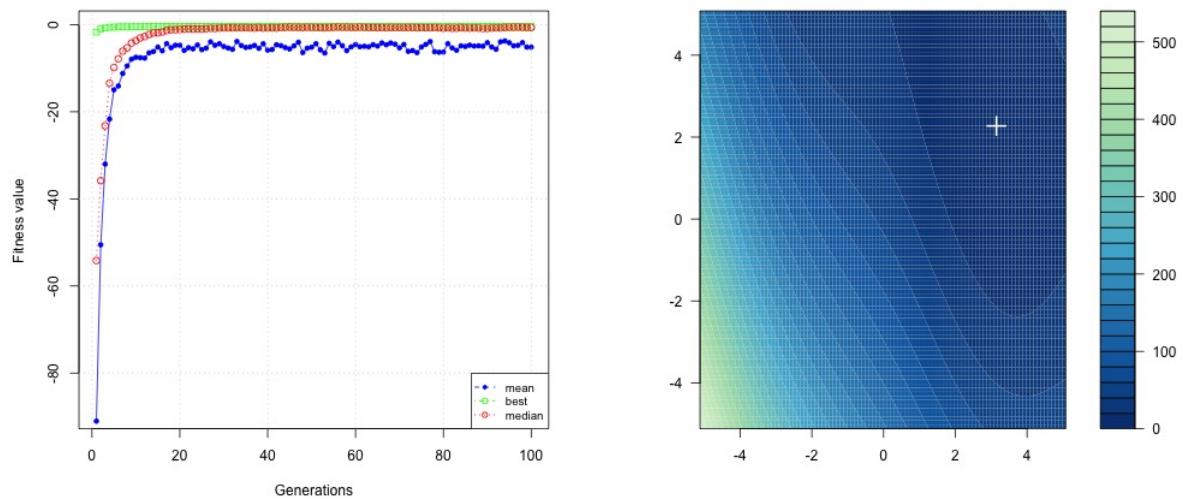
Rysunek 78: Test optymalizacji GA Branin p50 i200 c0.8 m0.1 e0.05



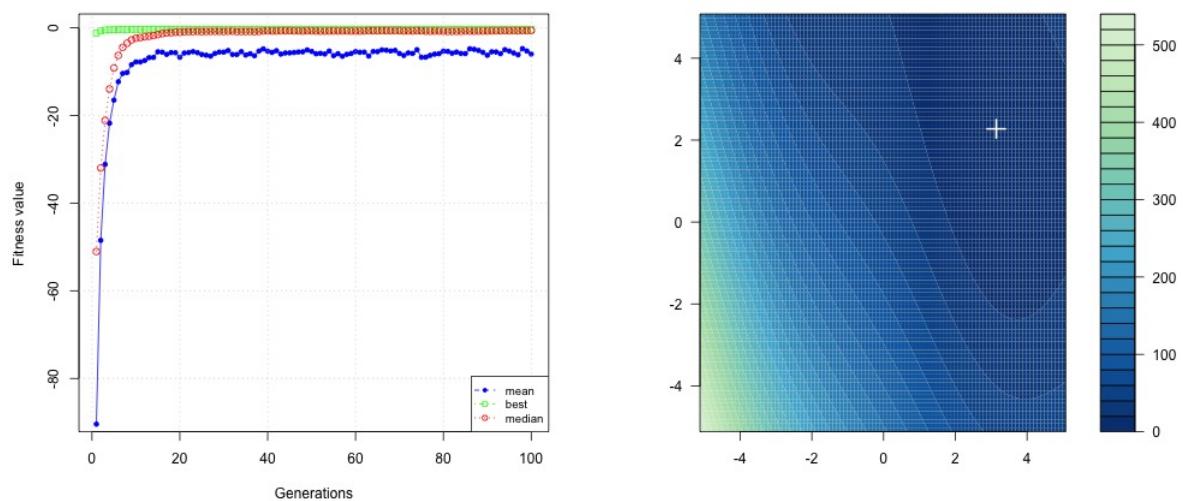
Rysunek 79: Test optymalizacji GA Branin p50 i250 c0.8 m0.1 e0.05

5.3.5 Modyfikacja parametru rozmiaru populacji

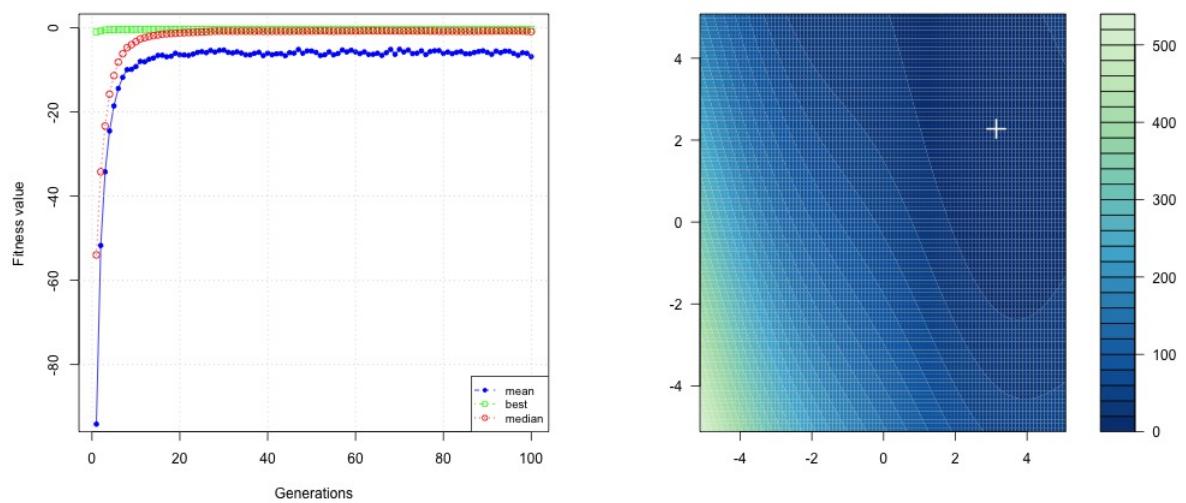
Funkcja Branin nie posiada wielu lokalnych minimów tak jak w jest to w przypadku funkcji Schubert. Dlatego algorytm znajduje optymalne rozwiązanie zawsze i wskazania mediany oraz średniej nie wykazują aby algorytm próbował znaleźć globalnego minimum w miejscu gdzie występuje lokalne minimum.



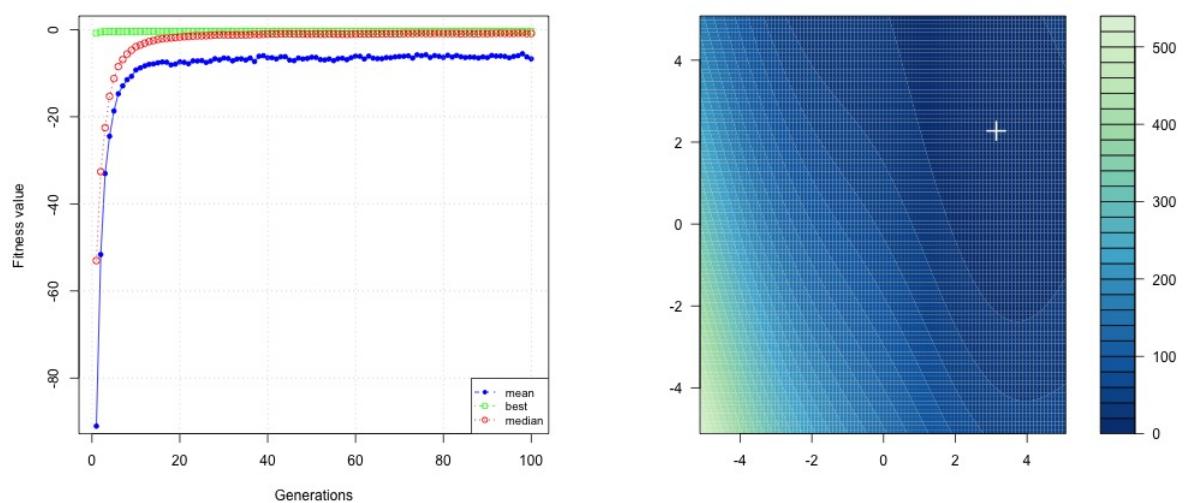
Rysunek 80: Test optymalizacji GA Branin p50 i100 c0.8 m0.1 e0.05



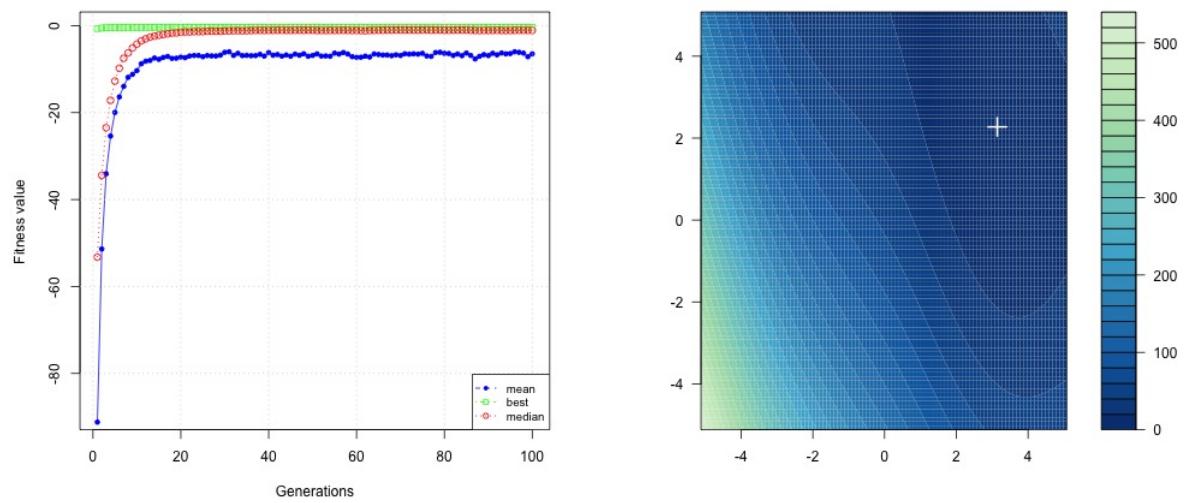
Rysunek 81: Test optymalizacji GA Branin p100 i100 c0.8 m0.1 e0.05



Rysunek 82: Test optymalizacji GA Branin p150 i100 c0.8 m0.1 e0.05



Rysunek 83: Test optymalizacji GA Branin p200 i100 c0.8 m0.1 e0.05



Rysunek 84: Test optymalizacji GA Branin p250 i100 c0.8 m0.1 e0.05

6 Wnioski