

## TREŚCI ZADAŃ

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

$$a) f(x,y) = (x^2 - y^2) - (1-x)^2, x, y \in \langle -2, 2 \rangle$$

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

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

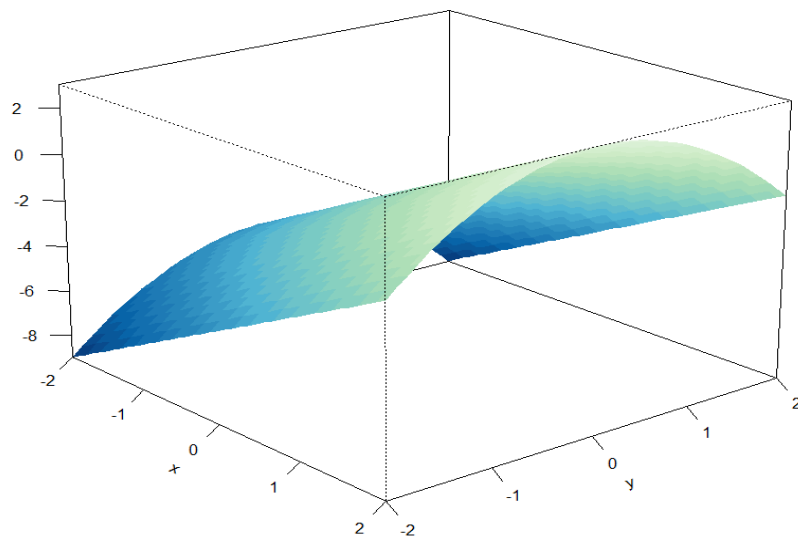
Jeżeli reszta z dzielenia nr Twojego indeksu przez 3 wynosi 1 znajdź maksimum funkcji z punktu a), jeżeli 2 z punktu b), jeżeli 0 z punktu c).

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

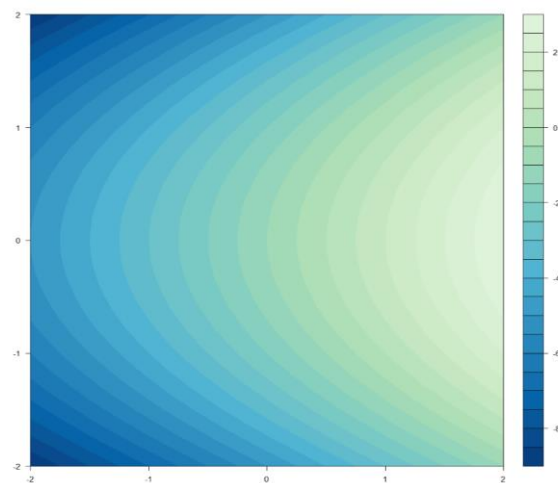
## Realizacja zadania

```
1 library(GA)
2
3 #ustawienie parametrów algorytmu ewolucyjnego
4
5 min_x = -2
6 max_x = 2
7 min_y = -2
8 max_y = 2
9 pop_size = 20
10 pc = 0.8
11 pm = 0.1
12 maxiter = 100
13 seed = 1
14 sleep = -1
15
16 #funkcja optymalizowana
17
18 Funkcjaa <- function(x,y)
19 {
20   (x^2 - y^2) - (1 - x)^2
21 }
22
23 #funkcja dopasowania
24 fitness = function(x) Funkcjaa(x[1], x[2])
25
26 # wykres funkcji
27
28
29
30 x <- y <- seq(-2,2, by = 0.1)
31 f <- outer(x, y, Funkcjaa)
32 persp3D(x, y, f, theta = 50, phi = 20, col.palette = bl2gr.colors)
33 filled.contour(x, y, f, color.palette = jet.colors)
34
35
36 # funkcja monitorująca działanie algorytmu
37
38 monitor <- function(obj){
39 {
40   contour(x, y, f, drawlabels = FALSE, col = grey(0.5))
41   title(paste("iteration = obj@iter", 100), font.main = 1)
42   points(obj@population, pch = 20, col = 2)
43   Sys.sleep(0.2)
44 }
45 }
46
47 # uruchomienie algorytmu
48
49 GA <- ga(
50   type = "real-valued"
51   , fitness = fitness
52   , lower = c(min_x, min_y)
53   , upper = c(max_x, max_y)
54   , popSize = pop_size
55   , pcrossover = pc
56   , pmutation = pm
57   , maxiter = maxiter
58   , keepBest = TRUE
59   , seed = seed
60 )
61
62 #wyświetlanie podsumowanie
63 summary(GA)
64
65 # wykres - najlepiej dopasowany osobnik
66 abline(v = GA@solution, lty = 3)
67
68 #wykres - mediana
69 plot(GA)
70
71 #wyświetlenie wartości najlepszego przystosowania osiągniętego w każdej iteracji
72 GA@bestSol
73
```

Wykres 3D funkcji:  $f(x,y)=(x^2-y^2)-(1-x)^2$



Wykres 2D funkcji:  $f(x,y)=(x^2-y^2)-(1-x)^2$



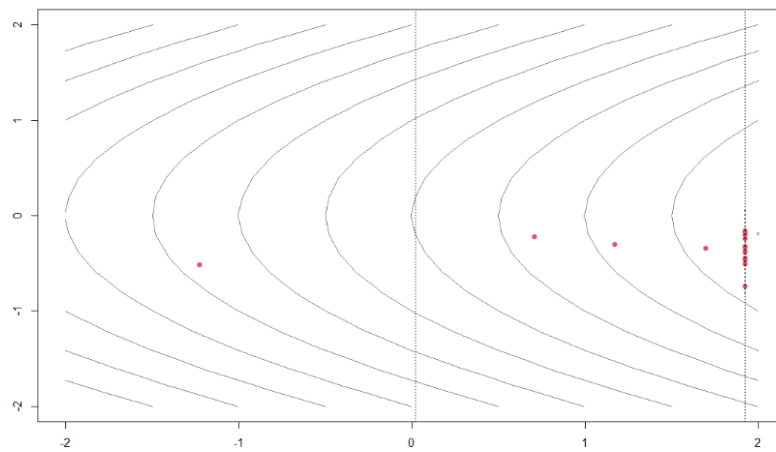
Konfiguracja początkowa – wyniki

```
-- Genetic Algorithm -----

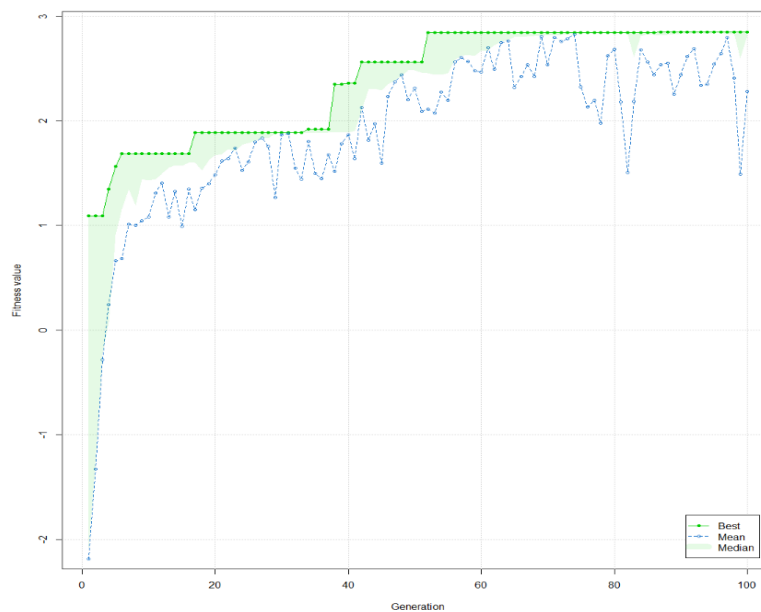
GA settings:
Type           = real-valued
Population size = 20
Number of generations = 100
Elitism        = 1
Crossover probability = 0.8
Mutation probability = 0.1
Search domain =
    x1 x2
lower -2 -2
upper  2  2

GA results:
Iterations      = 100
Fitness function value = 2.84818
Solution =
    x1      x2
[1,] 1.924283 0.01966459
```

Rys. Dane wejściowe i wyniki.



Rys. Wykres przedstawiający populację osobników po setnej iteracji.



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

```

[[87]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[88]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[89]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[90]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[91]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[92]]
  [,1]      [,2]
[1,] 1.925 0.04455477

[[93]]
  [,1]      [,2]
[1,] 1.925 0.04455477

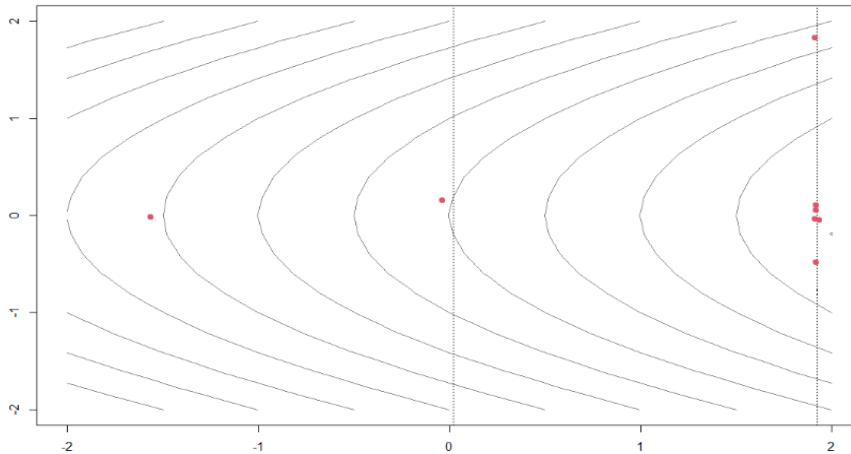
[[94]]
  [,1]      [,2]
[1,] 1.925 0.04455477

```

Rys. Część najlepszego przystosowania w iteracjach 87-94.

Prawdopodobnie w tym zakresie mieści się najlepszy wynik, ponieważ w tych iteracjach się nie zmieniał.

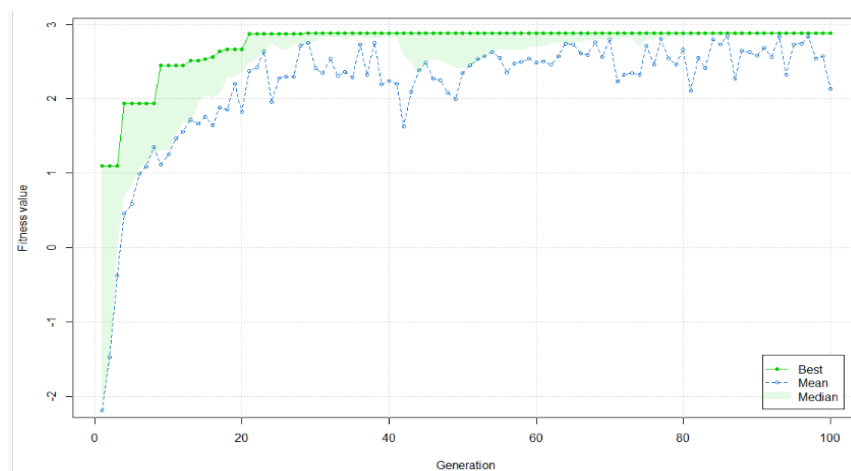
## Prawdopodobieństwo krzyżowania równe 1



Rys. Wykres przedstawiający populację osobników po setnej iteracji.

## Dane wejściowe i wynik algorytmu

```
-- Genetic Algorithm -----  
GA settings:  
Type           = real-valued  
Population size = 20  
Number of generations = 100  
Elitism         = 1  
Crossover probability = 1  
Mutation probability = 0.1  
Search domain =  
    x1 x2  
lower -2 -2  
upper  2  2  
  
GA results:  
Iterations      = 100  
Fitness function value = 2.880125  
Solution =  
    x1      x2  
[1,] 1.940135 0.01201432
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

```

[[87]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[88]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[89]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[90]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[91]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[92]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

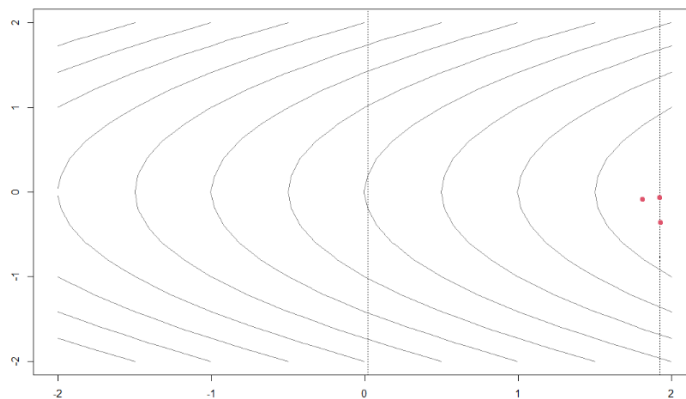
[[93]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

[[94]]
      [,1]      [,2]
[1,] 1.940135 0.01201432

```

Rys. Część najlepszego przystosowania w iteracjach 87-94.

## Prawdopodobieństwo krzyżowania równe 0.2



Rys. Wykres przedstawiający populację osobników po setnej iteracji.

## Dane wejściowe i wynik algorytmu

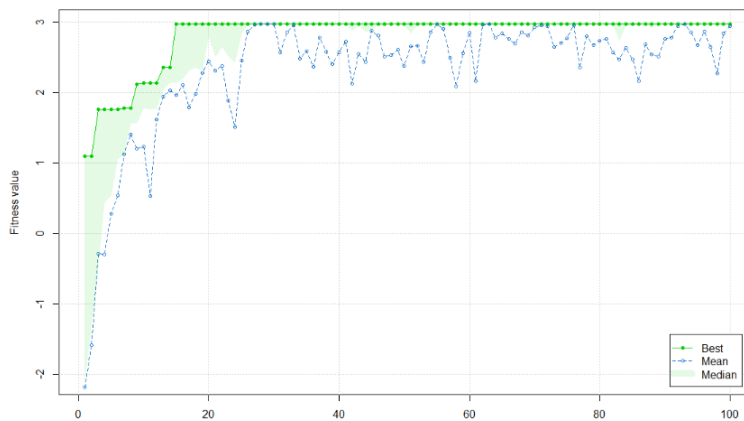
```

-- Genetic Algorithm -----

GA settings:
Type           = real-valued
Population size = 20
Number of generations = 100
Elitism         = 1
Crossover probability = 0.2
Mutation probability = 0.1
Search domain =
  x1 x2
lower -2 -2
upper 2 2

GA results:
Iterations           = 100
Fitness function value = 2.970167
Solution =
  x1      x2
[1,] 1.985087 -0.002768531

```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

## ANALIZA

Analizując otrzymane wyniki można dostrzec, że utrzymują się one na dość wysokim poziomie. W początkowej konfiguracji po setnej iteracji algorytmu, wynik pierwszego argumentu funkcji zbliżył się do 2 ( $x_1 = 1,924283$ ), zaś wynik argumentu drugiego argumentu ( $x_2 = 0,01966459$ ) zbliżył się do 0. Były to rozwiązania optymalne.

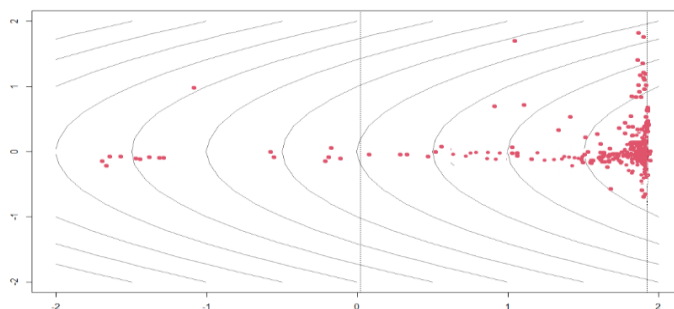
Po zmianie wartości prawdopodobieństwa na 100% wyniki uległy nieznacznemu polepszeniu ( $x_1 = 1.940135$  oraz  $x_2 = 0.01201432$ ), natomiast po zmniejszeniu wartości prawdopodobieństwa do 20% wyniki znacznie się polepszyły i znajdowały się bliżej optymalnych wartości ( $x_1 = 1.985087$ ,  $x_2 = -0.002768531$ ).

Polepszenie wyników można zauważyć porównując ze sobą optymalność procentową wyników:

PC	X1	X2
80%	96%	90%
100%	97%	96%
20%	99,5%	99,9%

## WPŁYW WIELKOŚCI POPULACJI

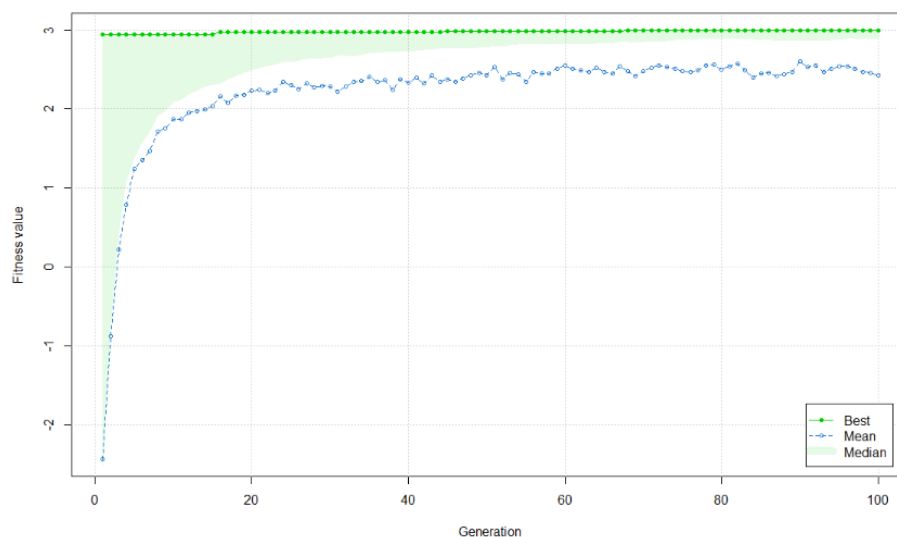
Wielkość populacji 500



Rys. Wykres przedstawiający populację osobników po setnej iteracji.

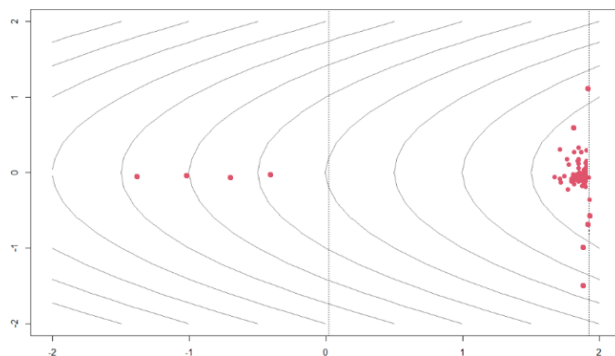
## Dane wejściowe i wynik algorytmu

```
-- Genetic Algorithm -----  
  
GA settings:  
Type           = real-valued  
Population size = 500  
Number of generations = 100  
Elitism         = 25  
Crossover probability = 0.8  
Mutation probability = 0.1  
Search domain =  
    x1 x2  
lower -2 -2  
upper  2  2  
  
GA results:  
Iterations      = 100  
Fitness function value = 2.99281  
Solution =  
    x1      x2  
[1,] 1.996464 0.01090895
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

Wielkość populacji 100

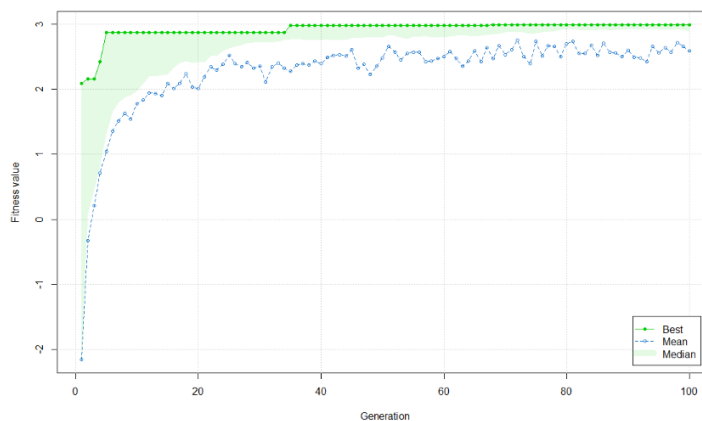


Rys. Wykres przedstawiający populację osobników po setnej iteracji.

## Dane wejściowe i wynik algorytmu

```
GA settings:
Type           = real-valued
Population size = 100
Number of generations = 100
Elitism        = 5
Crossover probability = 0.8
Mutation probability = 0.1
Search domain =
    x1 x2
lower -2 -2
upper  2  2
```

```
GA results:
Iterations      = 100
Fitness function value = 2.990783
Solution =
    x1      x2
[1,] 1.995416 0.00703946
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

---

## ANALIZA

---

Wpływ liczby osobników na wynik można zauważyć porównując ze sobą optymalność procentową wyników:

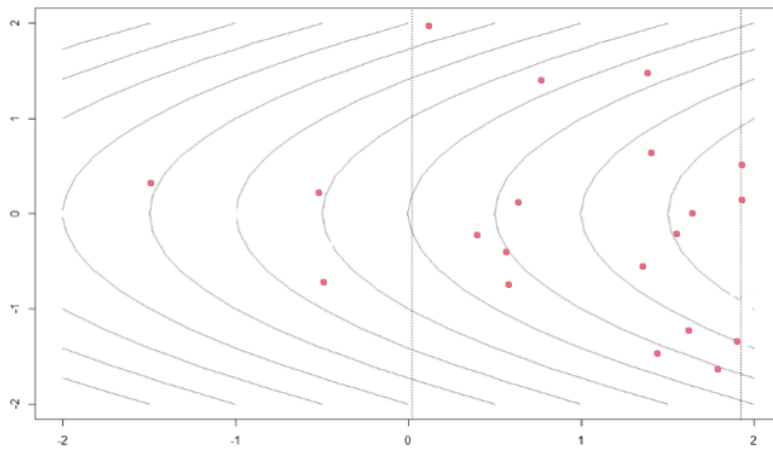
Liczba populacji	X1	X2
20	96%	90%
500	99,8%	99,5%
100	99,75%	99,65%

Na podstawie tabeli wyników można dostrzec że zwiększenie populacji do 500 polepszyło wyniki, natomiast po obniżeniu do 100, wynik drugiego argumentu zmalał, natomiast argument pierwszy wzrósł. Wyniki z liczbą populacji 100 są nieco lepsze, dlatego szukając najbardziej dopasowanej wartości celu powinniśmy szukać w przedziale 100-500 (z liczbami bardziej zmierzającymi do 100).



## Prawdopodobieństwo mutacji

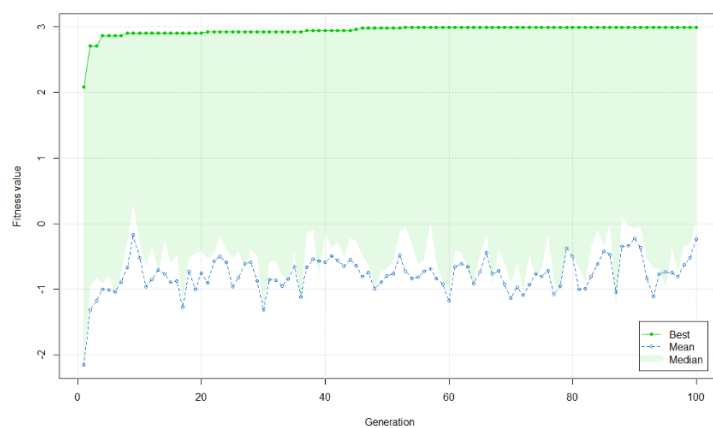
### Prawdopodobieństwo mutacji 1



Rys. Wykres przedstawiający populację osobników po setnej iteracji.

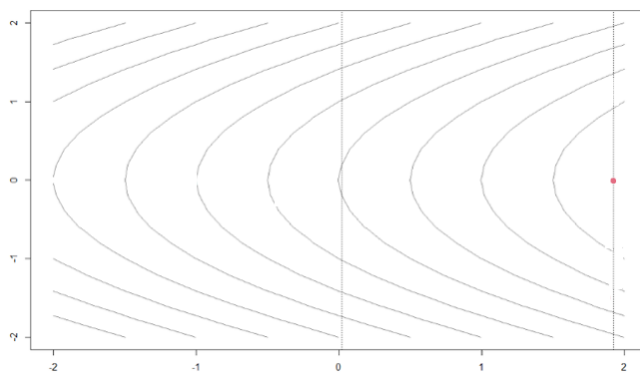
### Dane wejściowe i wynik algorytmu

```
-- Genetic Algorithm -----  
  
GA settings:  
Type           = real-valued  
Population size = 100  
Number of generations = 100  
Elitism        = 5  
Crossover probability = 0.8  
Mutation probability = 1  
Search domain =  
    x1 x2  
lower -2 -2  
upper  2  2  
  
GA results:  
Iterations           = 100  
Fitness function value = 2.991083  
Solution =  
    x1      x2  
[1,] 1.997269 -0.05877858
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

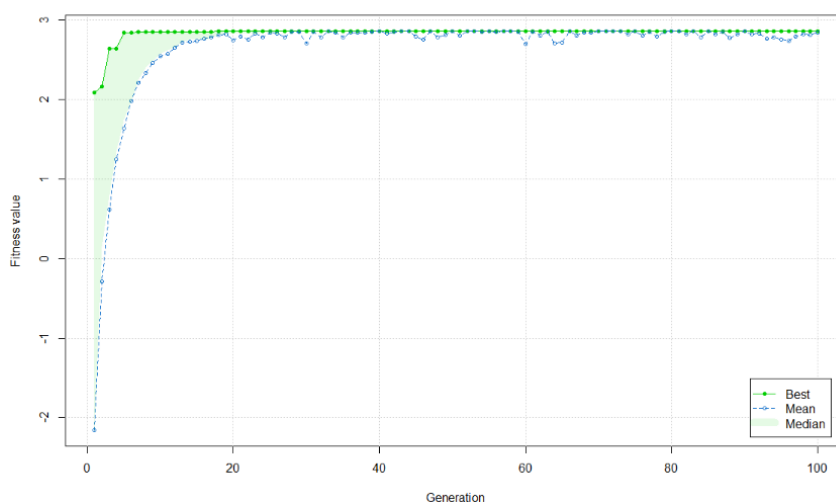
Prawdopodobieństwo mutacji 0.01



Rys. Wykres przedstawiający populację osobników po setnej iteracji.

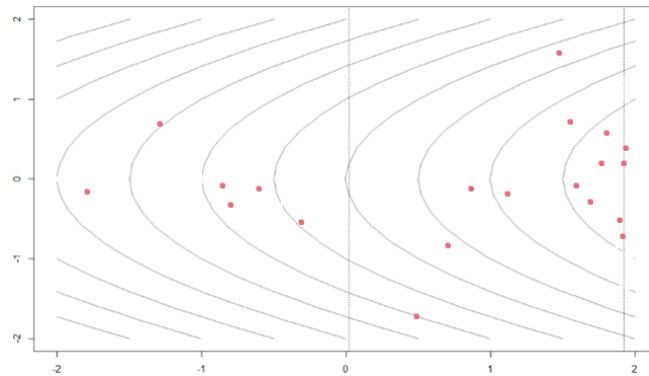
*Dane wejściowe i wynik algorytmu*

```
-- Genetic Algorithm -----  
  
GA settings:  
Type           = real-valued  
Population size = 100  
Number of generations = 100  
Elitism         = 5  
Crossover probability = 0.8  
Mutation probability = 0.01  
Search domain =  
    x1 x2  
lower -2 -2  
upper  2  2  
  
GA results:  
Iterations      = 100  
Fitness function value = 2.856328  
Solution =  
    x1      x2  
[1,] 1.928164 8.068798e-05
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

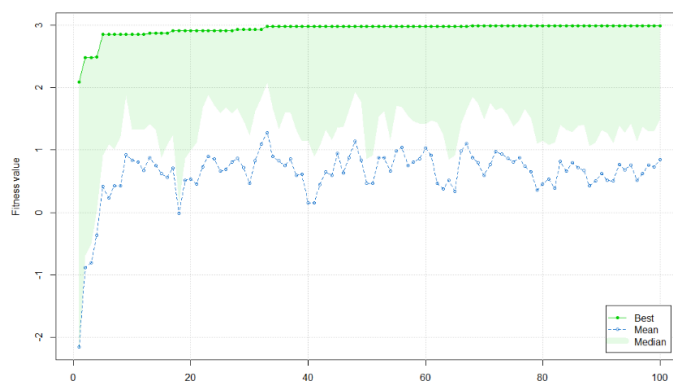
*Prawdopodobieństwo mutacji 0.5*



*Rys. Wykres przedstawiający populację osobników po setnej iteracji.*

*Dane wejściowe i wynik algorytmu*

```
-- Genetic Algorithm -----  
  
GA settings:  
Type                = real-valued  
Population size     = 100  
Number of generations = 100  
Elitism              = 5  
Crossover probability = 0.8  
Mutation probability = 0.5  
Search domain =  
    x1 x2  
lower -2 -2  
upper  2  2  
  
GA results:  
Iterations           = 100  
Fitness function value = 2.990289  
Solution =  
    x1      x2  
[1,] 1.997685 0.07128008
```

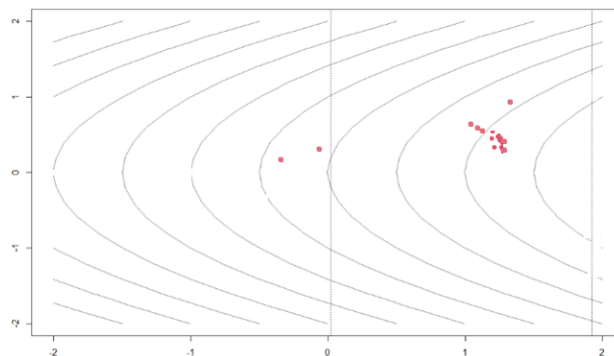


*Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.*

W wyniku zwiększania się prawdopodobieństwa mutacji, wyniki coraz bardziej zbliżają się do oczekiwanych wartości. Wraz ze wzrostem prawdopodobieństwa mutacji zwiększa się różnica między najlepszym wynikiem w danej iteracji, a wynikiem średnim, co doskonale reprezentują powyższe wykresy. Wzrost mutacji wpłynął też na szybkość znajdowania osobnika reprezentującego najlepszy wynik, poza tym nastąpił też spadek osobników zbliżonych do optymalnych wyników.

## Liczba iteracji

Liczba iteracji równa 10



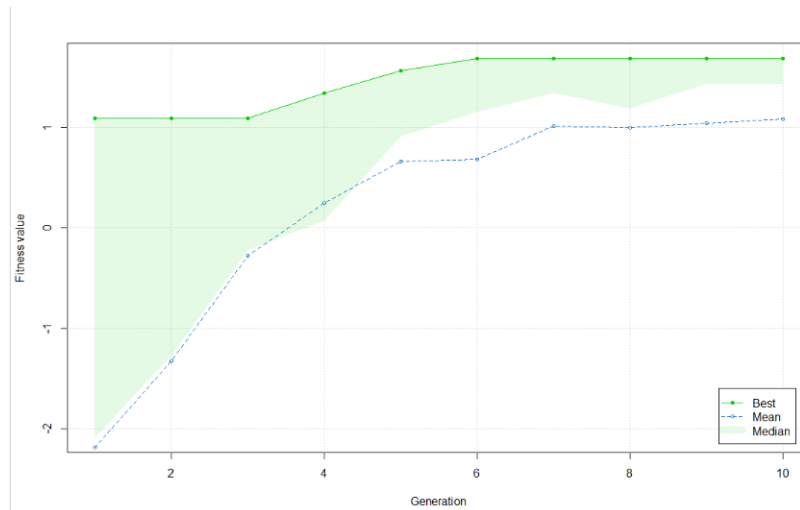
Rys. Wykres przedstawiający populację osobników po 10- iteracji.

## Dane wejściowe i wynik algorytmu

```
GA | iter = 1 | Mean = -2.188672 | Best = 1.094012
GA | iter = 2 | Mean = -1.326515 | Best = 1.094012
GA | iter = 3 | Mean = -0.280478 | Best = 1.094012
GA | iter = 4 | Mean = 0.2426753 | Best = 1.3464272
GA | iter = 5 | Mean = 0.6614948 | Best = 1.5635996
GA | iter = 6 | Mean = 0.683115 | Best = 1.687279
GA | iter = 7 | Mean = 1.014847 | Best = 1.687279
GA | iter = 8 | Mean = 1.002274 | Best = 1.687279
GA | iter = 9 | Mean = 1.043167 | Best = 1.687279
GA | iter = 10 | Mean = 1.083119 | Best = 1.687279
>
> #wyświetlanie podsumowanie
> summary(GA)
-- Genetic Algorithm -----

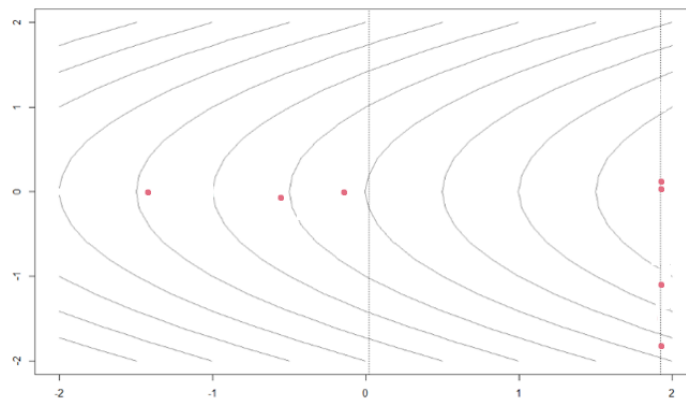
GA settings:
Type           = real-valued
Population size = 20
Number of generations = 10
Elitism        = 1
Crossover probability = 0.8
Mutation probability = 0.1
Search domain =
  x1 x2
lower -2 -2
upper  2  2

GA results:
Iterations           = 10
Fitness function value = 1.687279
Solution =
  x1      x2
[1,] 1.390324 0.3055639
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

Liczba iteracji równa 1000



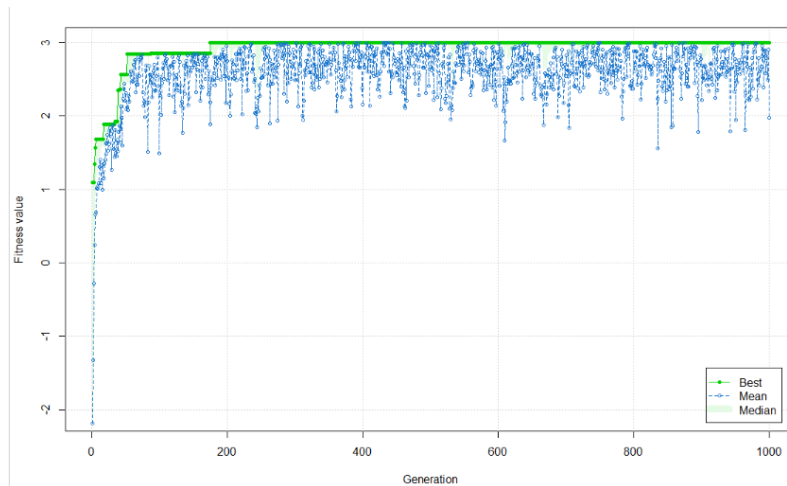
Rys. Wykres przedstawiający populację osobników po 1000- iteracji.

Dane wejściowe i wynik algorytmu

```
-- Genetic Algorithm -----

GA settings:
Type           = real-valued
Population size = 20
Number of generations = 1000
Elitism        = 1
Crossover probability = 0.8
Mutation probability = 0.1
Search domain =
  x1 x2
lower -2 -2
upper 2 2

GA results:
Iterations           = 1000
Fitness function value = 2.995495
Solution =
  x1 x2
[1,] 1.997747 0.0001000284
```



Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

## ANALIZA

Wpływ liczby iteracji na wynik można zauważyć porównując ze sobą optymalność procentową wyników:

Liczba iteracji	X1	X2
20	96%	90%
10	70%	85%
1000	99,86%	100%

Na podstawie wygenerowanych wyników można dostrzec, że wraz ze wzrostem iteracji zwiększa się ilość mutacji i krzyżowań, dzięki czemu wyniki stają się coraz lepsze. Wyniki w 1000 iteracji są najbardziej optymalne (najbardziej zbliżone do oczekiwanych wartości  $x1=2$  oraz  $x2=0$ ).

## Określenie najbardziej optymalnych parametrów na podstawie wcześniej otrzymanych wyników

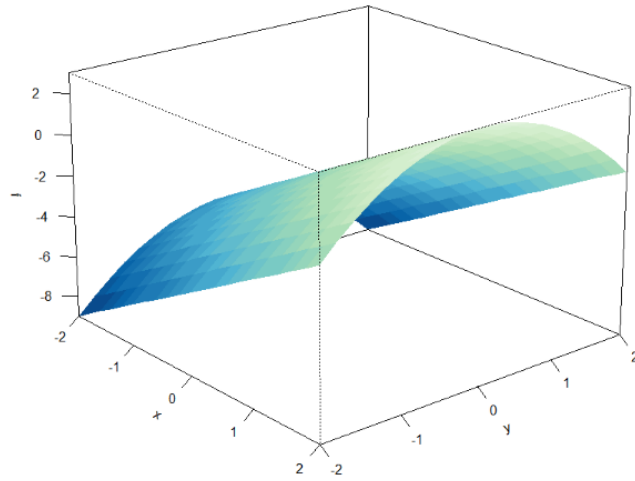
$\min\_x = -2$   $\max\_x = 2$

$\min\_y = -2$   $\max\_y = 2$

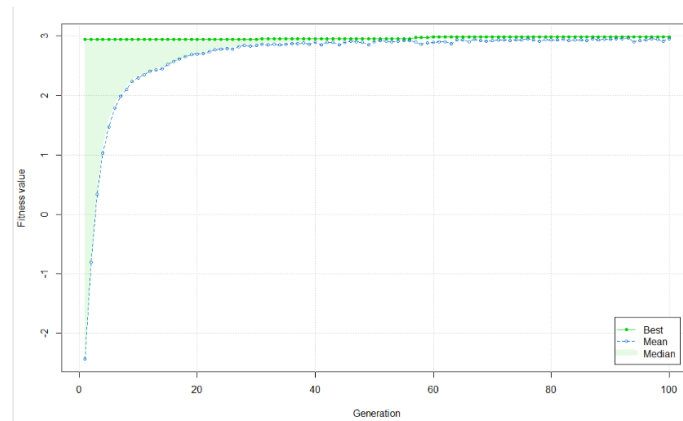
$\text{pop\_size} = 500$   $\text{pc} = 0.7$

$\text{pm} = 0.01$   $\text{maxiter} = 100$

$\text{seed} = 1$   $\text{sleep} = -1$



Rys. Wykres 3D



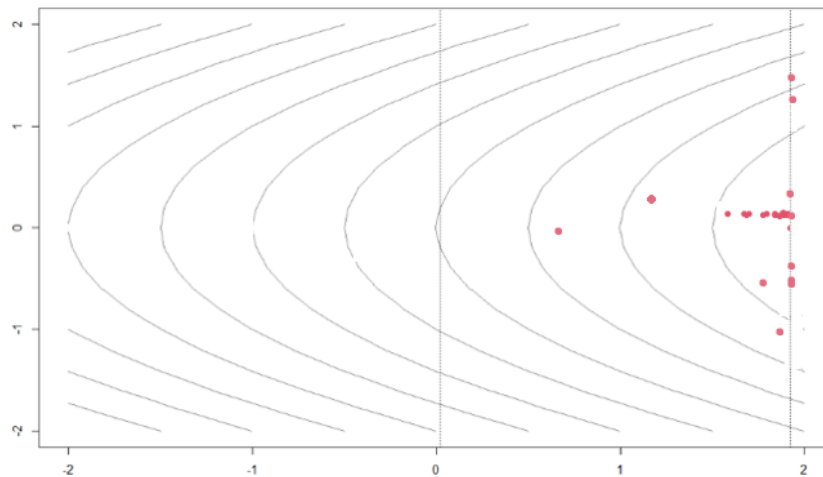
Rys. Mediana, średnia oraz najlepszy wynik w każdej iteracji.

```
-- Genetic Algorithm -----

GA settings:
Type           = real-valued
Population size = 500
Number of generations = 100
Elitism         = 25
Crossover probability = 0.7
Mutation probability = 0.01
Search domain =
    x1 x2
lower -2 -2
upper  2  2

GA results:
Iterations           = 100
Fitness function value = 2.980505
Solution =
    x1      x2
[1,] 1.990253 0.0007981353
```

Rys. Otrzymane wyniki końcowe.



Rys. Wykres przedstawiający populację osobników po iteracji.

---

### OGÓLNA ANALIZA

---

- W każdym z powyższych wariantów algorytm odnalazł w danej dziedzinie wartości bliskie najbardziej optymalnym.
- Wraz ze wzrostem populacji wyniki zwiększają się do maksimum.
- Mutacja i krzyżowanie mają duży wpływ na działanie algorytmu. Im większa liczba mutacji tym wyniki bardziej zbliżają się do oczekiwanych. Natomiast w przypadku krzyżowania im mniejsza ilość prawdopodobieństwa krzyżowania tym lepszy rezultat.
- Liczba iteracji ma wpływ na optymalność czasowa algorytmu.

Najbardziej optymalne rozwiązanie:

```
-- Genetic Algorithm -----

GA settings:
Type           = real-valued
Population size = 10000
Number of generations = 1000
Elitism         = 500
Crossover probability = 0.5
Mutation probability = 0.3
Search domain =
    x1 x2
lower -2 -2
upper  2  2

GA results:
Iterations           = 1000
Fitness function value = 2.999974
Solution =
    x1           x2
[1,] 1.999987 -0.0001109267
```





---

## WNIOSKI

---

- Im większa liczba operacji, tym lepszy wynik. Jest to efekt tego, że dłuższe działanie algorytmu , pozwala uzyskać bardziej dokładny wynik.
- Niestety zwiększenie liczby iteracji wpływa też niekorzystnie na optymalność czasową algorytmu.
- Tworzenie nowych pokoleń związane jest z zastosowaniem operatorów genetycznych tj. mutacja czy krzyżowanie.
- Podczas tworzenia kolejnych pokoleń na rodziców wybierane są osobniki o największej wartości przystosowania. Dzięki temu osobniki w kolejnych iteracjach są coraz lepiej przystosowane i dają lepsze wyniki.
- Każdy osobnik żyjący w danym środowisku musi się do niego przystosować, dzięki temu można powiedzieć, że im wyższy wynik, tym osobnik jest bardziej przystosowany do środowiska, w którym żyje.
- Algorytmy genetyczne znajdują bardzo szerokie zastosowanie w różnych dziedzinach życia codziennego. Przykładem jest np. problem komiwojażera, gdzie należy znaleźć najkrótszą drogę łączącą wszystkie miasta, tak aby przez każde z nich przejść tylko raz.

## PLIKI

-  *Najlepsze\_przystosowanie.R*
-  *GA.R –podstawowy algorytm*