# Sprawozdanie 1 (algorytmy genetyczne)

**Autorzy:** Remigiusz Nowakowski  
**Grupa:** INMN1(hybryda)\_1.1, INMN1(hybryda)\_sp\_wszyscy   
**Data:** 12.11.2023  
**Przedmiot:** Inteligencja obliczeniowa

|  |  |
| --- | --- |
| **Cel zadania** | Znalezienie minimum funkcji f(x1, x2) = (6\*x1^3-x2)\*exp(-x1^3+x2^3) dla x1 <-4;4> z przedziału i x2 z przedziału <-4,4;4,4> |
| **Źródło danych** | Wzór funkcji oraz zakres przedziałów x1 i x2 |
| **Struktura chromosomu** | x1 <-4;4> i x2 <-4,4;4,4> w funkcji f(x1,x2) |
| **Które chromosomy są dobre?** | x1 = 3.798053 i x2 = 4.176772 |
| **Działanie funkcji Fitness** | Funkcja ta ocenia jakość każdego chromosomu w populacji, co umożliwia algorytmowi ocenę, porównanie i ewolucję potencjalnych rozwiązań. Funkcja fitness jest zdefiniowana jako -fp(x[1], x[2]), co oznacza, że algorytm dąży do maksymalizacji tej wartości (minimalizacji wartości fp(x[1], x[2])). |
| **Wnioski i podsumowanie** | Kod implementuje algorytm genetyczny, który ma na celu znalezienie optymalnych wartości parametrów x1 i x2 dla zadanego problemu optymalizacyjnego. Funkcja celu fp(x1, x2) jest minimalizowana, a algorytm genetyczny ewoluuje populację chromosomów, starając się znaleźć najlepsze rozwiązanie. Rozwiązanie: x1 = -3.798053 i x2 = 4.176772 |
| **Bibliografia** | [1] Biblioteka GA  [2] Moodle  [3] Źródło własne (wzór funkcji oraz przedziały)  [4] |

## Kod źródłowy rozwiązania wraz z dodatkowymi (szczegółowymi) komentarzami:

|  |
| --- |
| # Wczytanie biblioteki GA  library(GA)  # Funkcja celu - przestrzeń poszukiwań  fp <- function(x1, x2) {  return ((6\*x1^3 - x2) \* exp(-x1^3 + x2^3))  }  # Tworzenie siatki punktów w przestrzeni poszukiwań  x1 <- seq(-4, 4, by=0.05)  x2 <- seq(-4.4, 4.4, by=0.05)  z <- outer(x1, x2, fp)  # Wizualizacja funkcji celu w przestrzeni 3D  persp(x1, x2, z, theta=30, phi=30, expand=0.5, col=7)  # Wizualizacja funkcji celu jako obraz  image(x1, x2, z, col=terrain.colors(50))  # Dodanie kontur do wizualizacji  contour(x1, x2, z, add=T, col="grey50", nlevels=50)  # Funkcja monitorująca postęp algorytmu genetycznego  fm <- function(obj) {  # Dodanie konturu do wizualizacji  contour(x1, x2, z, col="darkgrey", nlevels=50,  main=paste("iteration =", obj@iter))  # Dodanie punktów reprezentujących populację  points(obj@population, pch=20)  }  # Uruchomienie algorytmu genetycznego  wyniki <- ga(type="real-valued", lower=c(-4, -4.4), upper=c(4, 4.4),  fitness=function(x) -fp(x[1], x[2]), monitor=fm,  popSize=20, pcrossover=0.85, pmutation=0.05,  elitism=5, maxiter=60, seed=10)  # Podsumowanie wyników algorytmu genetycznego  summary(wyniki)  # Wygenerowanie wykresów przedstawiających proces ewolucji  plot(wyniki) |