# Sprawozdanie 2 (algorytmy genetyczne/ewolucyjne)

**Autorzy:** Maksymilian Romańczuk  
**Grupa:** INIS5\_PR2  
**Data:** 29.11.19  
**Przedmiot:** Sztuczna inteligencja

|  |  |
| --- | --- |
| **Cel zadania** | Rozwiązanie problemu plecakowego przy użyciu algorytmu genetycznego. Zakupy świąteczne, sprawdzenie, które przedmioty opłaca się kupić przy wadze plecaka 100. |
| **Źródło danych** | plecakDb = data.frame(  przedmiot = c("choinka", "bombki", "bombki premium", "karp", "lampki",  "stroj mikolaja", "sztuczny snieg",  "grzane wino", "prezenty", "oplatek", "barszcz" ),  wartosc = c(100, 25, 40, 65, 30, 25, 15, 10, 70, 10, 10 ),  waga = c(40, 25, 10, 20, 30, 15, 25, 30, 20, 10, 15 )  ) |
| **Struktura chromosomu** | Liczby Binarne |
| **Które chromosomy są dobre?** |  |
| **Działanie funkcji Fitness** | Funkcja pozwalająca dla danego osobnika określić jego, jakość (z punktu widzenia rozwiązywanego problemu). |
| **Wnioski i podsumowanie** | Wyniki: "Rozwiązanie: "  przedmiot wartosc waga  1 choinka 100 40  3 bombki premium 40 20  4 karp 65 20  9 prezenty 70 20  "Waga plecaka = 100"  "Wartość przedmiotów = 275"  Podsumowanie:  Plecak został całkowicie zapełniony przedmiotami i wybrał optymalne przedmioty o największej wartości(wybrał bombki premiumu ponad zwyczajne bombki) |
| **Bibliografia** | [1] plecak\_ga.r  [2] <https://pl.wikipedia.org/wiki/Problem_plecakowy>  [3] <https://moodle2.e-wsb.pl/course/view.php?id=42720>  [4] <https://pl.wikipedia.org/wiki/Problem_plecakowy> |

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

|  |
| --- |
| #Złodziej posiada plecak o udźwigu do 100 kg.  #Jakie przedmioty powinien kupić (do 100 kg), aby w sumie miały one mają większą wartość?  #instalacja pakietu GA  install.packages("GA")  library(GA)  #Definiujemy zbiór danych i limit plecaka  plecakDb = data.frame(  przedmiot = c(choinka, bombki, bombki premium, karp, lampki, stroj mikolaja, sztuczny snieg,  grzene wino, prezenty, oplatek, barszcz ),  wartosc = c(100, 25, 40, 65, 30, 25, 15, 10, 70, 10, 10 ),  waga = c(30, 2, 4, 10, 6, 8, 5, 15, 20, 1, 4 )  )  plecakLimit = 70  #plecakDb  #Definiujemy funkcję przystosowania  fitnessFunc = function(chr) {  calkowitaWartoscChr = chr %\*% plecakDb$wartosc  calkowitaWagaChr = chr %\*% plecakDb$waga  if (calkowitaWagaChr > plecakLimit) return(-calkowitaWartoscChr)  else return(calkowitaWartoscChr)  }  #Uruchamiamy algorytm genetyczny dla zadanych parametrów  wyniki=ga(type="binary",nBits=11,fitness=fitnessFunc,popSize=100,  pcrossover=0.85,pmutation=0.05,elitism=5,maxiter=30,seed=10)  #Podsumawanie działania algorytmu genetycznego  summary(wyniki)  plot(wyniki)  #Dekodowanie (prezentacja) pojedynczego rozwiązania  decode=function(chr){  print("Rozwiązanie: ")  print( plecakDb[chr == 1, ] )  print( paste("Waga plecaka =",chr %\*% plecakDb$waga) )  print( paste("Wartość przedmiotów =",chr %\*% plecakDb$wartosc) )  }  decode(wyniki@solution[1,]) |

Wykres problemu plecakowego:

