
Inteligencja obliczeniowa

Laboratorium 2: Algorytmy genetyczne

Zadanie 1

W problemie plecakowym pytamy, jakie przedmioty wziąć do plecaka o ograniczonej objętości, by ich wartość była najwyższa. Załóżmy, że złodziej wkradł się do czyjegoś domu. Ma plecak / udźwig w wysokości 25 kg. Złodziej wypatrzył przedmioty przedstawione poniżej w tabeli. Jakie przedmioty powinien wziąć (do max. 25 kg), aby miały w sumie jak największą wartość?

	przedmiot	wartosc	waga
1	zegar	100	7
2	obraz-pejzaż	300	7
3	obraz-portret	200	6
4	radio	40	2
5	laptop	500	5
6	lampka nocna	70	6
7	srebrne sztucce	100	1
8	porcelana	250	3
9	figura z brązu	300	10
10	skórzana torebka	280	3
11	odkurzacz	300	15

Problem ten jest NP-trudny, więc rozwiązanie go algorytmem typu „brute force” nie jest efektywne. Można jednak z powodzeniem rozwiązać go algorytmem genetycznym. Korzystając z pakietu R i paczki `genalg` rozwiąż powyższy problem.

a) Instalacja (jednorazowo) i załadowanie biblioteki

```
install.packages("genalg")
library(genalg)
```

b) Dodawanie zbioru danych i limitu plecaka.

```
plecakdb <- data.frame(
  przedmiot = c("zegar", "obraz-pejzaż",
    "obraz-portret", "radio", "laptop", "lampka nocna", "srebrne sztucce",
    "porcelana", "figura z brązu", "skórzana torebka", "odkurzacz"),
  wartosc = c(100, 300, 200, 40, 500, 70, 100, 250, 300, 280, 300),
  waga = c(7, 7, 6, 2, 5, 6, 1, 3, 10, 3, 15))
```

```
plecaklimit <- 25
```

c) Chromosomy (rozwiązania) to ciągi 11 bitów. Dla każdego z 11 przedmiotów wybieramy: 0 (nie bierzemy tego przedmiotu) lub 1 (bierzemy ten przedmiot do plecaka). Jakim przedmiotom odpowiada ciąg 00011001001? Ile są warte wszystkie przedmioty? Sprawdź za pomocą poniższych komend.

```
chromosome = c(0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1)
plecakdb[chromosome == 1, ]
```

```
cat(chromosome %*% plecakdb$wartosc)
```

d) Najważniejszy etap to zdefiniowanie funkcji fitness. Funkcja ta, dla chromosomu chr na wejściu, oblicza i zwraca, ile wynosi wartość wszystkich przedmiotów przez niego wskazanych. Jeśli jednak waga przedmiotów przekracza limit plecaka to zwracane jest 0.

Uwaga! Funkcja zwraca ujemną wartość przedmiotu (stąd minus przy return) bo paczka `genalg` traktuje najniższe oceny jako najlepsze.

```
fitnessFunc <- function(chr) {  
  calkowita_wartosc_chr <- chr %*% plecakdb$wartosc  
  calkowita_waga_chr <- chr %*% plecakdb$waga  
  if (calkowita_waga_chr > plecaklimit)  
    return(0) else return(-calkowita_wartosc_chr)  
}
```

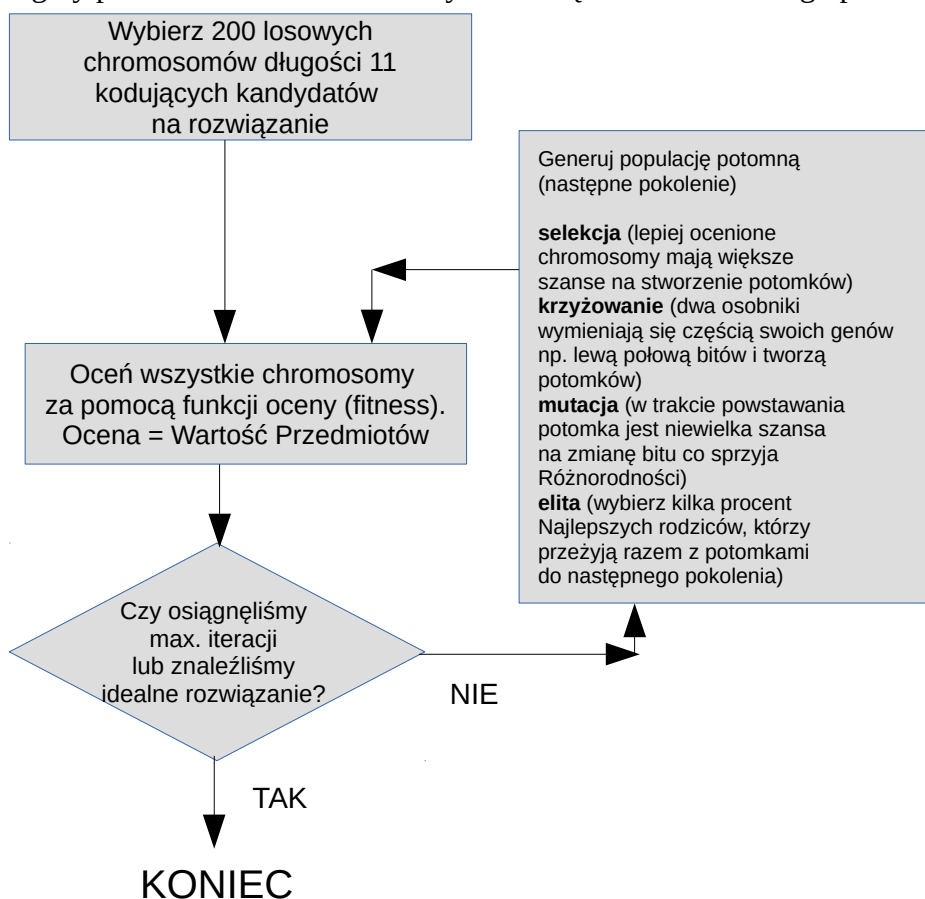
e) Definiujemy algorytm genetyczny wprowadzając odpowiednie parametry i uruchamiamy go. Jakie jest najlepsze rozwiązanie?

```
plecakGenAlg <- rbga.bin(size = 11, popSize = 200, iters = 100,  
  mutationChance = 0.05, elitism = T, evalFunc = fitnessFunc)
```

```
summary(plecakGenAlg, echo=TRUE)
```

f) Skorzystaj z pomysłu z punktu c) aby wyświetlić przedmioty i sumę wartości z najlepszego rozwiązania.

g) Wpisz w konsoli `plecakGenAlg` i zastanów się jakie informacje przechowuje ten obiekt. Przyjrzyj się schematowi działania algorytmu genetycznego poniżej. Po ilu pętłach (iteracjach, pokoleniach) mógłby przerwać działanie z idealnym rozwiązaniem dla naszego problemu?



Zadanie 2

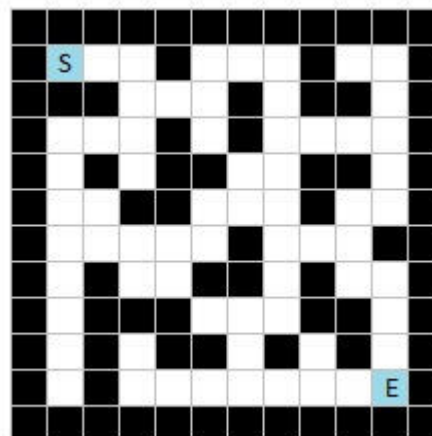
Idea działania algorytmu genetycznego dla problemu z poprzedniego zadania może być podsumowana w poniżej tabelce.

Problem 1: Problem plecakowy	
Struktura chromosomu	Chromosom jest ciągiem binarnym o długości 11. Każdy bit odpowiada wrzuceniu przedmiotu do plecaka (wartość 1) lub pozostawieniu przedmiotu (wartość 0)
Które chromosomy są dobre?	Chromosom jest tym lepszy, im wartościowszy zestaw przedmiotów wskazują jego bity (oczywiście pod warunkiem, że mieszczą się w plecaku).
Działanie funkcji Fitness	Funkcja dostaje chromosom chr. Oblicza wagę wskazywanych przez niego przedmiotów i w przypadku jej przekroczenia zwraca wartość 0 (najgorsza z możliwych ocen). Gdy przedmioty mieszczą się w plecaku, funkcja zwraca wartość zapakowanych przedmiotów (pomnożoną przez -1).
Maks. zakres funkcji Fitness	max: 0 (brak przedmiotów lub przekroczenie wagi) - najslabsza ocena min: wartość wszystkich przedmiotów (jeśli uda się upchnąć je wszystkie w plecaku) pomnożona przez -1 - najlepsza ocena

W tym zadaniu spróbujemy się zastanowić nad innymi problemami. Stwórz analogiczne tabele dla poniższych problemów. Scharakteryzuj ich chromosomy i funkcję oceny.

Problem 2: Labirynt

Mamy labirynt 12x12 pól, przedstawiony na rysunku. Przez labirynt poruszamy się przesuając z pola na pole (ruch: w lewo, prawo, do góry lub na dół). Nie możemy wchodzić na ściany (czarne pola). Naszym celem jest dojście z pola S, do pola E w maksymalnie 40 krokach. Czy istnieje taka droga?



Problem 3: 3-SAT

Formuła logiczna w koniunkcyjnej postaci normalnej 3-CNF składa się z koniunkcji klauzul. Każda klauzula to alternatywa trzech zmiennych lub ich negacji. W problemie spełnialności formuły 3-CNF (ang. „satisfiability problem”, skrót: 3-SAT) pytamy: czy istnieje podstawienie pod zmienne formuły takie, że będzie ona spełniona (czyli da wartość 1)?

Nasza formuła ma 4 zmienne x_1, x_2, x_3, x_4 i następującą postać składającą się z 7 klauzul:

$$\varphi = (\neg x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee x_2 \vee x_3)$$

Zadania domowe

Zadanie 3

Wybierz problem z zadania 2 (labirynt lub 3-SAT) i rozwiąż go z użyciem paczki `genalg` tak, jak zrobiono to w zadaniu 1 dla problemu plecakowego. Głównym wyzwaniem będzie napisanie funkcji fitness, która w obu przypadkach wymaga trochę umiejętności programistycznych.