# 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** | Kupienie możliwie trzech najlepszych napastników Ligi Mistrzów sezonu 2022/2023 mając określony budżet 300 mln euro. Bierzmy pod uwagę 10 najwyżej ocenionych napastników. |
| **Źródło danych** | Transfermarkt: wartość zawodnika Sofascore: średnia ocena za występy w LM, w sezonie 2022/2023 |
| **Struktura chromosomu** | Jest reprezentowana jako wektor binarny o długości 10 (parametr nBits=10), gdzie każdy bit odpowiada jednemu zawodnikowi z listy napastnicyDb. Wartość 1 na i-tej pozycji oznacza, że i-ty zawodnik został wybrany do kupienia, a wartość 0 oznacza, że napastnik ten nie został wybrany. |
| **Które chromosomy są dobre?** | 1 (Rodrygo), 2 (Haaland), 7 (Benzema) |
| **Działanie funkcji Fitness** | Funkcja fitness ma na celu ocenę danego chromosomu (chr) na podstawie określonych kryteriów. W tym konkretnym przypadku, ocena chromosomu jest równa sumie ocen zawodników (calkowitaOcenaChr), które zostały wybrane do kupienia zgodnie z wartościami binarnymi w chromosomie. |
| **Wnioski i podsumowanie** | Algorytm genetyczny jest używany do efektywnego rozwiązania problemu, a kod został dostosowany do uwzględnienia specyfiki tego konkretnego przypadku związanego z wyborem napastników. Wizualizacje i dekodowanie pomagają zrozumieć znalezione rozwiązania. |
| **Bibliografia** | [1] Biblioteka GA  [2] Moodle  [3] Sofascore  [4] Transfermarkt |

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

|  |
| --- |
| # Instalujemy/włączamy wymagane pakiety  #install.packages("GA")  library(GA)  # Definiujemy zbiór danych i limit wartości napastników  napastnicyDb = data.frame(  nazwisko = c("Rodrygo", "Haaland","Vinicius", "Kane", "Martinez", "Grealish",  "Benzema", "Dzeko", "Leao", "Ramos"),  ocena = c(7.26, 7.41, 7.11, 6.99, 7.0, 7.25, 7.16, 7.05, 7.03, 7.01),  wartosc = c(100, 180, 150, 110, 100, 75, 15, 4, 90, 50)  )  napastnicyLimit = 300  # Dodajemy ograniczenie dla maksymalnej liczby napastników  maxNapastnicy = 3  # Definiujemy funkcję przystosowania  fitnessFunc = function(chr) {  calkowitaOcenaChr = chr %\*% napastnicyDb$ocena  calkowitaWartoscChr = chr %\*% napastnicyDb$wartosc  # Nowe warunki sprawdzające ograniczenia  if (calkowitaWartoscChr > napastnicyLimit || sum(chr) > maxNapastnicy)  return(-calkowitaOcenaChr)  else return(calkowitaOcenaChr)  }  # Uruchamiamy algorytm genetyczny dla zadanych parametrów  wyniki = ga(type="binary", nBits=10, fitness=fitnessFunc, popSize=100,  pcrossover=0.85, pmutation=0.05, elitism=5, maxiter=30, seed=10)  # Podsumowanie działania algorytmu genetycznego  summary(wyniki)  plot(wyniki)  # Dekodowanie (prezentacja) pojedynczego rozwiązania  decode = function(chr) {  print("Rozwiązanie: ")  print(napastnicyDb[chr == 1, ])  print(paste("Wartość zawodników transfermarkt =", chr %\*% napastnicyDb$wartosc))  print(paste("Suma ocen zawodników sofascore =", chr %\*% napastnicyDb$ocena))  print(paste("Liczba zawodników =", sum(chr)))  }  decode(wyniki@solution[1,]) |