

POLITECHNIKA WROCŁAWSKA

INTELIGENCJA OBLICZENIOWA I JEJ ZASTOSOWANIA

Badanie algorytmu genetycznego z zakresu optymalizacji globalnej dla wybranych funkcji testowych

Autorzy:

Paweł ANDZIUL 200648

Marcin SŁOWIŃSKI 200638

Prowadzący:

dr hab. inż. Olgierd UNOLD,

prof. nadzw. PWr

29 marca 2017

Spis treści

1	Wprowadzenie	2
1.1	Opis zadania projektowego	2
1.2	Środowisko testowe i narzędzia	2
2	Implementacja	3
2.1	Parametryzacja skryptu	4
3	Przebieg badań	5
3.1	Funkcja Ackleys (wariant 2D)	5
3.2	Funkcja Ackleys (wariant 3D)	5
3.3	Funkcja Branin (wariant 3D)	6
3.4	Funkcja Schwefel (wariant 3D)	6
4	Podsumowanie	7

1 Wprowadzenie

Wstęp

1.1 Opis zadania projektowego

1.2 Środowisko testowe i narzędzia

2 Implementacja

Listing 1: Skrypt w języku R wykorzystany do badań

```
1
2 rm(list=ls())
3
4 require("GA")
5 require("globalOptTests")
6 require("rgl")
7
8 graphs <- TRUE
9
10 # Function proxies ----
11 ackley <- function(xx)
12 {
13   return(goTest(par=c(xx, rep(0, 10-length(xx))), fnName="Ackleys", checkDim =
14     TRUE))
15 }
16
17 branin <- function(xx)
18 {
19   return(goTest(par=c(xx), fnName="Branin", checkDim = TRUE))
20 }
21
22 schwef <- function(xx)
23 {
24   return(goTest(par=c(xx,0,0,0,0,0,0,0,0), fnName="Schwefel", checkDim = TRUE))
25 }
26
27 # Params ----
28 isSingleTest = FALSE
29 n <- 7 # default 7
30 GAPopulation <- 50 # default 500
31 GAlterations <- 5 # default 50
32 GAMutations <- 0.1 # % (def 0.1)
33 GACrossovers <- 0.8 # % (def 0.8)
34
35 # Ackleys 2D function ----
36
37 # xFrom <- -32.768
38 # xTo <- 32.768
39 # f <- Vectorize(ackley)
40 # is3D = FALSE
41
42 # Ackleys 3D function ----
43
44 d <- 0.8
45 B <- matrix(c(-32.768, -32.768, 32.768, 32.768), nrow=2, ncol=2, byrow = TRUE)
46 f <- function(x, y) ackley(c(x,y))
47 is3D = TRUE
48
49 # Branin function ----
50
```

```

51 # d <- 0.5
52 # B <- matrix(c(-5, 0, 10, 15), nrow=2, ncol=2, byrow = TRUE)
53 # f <- function(x, y) branin(c(x,y))
54 # is3D = TRUE
55
56 # Schwefel function ----
57
58 # d <- 5
59 # B <- matrix(c(-500, -500, 500, 500), nrow=2, ncol=2, byrow = TRUE)
60 # f <- function(x, y) schwef(c(x,y))
61 # is3D = TRUE
62
63 # Processing ----
64
65 if (is3D) {
66   x <- seq(B[1,1], B[2,1], by = d)
67   y <- seq(B[1,2], B[2,2], by = d)
68 }
69
70 if (graphs) {
71   if (is3D) {
72     z <- outer(x, y, Vectorize(f))
73     nbcol = 100
74     color = rev(rainbow(nbcol, start = 0/6, end = 4/6))
75     zcol = cut(z, nbcol)
76     persp3d(x, y, z, theta=50, phi=25, expand=0.75, col=color[zcol],
77             ticktype="detailed", axes=TRUE)
78     persp3D(x, y, z, theta = -45, phi = 20, color.palette = jet.colors)
79   } else {
80     curve (f, xFrom, xTo)
81   }
82 }

```

2.1 Parametryzacja skryptu

3 Przebieg badań

Akapit

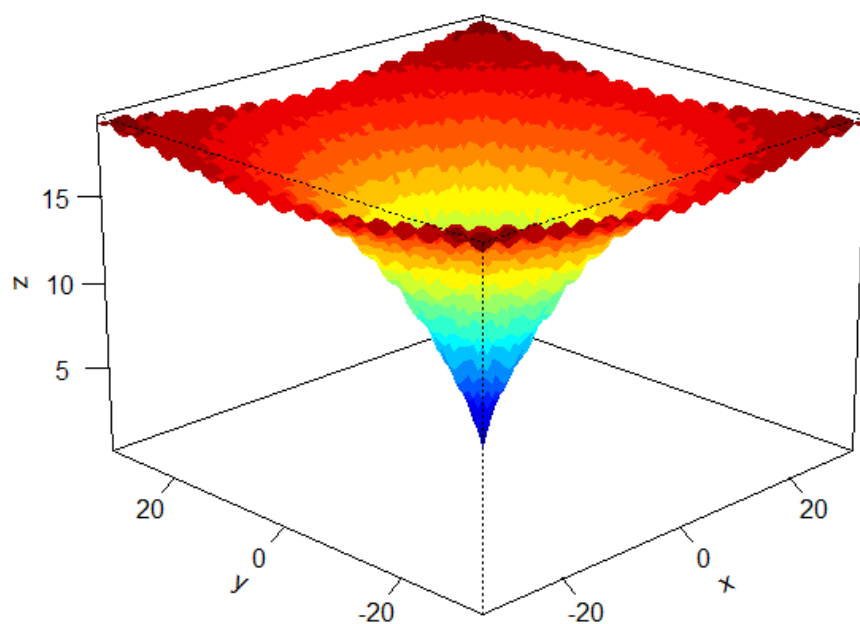
3.1 Funkcja Ackleys (wariant 2D)

Test

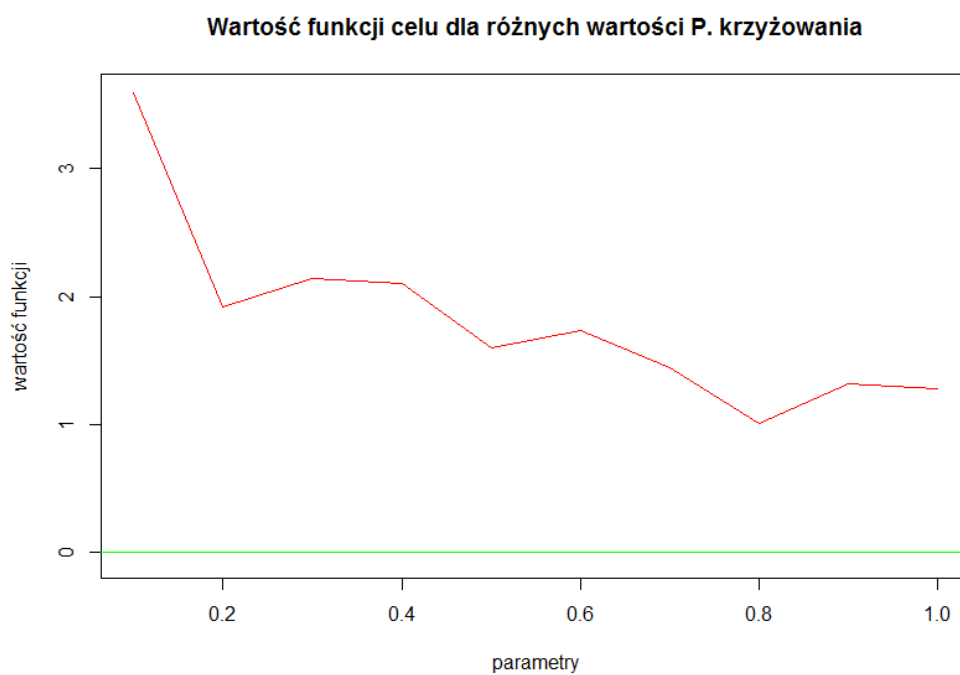
3.2 Funkcja Ackleys (wariant 3D)

Test

Na ilustracji (rys. 1) przedstawiono wykres omawianej funkcji.



Rysunek 1: Wykres funkcji Ackleys (d=3)



Rysunek 2: Wartość znalezionej minimum funkcji w zależności od P. krzyżowania

3.3 Funkcja Branin (wariant 3D)

Test

3.4 Funkcja Schwefel (wariant 3D)

Minimum -837,9658 w punkcie (420,9687; 420,9687).

4 Podsumowanie

Test

Akapit

Literatura

- [1] Artur Suchwałko “Wprowadzenie do R dla programistów innych języków”
<https://cran.r-project.org/doc/contrib/R-dla-programistow-innych-jezykow.pdf>