

## Sprawozdanie z pracowni specjalistycznej

### *Sztuczna inteligencja*

Ćwiczenie numer: 5

Temat: **Obliczenia ewolucyjne**

Wykonujący ćwiczenie: **Aleksander Kalinowski**

Studia dzienne

Kierunek: Informatyka

Semestr: IV

Grupa zajęciowa: PS7

Prowadzący ćwiczenie: mgr inż. Dariusz Jankowski

Data wykonania ćwiczenia:  
15.05.2021

## 1. Treść zadań:

Wykorzystując szkielet skryptu z zadania 1.5.3 znajdź maksimum funkcji określonej wzorem:

a)  $f(x, y) = (x^2 - y^2) - (1 - x)^2$ ,  $x, y \in <-2, 2>$

b)  $f(x, y) = x^2 + \sin(y)$ ,  $x \in <0, 1>$ ,  $y \in <0, 3.14>$

c)  $f(x, y) = \sin(3x + y) \times \sin(y - x)x$ ,  $x, y \in <-5, 5>$

Jeżeli reszta z dzielenia nr Twojego indeksu przez 3 wynosi 1 znajdź maksimum funkcji z punktu a), jeżeli 2 z punktu b), jeżeli 0 z punktu c).

Przeprowadź analizę dokładności uzyskanego rozwiązania w zależności od prawdopodobieństwa krzyżowania, wielkości populacji, liczby iteracji oraz prawdopodobieństwa mutacji. W oparciu o przeprowadzone eksperymenty określ optymalne parametry algorytmu genetycznego.

W moim przypadku należy wykonać podpunkt c.

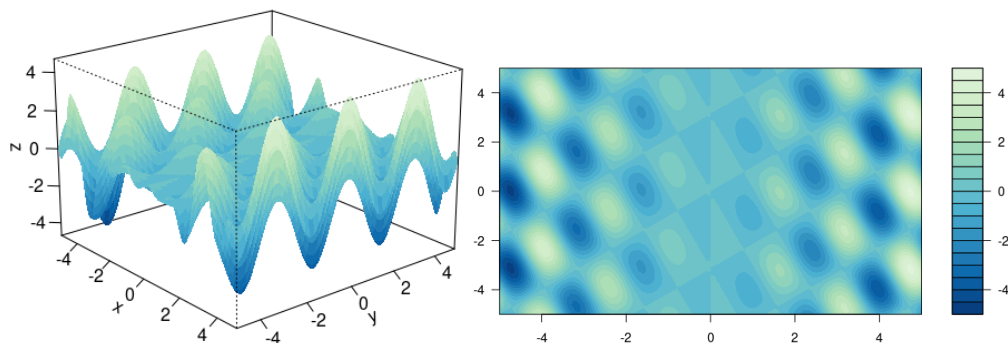
## 2. Wyniki i analiza:

Ćwiczenie rozpocząłem od wykonania algorytmu w języku programowania R w programie Rstudio. Kod wygląda następująco:

```
1 library(GA)
2
3 x <- y <- seq(-5, 5, by = 0.1)
4 f <- function(x, y)
5 {
6   sin(3*x+y)*sin(y-x)*x
7 }
8
9 z <- outer(x, y, f)
10 persp3D(x, y, z, theta = 50, phi = 20, col.palette = bl2gr.colors)
11 filled.contour(x, y, z, color.palette = bl2gr.colors)
12
13 GA <- ga(type = "real-valued", fitness = function(x) f(x[1], x[2]),
14         lower = c(-5, -5), upper = c(5, 5),
15         popSize = 10, maxiter = 6)
16 summary(GA)
17 plot(GA)
18 abline(v = GA@solution, lty = 3)|
```

1.Implementacja kodu w języku R

Aby odnaleźć optymalne rozwiązanie należy obejrzeć otrzymane wykresy oraz wyniki:

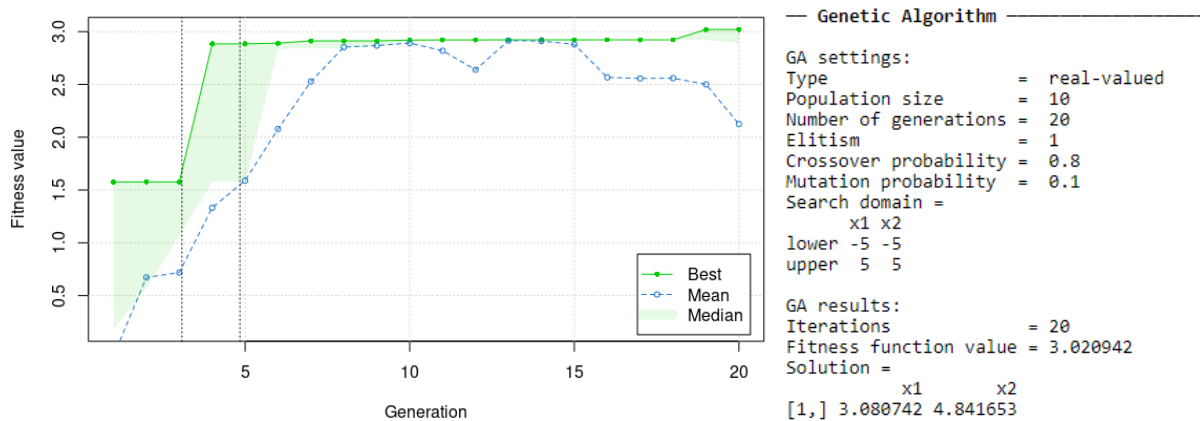


```
GA settings:
Type                = real-valued
Population size     = 10
Number of generations = 6
Elitism             = 1
Crossover probability = 0.8
Mutation probability = 0.1
Search domain =
  x1 x2
lower -5 -5
upper 5 5

GA results:
Iterations          = 6
Fitness function value = 2.264584
Solution =
      x1      x2
[1,] -2.315513 2.439887
```

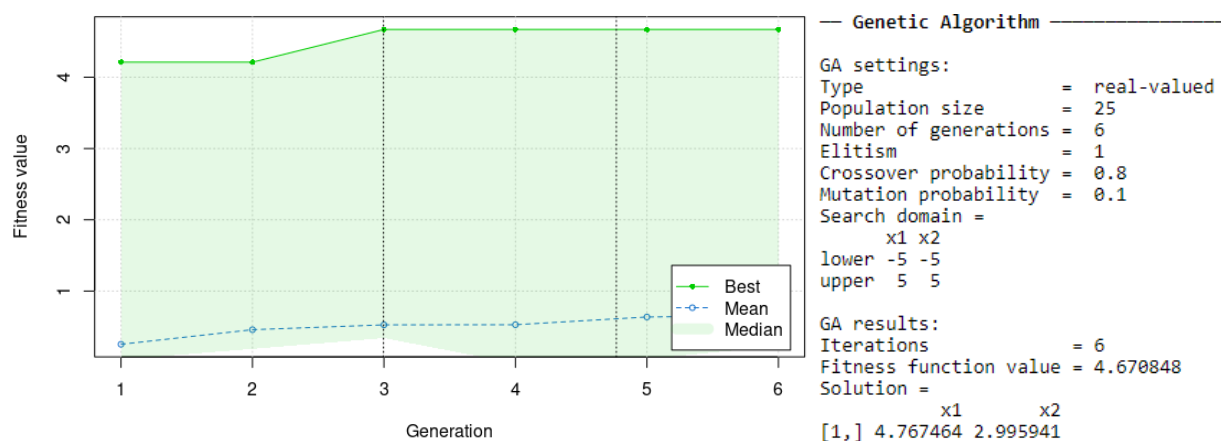
Z otrzymanych wyników można określić, iż przy wielkości populacji równej 10, prawdopodobieństwa mutacji 0.1, prawdopodobieństwa krzyżowania 0.8 oraz liczby iteracji równej 6 wynika maksymalna wartość funkcji wynoszącą 2.26.

Aby sprawdzić wpływ różnych parametrów należy przeprowadzić kilka symulacji, pierwszym parametrem który zostanie zmieniony jest maksymalna ilość iteracji, która została ustawiona na 20.



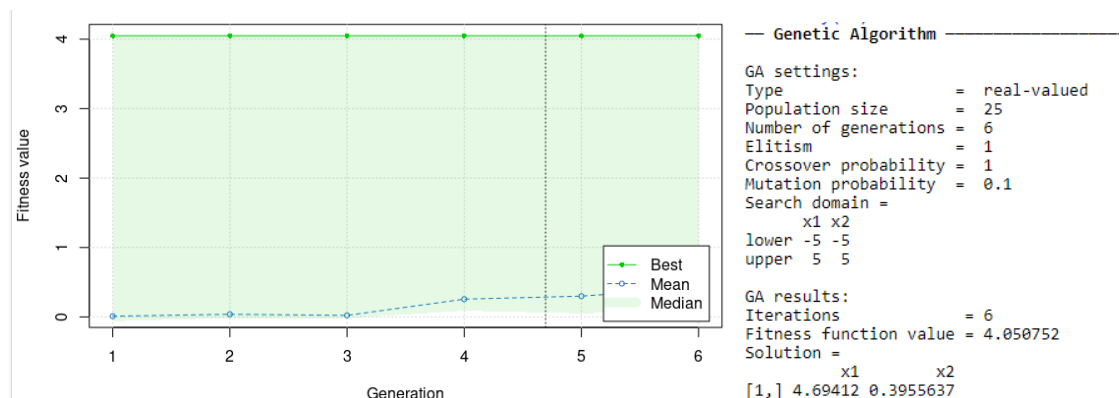
W tym wypadku zmiana parametru wpłynęła pozytywnie na optymalną wartość, która wyniosła 3.02 dla  $x=3.08$  i  $y=4.82$ .

Kolejnym badanym parametrem będzie rozmiar populacji ustawiony do 25.



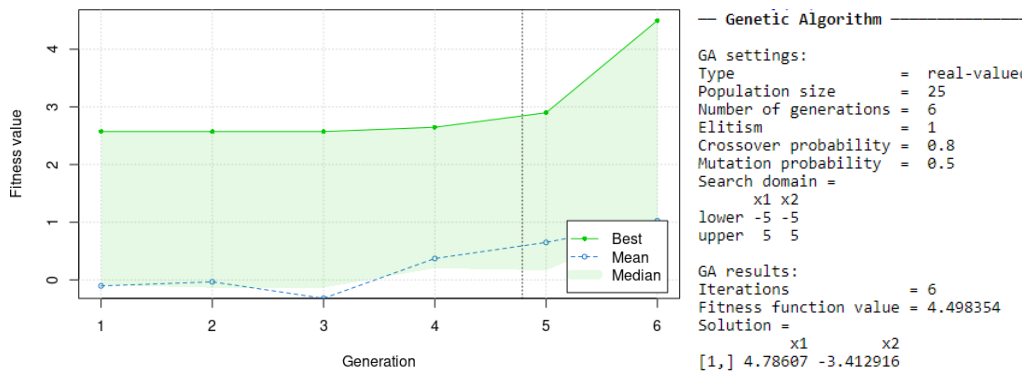
W tym wypadku rozmiar populacji znacząco wpłynęła na wynik końcowy, pokazując 4.67 dla  $x=4.77$  i  $y=3.00$ .

Następnym parametrem jest prawdopodobieństwo krzyżowania ustawione na 1.0.

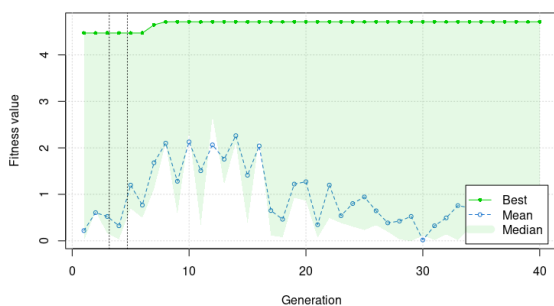


Jak możemy zaobserwować większa szansa na krzyżowanie populacji procentuje w precyzyjność wyniku dając 4.05 dla  $x=4.69$  i  $y=0.40$ .

Kolejnym, a zarazem ostatnim pojedynczo zmienianym parametrem jest szansa na mutację, która została ustawiona na wartość 0.5.



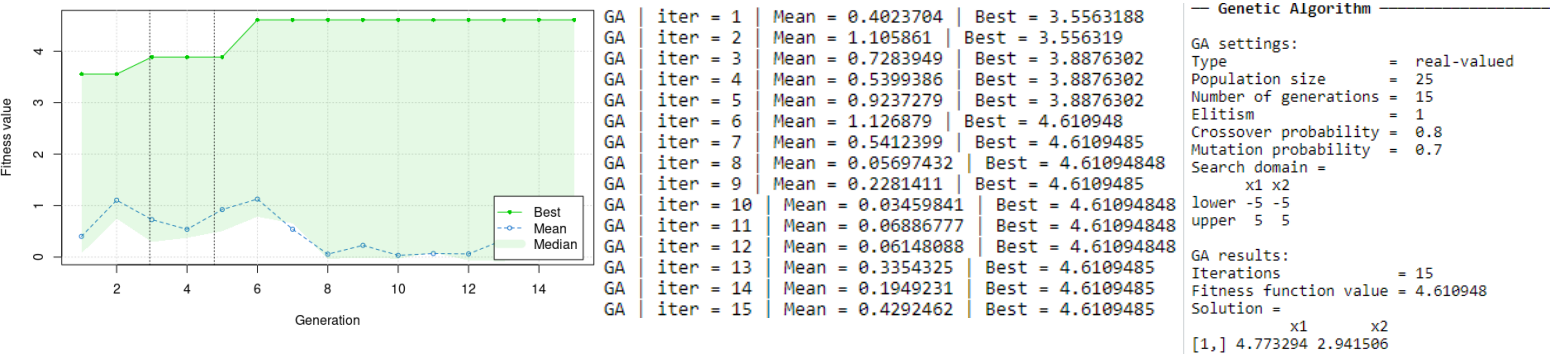
Kolejnym krokiem ćwiczenia jest znalezienie najoptymalniejszych parametrów aby uzyskać najlepszy wynik. W pierwszej kolejności zmodyfikowane zostały rozmiary populacji oraz ilość maksymalnych iteracji.



GA	iter = 1	Mean = 0.2200448	Best = 4.4694214
GA	iter = 2	Mean = 0.606172	Best = 4.469421
GA	iter = 3	Mean = 0.5236649	Best = 4.4694214
GA	iter = 4	Mean = 0.3226614	Best = 4.4694214
GA	iter = 5	Mean = 1.198383	Best = 4.469421
GA	iter = 6	Mean = 0.7682383	Best = 4.4694214
GA	iter = 7	Mean = 1.679697	Best = 4.645904
GA	iter = 8	Mean = 2.100796	Best = 4.712182
GA	iter = 9	Mean = 1.281268	Best = 4.712182
GA	iter = 10	Mean = 2.128380	Best = 4.712182
GA	iter = 11	Mean = 1.506859	Best = 4.712182
GA	iter = 12	Mean = 2.064488	Best = 4.712182
GA	iter = 13	Mean = 1.756261	Best = 4.712182
GA	iter = 14	Mean = 2.262072	Best = 4.712182
GA	iter = 15	Mean = 1.410450	Best = 4.712182
GA	iter = 16	Mean = 2.042000	Best = 4.712182
GA	iter = 17	Mean = 0.6510414	Best = 4.7121818
GA	iter = 18	Mean = 0.4639465	Best = 4.7121818
GA	iter = 19	Mean = 1.220650	Best = 4.712182
GA	iter = 20	Mean = 1.271385	Best = 4.712182
GA	iter = 21	Mean = 0.347398	Best = 4.712182
GA	iter = 22	Mean = 1.199409	Best = 4.712182
GA	iter = 23	Mean = 0.5399638	Best = 4.7121818
GA	iter = 24	Mean = 0.8019045	Best = 4.7121818
GA	iter = 25	Mean = 0.9439975	Best = 4.7121818
GA	iter = 26	Mean = 0.6460733	Best = 4.7121818
GA	iter = 27	Mean = 0.3856692	Best = 4.7121818
GA	iter = 28	Mean = 0.4242414	Best = 4.7121818
GA	iter = 29	Mean = 0.5251347	Best = 4.7121818
GA	iter = 30	Mean = 0.01891467	Best = 4.7121818
GA	iter = 31	Mean = 0.323686	Best = 4.712182
GA	iter = 32	Mean = 0.4932553	Best = 4.7121818
GA	iter = 33	Mean = 0.7576557	Best = 4.7121818
GA	iter = 34	Mean = 0.7112947	Best = 4.7121818
GA	iter = 35	Mean = 0.9012366	Best = 4.7121818
GA	iter = 36	Mean = 0.5292211	Best = 4.7121818
GA	iter = 37	Mean = 0.1833796	Best = 4.7121818
GA	iter = 38	Mean = 0.3533198	Best = 4.7121818
GA	iter = 39	Mean = 0.3585412	Best = 4.7121818
GA	iter = 40	Mean = 0.558618	Best = 4.7121818

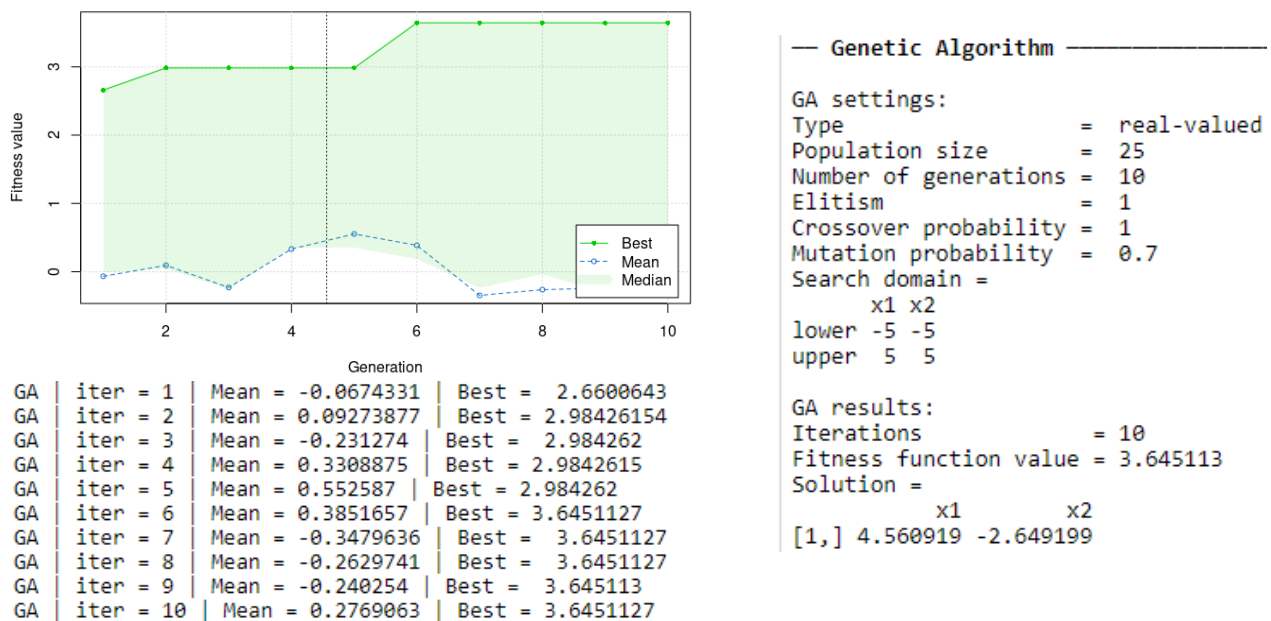
Dzięki zwiększeniu rozmiaru populacji uzyskaliśmy w mniejszej ilości iteracji najoptymalniejsze rozwiązanie.

Z powyższego przykładu możemy wnioskować, iż ilość maksymalna iteracji przy większej populacji nie wpływa tak pozytywnie na algorytm więc kolejnym krokiem jest zmniejszenie liczby iteracji, a podniesieniem prawdopodobieństwa mutacji.



Dzięki zwiększeniu prawdopodobieństwa mutacji jeszcze bardziej przyspieszyliśmy wyszukanie optymalnego rozwiązania, dzięki czemu możemy zmniejszyć ilość iteracji przy czym zoptymalizować algorytm.

Ostatnim parametrem pozostałym do edycji w celu poszukania najoptymalniejszego zestawu zmiennych jest prawdopodobieństwo krzyżowania populacji, w tym wypadku zostanie ono ustawione na 1.0.



W tym wypadku zmiana parametru nie poprawiła wyniku.

### 3. Wnioski

Kluczowym do poprawnego działania algorytmu genetycznego jest prawidłowe dobranie parametrów wejściowych aby program działał szybko i sprawnie. W moim przypadku niska liczba iteracji i wysoki parametr wielkości populacyjnej z utrzymanymi średnimi prawdopodobieństwami mutacji i krzyżowania dawało najlepsze efekty, które ukazały się z najbliższą ideału maksymalną wartością funkcji.