

# Algorytm genetyczny

Algorytmy genetyczne oraz ich nadzbiór w postaci algorytmów ewolucyjnych to jedne z częściej wykorzystywanych algorytmów inteligencji obliczeniowej. Podczas swego działania naśladują one ewolucję. Główną ideą jest to, że osobniki reprezentują rozwiązanie, im lepszy osobnik, tym lepsze rozwiązanie reprezentuje i tym więcej potomstwa będzie posiadał. Najlepsze osobniki będą miały najwięcej potomstwa, przez co najlepsze osobniki (i rozwiązania) będą wypierać osobniki gorsze (i gorsze rozwiązania). Kolejnym założeniem jest, że potomkowi będą mocno podobni do swoich rodziców, a różnice w stosunku do rodziców będą powstawać w sposób mniej lub bardziej losowy.

Główną zaletą stosowania algorytmów genetycznych jest to, że nie trzeba wiedzieć dokładnie w jaki sposób należy znaleźć rozwiązanie, ale trzeba mieć jedynie funkcję, która potrafi ocenić jak dobre jest każde z rozwiązań (w postaci poj. osobnika) oraz trzeba wiedzieć z jakim systemem należy użyć oraz jakie parametry nim sterują (czyli trzeba mieć wybrany model). Celem stosowania algorytmu genetycznego jest znalezienie takiego rozwiązania (a właściwie takich wartości parametrów), które ma jak najlepszą wartość funkcji przystosowania. W zależności od zastosowania zwykle szuka się jak najwyższej lub jak najniższej wartości tej funkcji.

## Słowniczek:

- Funkcja przystosowania – określa jak dobry jest pojedynczy osobnik. Zwykle zwraca wartość rzeczywistą.
- Osobnik – zawiera informacje rozwiązania, w naszym przypadku zawiera wartości wszystkich parametrów podlegających uczeniu.
- Parametr – określa parametr naszego systemu, w naszym przypadku znamy z góry liczbę parametrów ich liczba jest nie zmienna i wszystkie one są liczbami rzeczywistymi.
- Chromosom – najmniejsza część składowa opisująca osobnika. W naszym przypadku każdy parametr kodowany jest przez stałą liczbę chromosomów, a wartość chromosomu jest bitem.
- Model – system służący do rozwiązania konkretnego zadania (na przykład klasyfikator, aproksymator), którego praca sterowana jest za pomocą parametrów. Sam model może być bezużyteczny, jeśli nieznaną jest poprawna wartość parametrów.
- Liczba chromosomów (tutaj bitów) na parametr – określa jak wiele chromosomów odpowiedzialnych jest za reprezentację wartości parametru w osobniku. Liczba ta może być różna dla każdego parametru, chociaż zwykle nie jest.

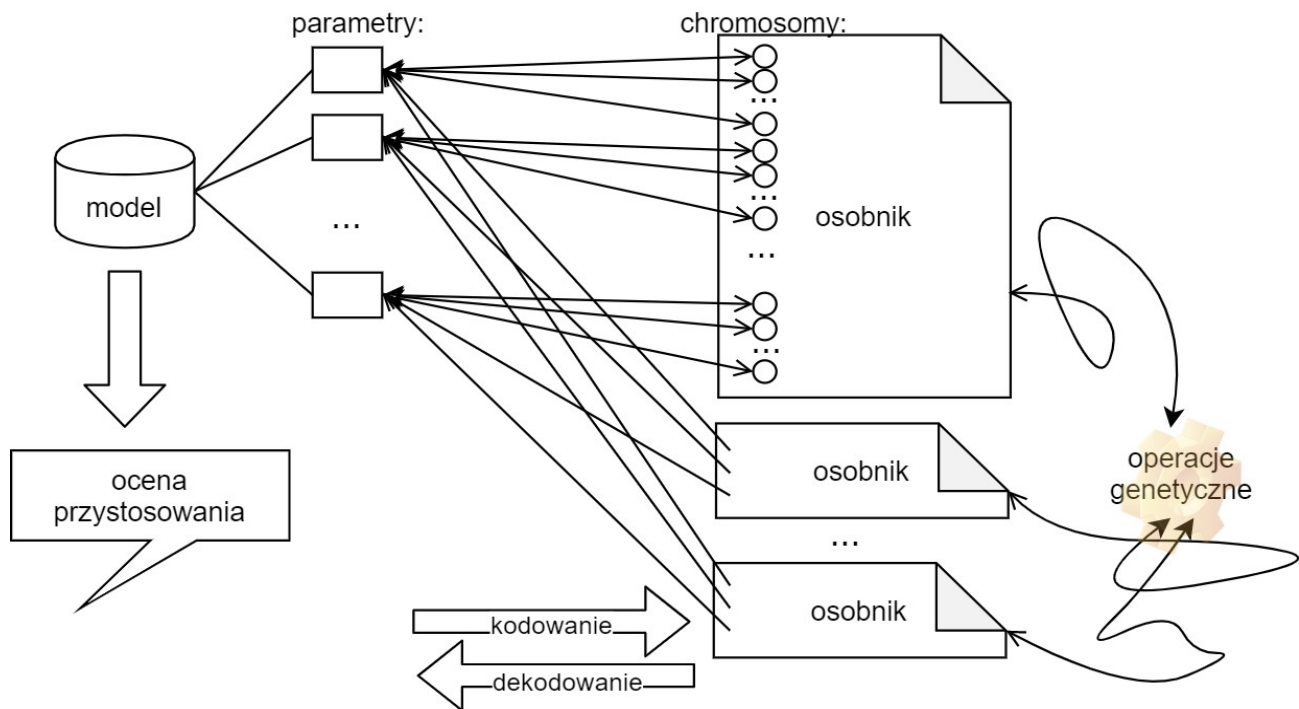
## Algorytm

Bardzo ogólny schemat algorytmu genetycznego to:

1. Stwórz pulę osobników w sposób losowy.
2. Oceń powstałe osobniki za pomocą funkcji przystosowania.
3. Wykonaj alg. genetyczny w pętli programowej przez zadaną liczbę iteracji.
  1. Na bazie obecnej puli osobników stwórz pulę osobników potomnych, tak by osobniki lepsze miały więcej potomstwa, a dzieci nieco różniły się od rodziców. Użyj do tego celu operatorów genetycznych i uzyskanej wcześniej oceny osobników.
  2. Oceń powstałe właśnie osobniki.
  3. Zastąp starą pulę osobników nową pulą składającą się z najlepszych osobników (z puli potomnej i/lub ze starej puli).
  4. *Krok mocno zalecany, ale niepotrzebny.* Wypisz funkcję przystosowania najlepszego osobnika w puli oraz średnią wartość funkcji przystosowania.

W najczęstszej wersji algorytmu genetycznego każdy osobnik opisuje całkowite rozwiązanie problemu, czyli opis całego gotowego systemu. System ten składa się ze stałej liczby parametrów podlegających

uczeniu. Osobnik natomiast składa się z chromosomów, będących najczęściej bitami. Chromosomy zajmują się prezentacją parametrów systemu (parametrów rozwiązania) w taki sposób by możliwa była praca na nich za pomocą algorytmu genetycznego. Z uwagi na to, że chromosom jest zwykle bitem, dlatego do reprezentacji parametrów, które są liczbami wykorzystuje się wiele chromosomów. Proces przekształcania liczb na ciąg bitów (czyli ciągu chromosomów) nazywany jest kodowaniem. Proces odwrotny to dekodowanie. Dzięki dekodowaniu możliwa jest ocena przystosowania danych osobników.



## Kodowanie i dekodowanie

Należy przyjrzeć się jak wygląda dokładnie kodowanie, czyli zamiana wartości parametru na ciąg chromosomów.

Najczęstszym sposobem jest kodowanie równomierne liczb wymiernych. Stałymi sterującymi jest liczba chromosomów na parametr ( $LBnP$ ) oraz dopuszczalny zakres zmienności parametrów modelu (przedział  $[ZDMin; ZDMax]$ ;  $ZD = ZDMax - ZDMin$ ). Stałe te ( $LBnP$ ,  $ZDMin$ ,  $ZDMax$ ) mogą być różne dla każdego z parametrów modelu, lecz tutaj stałe te są dla każdego parametru takie same. Kodowanie to wygląda tak, że ciąg bitów tworzy liczbę nieujemną. Ciągu składającego się z samych zer koduje się minimalną wartość parametru modelu ( $ZDMin$ ), a ciąg bitów składających się z samych jedynek wartość maksymalną parametru modelu ( $ZDMax$ ). Wartości nieskrajne równomiernie kodują pozostałe wartości z przedziału ( $ZDMin$ ;  $ZDMax$ ).

Oto przykład kodowania dla  $ZDMin = -1$ ;  $ZDMax = 2$ ;  $LBnP=3$ ;

kod, wartość chromosomów (kolejność $cb_2; cb_1; cb_0$ )	wartość parametru modelu
000	-1
001	-0,57
010	-0,14
011	0,28
100	0,71
101	1,14
110	1,57
111	2

Warto zauważyć, że proces kodowania w ten sposób umożliwia reprezentację tylko wybranych wartości, tak więc podczas tego procesu dochodzi do kwantyzacji i przez to utraty dokładności.

Proces kodowania można przedstawić jako:

wejścia:  $pm$

stałe (parametry):  $ZDMin, ZDMax, LBnP, ZD = ZDMax - ZDMin$

wyjście:  $cb$

$pm := \max\{pm, ZDMin\}$

$pm := \min\{pm, ZDMax\}$

$ctmp := \text{zaokrąglj}\left(\frac{pm - ZDMin}{ZD} \cdot (2^{LBnP} - 1)\right)$

dla każdego  $b=0..LBnP-1$ :  $cb_b := \text{modulo}\left(\text{zaokrąglj\_w\_dół}\left(\frac{ctmp}{2^b}\right); 2\right)$

, gdzie  $pm$  to wartość parametru modelu,  $cb$  to ciąg bitów (ciąg chromosomów).

Dekodowania można przedstawić jako:

wejścia:  $cb$

stałe (parametry):  $ZDMin, ZDMax, LBnP, ZD = ZDMax - ZDMin$

wyjście:  $pm$

$ctmp := 0$

dla każdego  $b=0..LBnP-1$ :  $ctmp += cb_b \cdot 2^b$

$pm := ZDMin + \frac{ctmp}{2^{LBnP} - 1} \cdot ZD$

## Operator selekcji turniejowej

Operator selekcji służy do wyboru osobników tak, aby osobniki reprezentujące lepsze rozwiązania były wybierane częściej, ale nie wybierać jedynie osobnika najlepszego oraz zadbać o to by im gorszy jest osobnik tym rzadziej był wybrany. Tutaj przedstawiony zostanie operator selekcji turniejowej (w wersji prostszej). Pojedyncze wykorzystanie tego operatora zwraca pojedynczego osobnika. Algorytm ten sterowany jest jedynie za pomocą stałej określającej rozmiar turnieju (TurRozm) i wygląda następująco:

wejścia:  $pula\_osobników, ocena\_osobników$

stałe (parametry):  $TurRozm$

wyjście:  $os\_wybr$

$skład\_turnieju := TurRozm$  różnych osobników wybranych losowo z  $pula\_osobników$

$os\_wybr :=$  osobnik z  $skład\_turnieju$  mający najlepszą  $ocena\_osobników$

gdzie  $pula\_osobników$  to zbiór osobników na podstawie których zostanie wybrany poj. osobnik,  $ocena\_osobników$  to ocena (wartość funkcji przystosowania) każdego osobnika z  $pula\_osobników$ ,  $os\_wybr$  to wybrany osobnik przed operator selekcji turniejowej.

### Przykład:

$pula\_osobników$  składa się z następujących osobników:  $os_1 - 0111, os_2 - 0100, os_3 - 1000, os_4 - 0100$  ,

$ocena\_osobników$  to  $2,2$  (dla  $os_1$ );  $1,8$  (dla  $os_2$ );  $4,2$  (dla  $os_3$ );  $1,8$  (dla  $os_4$ ) ,

w przykładzie tym wybrano, że im jest mniejsza funkcja przystosowania tym lepiej (ta własność brana jest z treści zadania).

$TurRozm = 3$ .

W wyniku działania algorytmu wybrano losowo  $TurRozm$  różnych osobników,

$skład\_turnieju := \{os_3; os_2; os_4\}$  (nie jest istotne, że  $\{os_2; os_4\}$  mają takie same wartości chromosomów).

Okazało się, że wartości ich funkcji przystosowania to  $4,2$  (dla  $os_3$ );  $1,8$  (dla  $os_2$ );  $1,8$  (dla  $os_4$ ) .

Pierwszym najlepszym osobnikiem okazał się  $os_2$  , ponieważ posiadał on najlepszą wartość funkcji przystosowania, dlatego osobnik ten został wybrany jako  $os\_wybr$  posiadający wartość chromosomów

0100.

### Przykład:

*pula\_osobników* składa się z następujących osobników:  $os_1 - 0111, os_2 - 0100, os_3 - 1000, os_4 - 0100$  ,

*ocena\_osobników* to  $2,2$  (dla  $os_1$ );  $1,8$  (dla  $os_2$ );  $4,2$  (dla  $os_3$ );  $1,8$  (dla  $os_4$ ) ,

w przykładzie tym wybrano, że im jest większa funkcja przystosowania tym lepiej,

$TurRozm = 2$ .

Operator selekcji turniejowej użyto 5 razy.

które użycie operatora selekcji	skład_turnieju	os_wybr
1.	$os_1; os_3$	$os_3$
2.	$os_4; os_1$	$os_1$
3.	$os_2; os_3$	$os_3$
4.	$os_1; os_4$	$os_1$
5.	$os_3; os_2$	$os_3$

## Operator selekcji „hot deck” („wybierz najlepszego”)

Operator selekcji elitarniej służy do wybrania osobników w sposób nielosowy tak, aby osobniki jak najlepsze otrzymały jak największą liczbę potomków i proporcjonalnie z zajmowanym miejscem liczba ich potomków malała.

Tutaj zostanie użyty operator „hot deck”, który służy do tego, aby wybrać tylko jednego osobnika - najlepszego z puli. Zwykle osobnik ten nie podlega później jakiegokolwiek zmianie.

## Operator krzyżowania

Operator ten służy do tworzenia nowych osobników na podstawie ich rodziców. Proces ten wygląda tak, iż chromosom potomków okazuje się być składanką chromosomów rodziców. Tutaj przedstawiony będzie proces krzyżowania jednopunktowego, który tworzy 2 potomków. Oto algorytm:

wejscia:  $cb_{r1}, cb_{r2}$

stałe (parametry):  $LBnOs$  = liczba bitów opisująca poj. osobnika

wyjście:  $cb_{p1}, cb_{p2}$

$b\_ciecie$  := całkowita liczba losowa z przedziału  $[0; LBnOs - 2]$  ,

dla każdego  $b = 0..b\_ciecie$ :

$cb_{p1,b} := cb_{r1,b}; cb_{p2,b} := cb_{r2,b}$

dla każdego  $b = b\_ciecie + 1..LBnOs - 1$ :

$cb_{p1,b} := cb_{r2,b}; cb_{p2,b} := cb_{r1,b}$

gdzie  $cb_{r1}, cb_{r2}$  to wartość chromosomów obu rodziców,  $cb_{p1}, cb_{p2}$  to wartości chromosomów obu potomków.

rodzic 1:	[	a1	a2	a3		a4	a5	a6	a7	a8	a9	]
rodzic 2:	[	b1	b2	b3		b4	b5	b6	b7	b8	b9	]
dziecko 1:	[	a1	a2	a3		b4	b5	b6	b7	b8	b9	]
dziecko 2:	[	b1	b2	b3		a4	a5	a6	a7	a8	a9	]

### Przykład:

Na podstawie dwóch osobników o wartości chromosomów  $cb_{r_1}=[101011]$ ,  $cb_{r_2}=[001100]$  należy stworzyć dwóch potomków za pomocą krzyżowania jednopunktowego.

Wylosowano punkt cięcia  $b_{cięcie} = 3$ .

W wyniku krzyżowania otrzymano  $cb_{p_1}=[101000]$ ,  $cb_{p_2}=[001111]$  .

## Operator mutacji

Operator ten służy do losowej zmiany wartości chromosomu. Tutaj będzie użyty operator mutacji jednopunktowej. Oto algorytm:

wejścia:  $cb_{we}$

stałe (parametry):  $LBnOs$  = liczba bitów opisująca poj. osobnika

wyjście:  $cb_{wy}$

$b\_punkt$  = całkowita liczba losowa z przedziału  $[0; LBnOs - 1]$

$cb_{wy} := cb_{we}$

$cb_{wy, b\_punkt} := \neg cb_{wy, b\_punkt}$

### Przykład:

Wartość ciągu chromosomów przed mutacją to  $cb_{we}=[10100101]$  .

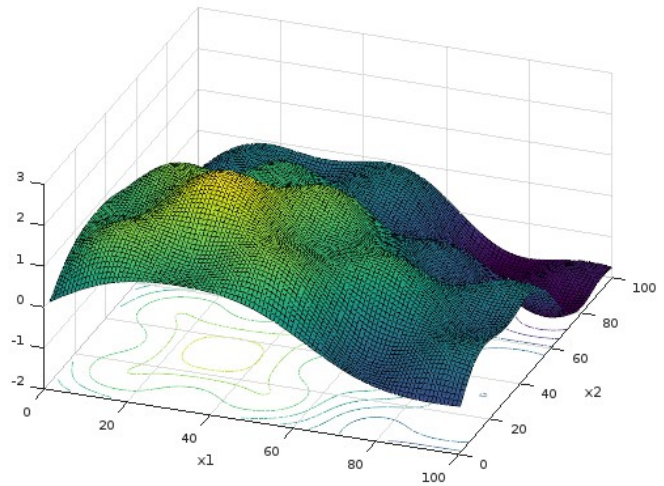
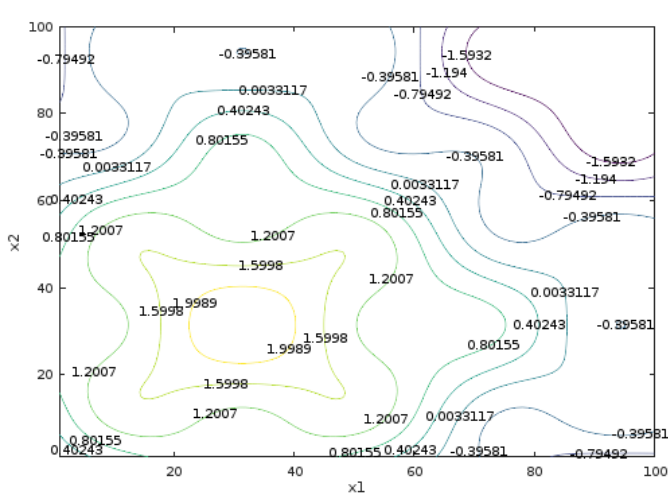
Wylosowano punkt mutacji jako  $b\_punkt = 4$ ;

Wartość ciągu chromosomów po mutacją to  $cb_{wy}=[10101101]$  .

## Zadanie 1. „zmutowany dywanik”

Celem działania algorytmu genetycznego jest znalezienie takiej wartości parametrów  $x_1$ ,  $x_2$  aby otrzymać jak największą wartość funkcji  $F(x_1, x_2) = \sin(x_1 * 0.05) + \sin(x_2 * 0.05) + 0.4 * \sin(x_1 * 0.15) * \sin(x_2 * 0.15)$ .

Dopuszczalny przedział zmienności tych parametrów to  $[0; 100]$ . Oto wykresy tej funkcji:



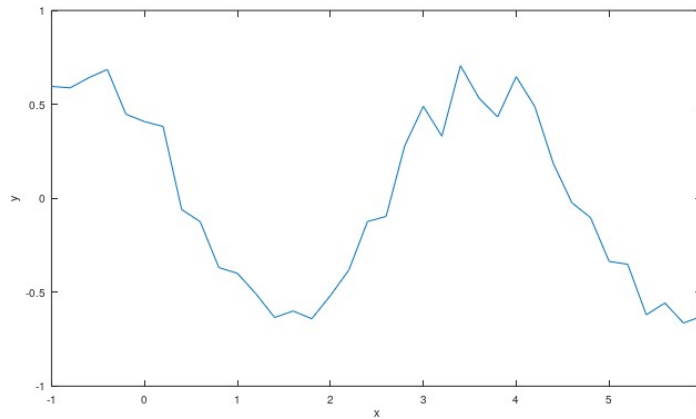
Należy tak dobrać parametry algorytmu genetycznego, aby najlepszy osobnik z puli osiągnął wartość jak najwyższą, najlepiej powyżej 1,8 (jeśli to możliwe). Należy wybrać liczbę chromosomów na parametr co najmniej 3 bity, liczbę iteracji co najmniej 20, liczba osobników ma być nieparzysta i wynosić co najmniej 9, rozmiar turnieju od 2 do 20% liczby osobników. Po każdej iteracji i przed pierwszą należy wypisać funkcję przystosowania najlepszego osobnika w puli oraz średnią wartość funkcji przystosowania. Nie trzeba używać operatora krzyżowania. Do puli potomnej ma zostać również dodany w postaci niezmienionej najlepszy osobnik z puli rodzicielskiej.

W celu wytłumaczenia można przedstawić schemat działania tego algorytmu w następujący sposób:

1. Stwórz pulę osobników w sposób losowy. Każdy osobnik opisany jest przez liczbę bitów (chromosomów) równą  $LBP * \text{liczba parametrów}$  (tutaj 2). Wartość każdego bitu dobierana jest losowo.
2. Dla każdego osobnika osobno dekoduj wartości  $x_1$ ,  $x_2$  i wylicz na ich podstawie funkcję przystosowania.
3. Wykonaj alg. genetyczny w pętli programowej przez zadaną liczbę iteracji.
  1. Użyj operatora selekcji turniejowej, aby stworzyć nową pulę populacji o rozmiarze liczba\_osobników - 1.
  2. Użyj na każdym osobniku w nowej puli operator mutacji jednopunktowej.
  3. Dodaj do nowej puli najlepszego osobnika ze starej puli populacji (operator „hot deck”).
  4. Dla każdego osobnika w nowej populacji osobno dekoduj wartości  $x_1$ ,  $x_2$  i wylicz na ich podstawie funkcję przystosowania.
  5. Wypisz najlepszą oraz średnią wartość funkcji przystosowania osobników w nowej puli.
  6. Zastąp starą pulę osobników nową pulą.

## Zadanie 2. (zadanie niekonieczne – na wyższą ocenę, implementacja tego zadania razem z kolejnym jest niepotrzebna, lepiej zaimplementować następne zadanie)

Celem działania algorytmu genetycznego jest znalezienie takich wartości parametrów  $p_a$ ,  $p_b$ ,  $p_c$ , aby wartość funkcji  $f(x) = p_a * \sin(p_b * x + p_c)$  była jak najbardziej podobna do wartości uzyskanych dla niżej zamieszczonej bazy próbek. Wartość funkcji przystosowania równa jest sumie kwadratów błędów dla wszystkich próbek pomiędzy tym co uzyskano, a tym co być powinno. Oczywiście trzeba tak dobrać wartość parametrów  $p_a$ ,  $p_b$ ,  $p_c$ , aby wartość tej funkcji była możliwie najniższa. Przedział zmienności parametrów  $p_a$ ,  $p_b$ ,  $p_c$  to  $[0; 3]$ .



Przykład działania funkcji przystosowania:

nr próbki	x	y	f(x; pa, pb, pc)
1	-1	0,59	0,4
2	-0,8	0,58	0,6
3	..	..	..
4	..	..	..

Funkcja przystosowania =  
 $(0,59 - 0,4)^2 + (0,58 - 0,6)^2 + \dots$

Należy wybrać liczbę chromosomów na parametr co najmniej 4 bity, liczbę iteracji co najmniej 100, liczba osobników to 13, rozmiar turnieju to 3. Po każdej iteracji należy i przed pierwszą należy wypisać funkcję przystosowania najlepszego osobnika w puli oraz średnią wartość funkcji przystosowania. W puli potomnej mają się znaleźć: 4 osobniki powstałe przez selekcję i krzyżowanie; 4 osobniki powstałe przez selekcję i mutację; 4 osobniki powstałe przez selekcję, krzyżowanie i mutację; najlepszy osobnik ze starej puli.

Baza próbek *sinusik.txt*:

```
-1,00000 0,59014
-0,80000 0,58012
-0,60000 0,57010
-0,40000 0,56008
-0,20000 0,55006
0,00000 0,54004
0,20000 0,53002
0,40000 0,52000
0,60000 0,51000
0,80000 0,50000
1,00000 0,49000
1,20000 0,48000
1,40000 0,47000
1,60000 0,46000
1,80000 0,45000
2,00000 0,44000
2,20000 0,43000
2,40000 0,42000
2,60000 0,41000
2,80000 0,40000
3,00000 0,39000
3,20000 0,38000
3,40000 0,37000
3,60000 0,36000
3,80000 0,35000
4,00000 0,34000
4,20000 0,33000
4,40000 0,32000
4,60000 0,31000
4,80000 0,30000
5,00000 0,29000
5,20000 0,28000
5,40000 0,27000
5,60000 0,26000
5,80000 0,25000
6,00000 0,24000
```

opis *sinusik\_names.txt*:

x r  
y z

$0,6 * \sin(1,5 * x + 2,4) + \text{szum}[-0,15; +0,15]$

Można przedstawić schemat działania algorytmu genetycznego w tym zadaniu w następujący sposób:

1. Stwórz pulę osobników w sposób losowy. Każdy osobnik opisany jest przez liczbę bitów (chromosomów) równą  $L_{BnP} * \text{liczba parametrów}$  (tutaj 3). Wartość każdego bitu dobierana jest losowo.
2. Dla każdego osobnika dekoduj wartości pa, pb, pc i wylicz na ich podstawie funkcję przystosowania.
3. Wykonaj alg. genetyczny w pętli programowej przez zadaną liczbę iteracji.
  1. Użyj operatora selekcji turniejowej, aby stworzyć nową pulę populacji o rozmiarze liczba\_osobników - 1.
  2. Użyj na nowej puli populacji operatora krzyżowania 4 razy: na pierwszych dwóch osobnikach, na osobnikach trzecim i czwartym, na osobnikach dziewiątym i dziesiątym oraz na ostatnich dwóch osobnikach.
  3. Użyj na osobnikach z nowej puli od 5go do ostatniego operatora mutacji jednopunktowej.

4. Dodaj do nowej puli najlepszego osobnika ze starej puli populacji (operator „hot deck”).
5. Dla każdego osobnika w nowej populacji dekoduj wartości  $p_a$ ,  $p_b$ ,  $p_c$  i wylicz na ich podstawie funkcję przystosowania.
6. Wypisz najlepszą oraz średnią wartość funkcji przystosowania osobników w nowej puli.
7. Zastąp starą pulę osobników nową pulą.

### Zadanie 3. XOR (zadanie niekonieczne – na wyższą ocenę, implementacja tego zadania razem z wcześniejszym jest niepotrzebna, lepiej zaimplementować to zadanie)

Celem zadania jest znalezienie optymalnych wartości wag sieci neuronowej służącej do rozwiązania problemu XOR. Sieć ta ma mieć dokładnie taki sam kształt i parametry jak w instrukcji dot. wstecznej propagacji. Jako funkcję przystosowania niech posłuży suma kwadratów błędów otrzymanych na wyjściach sieci dla wszystkich 4 próbek. Celem działania algorytmu genetycznego jest znalezienie takiej wartości wag, aby suma kwadratów błędów była możliwie jak najmniejsza. Przykład działania funkcji przystosowania:

nr próbki	$w_{e_1}$	$w_{e_2}$	d	$wy_{całejSieci}$
1	0	0	0	0,4
2	0	1	1	0,6
3	1	0	1	0,3
4	1	1	0	0,1

**Funkcja przystosowania =**  
 $(0-0,4)^2+(1-0,6)^2+(1-0,3)^2+(0-0,1)^2$

Liczba parametrów modelu to  $3*3$  (3 neurony, 3 wagi na neuron). Przedział zmienności parametrów to  $[-10;+10]$ . Pozostałe parametry algorytmu genetycznego są takie same jak w poprzednim zadaniu.