

Poszukiwanie ekstremum funkcji za pomocą algorytmu ewolucyjnego

Celem ćwiczenia będzie poszukiwanie minimum funkcji zmiennych rzeczywistych $F(\mathbf{X})$ bez ograniczeń i z ograniczeniami. Poszukiwania mają być zrealizowane za pomocą pakietu GAOT, z zastosowaniem rzeczywistego kodowania chromosomu.

Zakładamy, że ograniczenia będą miały postać: $G_i(\mathbf{X}) \geq 0$, dla $i = 1 \dots m$, gdzie m to ilość ograniczających nierówności.

Ograniczenia mogą być uwzględniane w minimalizowanej funkcji jako dodatkowy składnik kary:

$$F'(\mathbf{X}) = F(\mathbf{X}) + r \cdot \sum_{i=1}^m f_i^{\alpha}(\mathbf{X}),$$

gdzie: $f_i(\mathbf{X}) = |\min\{G_i(\mathbf{X}), 0\}|$, α – współczynnik dobierany empirycznie (przykładowo 2), r – współczynnik kary (również dobierany empirycznie).

Zadania do wykonania na zajęciach:

Dla wybranych trzech funkcji bez ograniczeń i trzech z ograniczeniami należy odnaleźć położenie i wartość minimum. Poszukiwania należy realizować z zastosowaniem różnych metod selekcji oraz różnych parametrów krzyżowania i mutacji. W badaniach należy przyjąć **rzeczywiste kodowanie** chromosomu.

W sprawozdaniu:

Dla każdej z sześciu badanych funkcji należy zamieścić:

- wykres powierzchni funkcji (tylko dla funkcji dwuargumentowych) oraz wykres ograniczeń (tylko dla funkcji z ograniczeniami)
- tabelę z wynikami badań

metoda selekcji	parametry krzyżowania	parametry mutacji	współczynniki kary	czas po jakim odnaleziono rozwiązanie (w krokach obliczeń)	położenie minimum	wartość minimalna funkcji

Dla każdej funkcji należy przeprowadzić kilka eksperymentów, w celu odnalezienia najlepszej metody selekcji i najlepszych parametrów krzyżowania i mutacji. Za najlepszą metodę i parametry uznawać będziemy te, które zapewniają odnalezienie minimum w najkrótszym czasie.

- najlepsze parametry poszukiwań należy wyróżnić w tabeli i dodatkowo należy zamieścić dla nich wykres obrazujący wartość najlepszego chromosomu i średnią wartość populacji w czasie poszukiwań
- wnioski

Przykładowe funkcje bez ograniczeń

Funkcje pochodzą z książki Z. Michalewicz: „Algorytmy genetyczne + struktury danych = programy ewolucyjne”.

1. Funkcja De Jonga F1:

$$\sum_{i=1}^3 x_i^2, \quad \text{gdzie } -5,12 \leq x_i \leq 5,12.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2, x_3) = (0, 0, 0)$.

2. Funkcja De Jonga F2:

$$100(x_1^2 - x_2)^2 + (1 - x_1)^2, \quad \text{gdzie } -2,048 \leq x_i \leq 2,048.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (1, 1)$.

3. Funkcja De Jonga F3:

$$\sum_{i=1}^5 \text{integer}(x_i), \quad \text{gdzie } -5,12 \leq x_i \leq 5,12.$$

Ta funkcja przyjmuje globalną wartość minimalną -30 dla wszystkich $-5,12 \leq x_i \leq -5,0$.

4. Funkcja De Jonga F4:

$$\sum_{i=1}^{30} i x_i^4 + \text{Gauss}(0, 1), \quad \text{gdzie } -1,28 \leq x_i \leq 1,28.$$

Ta funkcja (bez zakłócania gaussowskiego) przyjmuje globalną wartość minimalną 0 w $(x_1, x_2, \dots, x_{30}) = (0, 0, \dots, 0)$.

5. Funkcja De Jonga F5:

$$\frac{1}{1/K + \sum_{j=1}^{25} f_j^{-1}(x_1, x_2)}, \quad \text{gdzie } f_j(x_1, x_2) = c_j + \sum_{i=1}^2 (x_i - a_{ij})^6,$$

przy czym $-65,536 \leq x_i \leq 65,536$, $K = 500$, $c_j = j$, oraz

$$[a_{ij}] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

Ta funkcja przyjmuje globalną wartość minimalną 0,998 w $(x_1, x_2) = (-32, -32)$.

6. Funkcja Schaffera F6:

$$0,5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0,5}{[1,0 + 0,001(x_1^2 + x_2^2)]^2}, \quad \text{gdzie } -100 \leq x_i \leq 100.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (0, 0)$.

7. Funkcja Schaffera F7:

$$(x_1^2 + x_2^2)^{0,25} [\sin^2(50(x_1^2 + x_2^2)^{0,1}) + 1,0], \quad \text{gdzie } -100 \leq x_i \leq 100.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (0, 0)$.

8. Funkcja Goldsteina-Price'a:

$$\begin{aligned} & [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \cdot \\ & \cdot [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)], \\ & \text{gdzie } -2 \leq x_i \leq 2 \end{aligned}$$

Ta funkcja przyjmuje globalną wartość minimalną 3 w $(x_1, x_2) = (0, -1)$.

9. Funkcja Branina RCOS:

$$\begin{aligned} & a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f)\cos(x_1) + e, \\ & \text{gdzie } -5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15, \text{ oraz } a = 1, b = 5,1/(4\pi^2), c = 5/\pi, \\ & d = 6, e = 10, f = 1/(8\pi). \end{aligned}$$

Ta funkcja przyjmuje globalną wartość minimalną 0,397887 w trzech różnych punktach $(x_1, x_2) = (-\pi, 12,275)$, $(\pi, 2,275)$ oraz $(9,42478, 2,475)$.

10. Rodzina funkcji czterowymiarowych Shekela SQRN5, SQRN7, SQRN10:

$$s3(x_1, x_2, x_3, x_4) = - \sum_{j=1}^5 \frac{1}{\sum_{i=1}^4 (x_i - a_{ij})^2 + c_j}$$

$$s4(x_1, x_2, x_3, x_4) = - \sum_{j=1}^7 \frac{1}{4 \sum_{i=1}^4 (x_i - a_{ij})^2 + c_j}$$

$$s5(x_1, x_2, x_3, x_4) = - \sum_{j=1}^{10} \frac{1}{4 \sum_{i=1}^4 (x_i - a_{ij})^2 + c_j}$$

gdzie $0 \leq x_i \leq 10$ dla $i = 1, 2, 3, 4$, a a_{ij} oraz c_j są podane w tabl. B.1.

Te funkcje przyjmują globalne wartości minimalne w punktach

$(x_1, x_2, x_3, x_4) = (4, 4, 4, 4)$ z następującymi wartościami $s3_{min} = -10,15320$, $s4_{min} = -10,402820$, $s5_{min} = -10,53628$.

Tablica B.1. Dane dla funkcji $s3$, $s4$, $s5$

j	a_{1j}	a_{2j}	a_{3j}	a_{4j}	c_j
1	4,0	4,0	4,0	4,0	0,1
2	1,0	1,0	1,0	1,0	0,2
3	8,0	8,0	8,0	8,0	0,2
4	6,0	6,0	6,0	6,0	0,4
5	3,0	7,0	3,0	7,0	0,6
6	2,0	9,0	2,0	9,0	0,6
7	5,0	5,0	3,0	3,0	0,3
8	8,0	1,0	8,0	1,0	0,7
9	6,0	2,0	6,0	2,0	0,5
10	7,0	3,6	7,0	3,6	0,5

11. Funkcja grzbietu wielbłąda sześciogarnego:

$$\left(4 - 2,1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$$

gdzie $-3 \leq x_1 \leq 3$ oraz $-2 \leq x_2 \leq 2$.

Ta funkcja przyjmuje globalną wartość minimalną $-1,0316$ w dwóch różnych punktach $(x_1, x_2) = (-0,0898, 0,7126)$ oraz $(0,0898, -0,7126)$.

12. Funkcja Shuberta:

$$\sum_{i=1}^5 i \cos[(i+1)x_1 + i] \cdot \sum_{i=1}^5 i \cos[(i+1)x_2 + i],$$

gdzie $-10 \leq x_i \leq 10$ dla $i = 1, 2$.

Ta funkcja ma 760 lokalnych minimów, przy czym w 18 z nich przyjmuje wartość $-186,73$.

13. Funkcja Stuckmana:

$$f_8(x_1, x_2) = \begin{cases} \lfloor (\lfloor m_1 \rfloor + \frac{1}{2}) \sin(a_1)/a_1 \rfloor, & \text{jeśli } 0 \leq x_1 \leq b \\ \lfloor (\lfloor m_2 \rfloor + \frac{1}{2}) \sin(a_2)/a_2 \rfloor, & \text{jeśli } b < x_1 \leq 10, \end{cases}$$

gdzie $0 \leq x_i \leq 10$ dla $i = 1, 2$, a m_i jest zmienną losową z zakresu $(0, 100)$, b jest zmienną losową z zakresu $(0, 10)$ oraz

$$a_i = \lfloor |x_1 - r_{1i}| \rfloor + \lfloor |x_2 - r_{2i}| \rfloor,$$

gdzie r_{11} jest zmienną losową z zakresu $(0, b)$, r_{12} jest zmienną losową z zakresu $(b, 10)$, r_{21} jest zmienną losową z zakresu $(0, 10)$, a r_{22} jest zmienną losową z zakresu $(0, 10)$ (wszystkie zmienne losowe mają rozkład równomierny).

Maksimum globalne znajduje się w

$$(x_1, x_2) = \begin{cases} (r_{11}, r_{21}), & \text{jeśli } m_1 > m_2 \\ (r_{12}, r_{22}), & \text{w pozostałym przypadku} \end{cases}$$

14. Funkcja Easoma:

$$-\cos(x_1) \cos(x_2) e^{-(x_1 - \pi)^2 - (x_2 - \pi)^2}, \quad \text{gdzie } -100 \leq x_i \leq 100 \text{ dla } i = 1, 2.$$

Ta funkcja przyjmuje globalną wartość minimalną -1 w $(x_1, x_2) = (\pi, \pi)$.

15. Funkcja 1 Bohachevsky'ego:

$$x_1^2 + 2x_2^2 - 0,3 \cos(3\pi x_1) - 0,4 \cos(4\pi x_2) + 0,7, \quad \text{gdzie } -50 \leq x_i \leq 50.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (0, 0)$.

16. Funkcja 2 Bohachevsky'ego:

$$x_1^2 + 2x_2^2 - 0,3(\cos(3\pi x_1) \cos(4\pi x_2)) + 0,3, \quad \text{gdzie } -50 \leq x_i \leq 50.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (0, 0)$.

17. Funkcja 3 Bohachevsky'ego:

$$x_1^2 + 2x_2^2 - 0,3 \cos(3\pi x_1) + \cos(4\pi x_2) + 0,3, \quad \text{gdzie } -50 \leq x_i \leq 50.$$

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2) = (0, 0)$.

18. Funkcja Coldville'a:

$$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + \\ + 10,1((x_2 - 1)^2 + (x_4 - 1)^2) + 19,8(x_2 - 1)(x_4 - 1),$$

gdzie $-10 \leq x_i \leq 10$.

Ta funkcja przyjmuje globalną wartość minimalną 0 w $(x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$.

Przykładowe funkcje z ograniczeniami

Funkcje pochodzą z książki Z. Michalewicz: „Algorytmy genetyczne + struktury danych = programy ewolucyjne”.

Zadanie nr 1

Funkcja:

$$F(\mathbf{X}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Ograniczenia:

$$x_1 + x_2^2 \geq 0$$

$$x_1^2 + x_2 \geq 0$$

$$-0.5 \leq x_1 \leq 0.5 \quad \text{oraz} \quad -1.0 \leq x_2 \leq 1.0$$

Zadanie nr 2

Funkcja:

$$F(\mathbf{X}) = -x_1 - x_2$$

Ograniczenia:

$$x_2 \leq 2x_1^4 - 8x_1^3 + 8x_1^2 + 2$$

$$x_2 \leq 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 + 36$$

$$0 \leq x_1 \leq 3 \quad \text{oraz} \quad 0 \leq x_2 \leq 4$$

Zadanie nr 3

Funkcja:

$$F(\mathbf{X}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Ograniczenia:

$$(x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0$$

$$-(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 \geq 0$$

$$13 \leq x_1 \leq 100 \quad \text{oraz} \quad 0 \leq x_2 \leq 100$$

Zadanie nr 4

Funkcja:

$$F(\mathbf{X}) = 0.01x_1^2 + x_2^2$$

Ograniczenia:

$$x_1x_2 - 25 \geq 0$$

$$x_1^2 + x_2^2 - 25 \geq 0$$

$$2 \leq x_1 \leq 50 \quad \text{oraz} \quad 0 \leq x_2 \leq 50$$

Zadanie nr 5

Funkcja:

$$F(\mathbf{X}) = (x_1 - 2)^2 + (x_2 - 1)^2$$

Ograniczenia:

$$-x_1^2 + x_2 \geq 0$$

$$x_1 + x_2 \leq 2$$

$$-5 \leq x_1 \leq 5 \quad \text{oraz} \quad -5 \leq x_2 \leq 5$$

Zadanie nr 6

Funkcja:

$$F(\mathbf{X}) = \begin{cases} x_2 + 10^{-5}(x_2 - x_1)^2 - 1 & \text{dla } 0 \leq x_1 < 2 \\ \frac{1}{27\sqrt{3}}((x_1 - 3)^2 - 9)x_2^3 & \text{dla } 2 \leq x_1 < 4 \\ \frac{1}{3}(x_1 - 2)^3 + x_2 - \frac{11}{3} & \text{dla } 4 \leq x_1 \leq 6 \end{cases}$$

Ograniczenia:

$$\frac{x_1}{\sqrt{3}} - x_2 \geq 0$$

$$-x_1 - \sqrt{3}x_2 + 6 \geq 0$$

$$0 \leq x_1 \leq 6 \quad \text{oraz} \quad 0 \leq x_2 \leq 6$$

GAOT – Genetic Algorithms for Optimization Toolbox

1. Wprowadzenie

Pakiet GAOT przeznaczony jest do rozwiązywania zadań optymalizacji za pomocą algorytmów genetycznych i algorytmów ewolucyjnych. Do prawidłowego działania wymaga wersji MATLABA od 5.1 w górę.

2. Funkcja przystosowania

Funkcja przystosowania (dopasowania, oceniająca) określa w jakim stopniu dany osobnik (chromosom) rozwiązuje analizowane zagadnienie. Należy pamiętać, że pakiet GAOT poszukuje tylko **maksimum** funkcji przystosowania, stąd jeśli chcemy odnaleźć minimum funkcji należy zdefiniować ją ze znakiem „-”.

Przykładowo, (przykład z książki Z. Michalewicza: „Algorytmy genetyczne + struktury danych = programy ewolucyjne”) chcemy znaleźć maksimum funkcji:

$$f(x_1, x_2) = 21.5 + x_1 \cdot \sin(4\pi x_1) + x_2 \cdot \sin(20\pi x_2) .$$

w przedziałach: $-3.0 \leq x_1 \leq 12.1$ i $4.1 \leq x_2 \leq 5.8$.

Musimy przygotować m-plik z definicją funkcji. Dla naszego przykładu może on wyglądać następująco:

```
function [x,val] = michal(x,options)

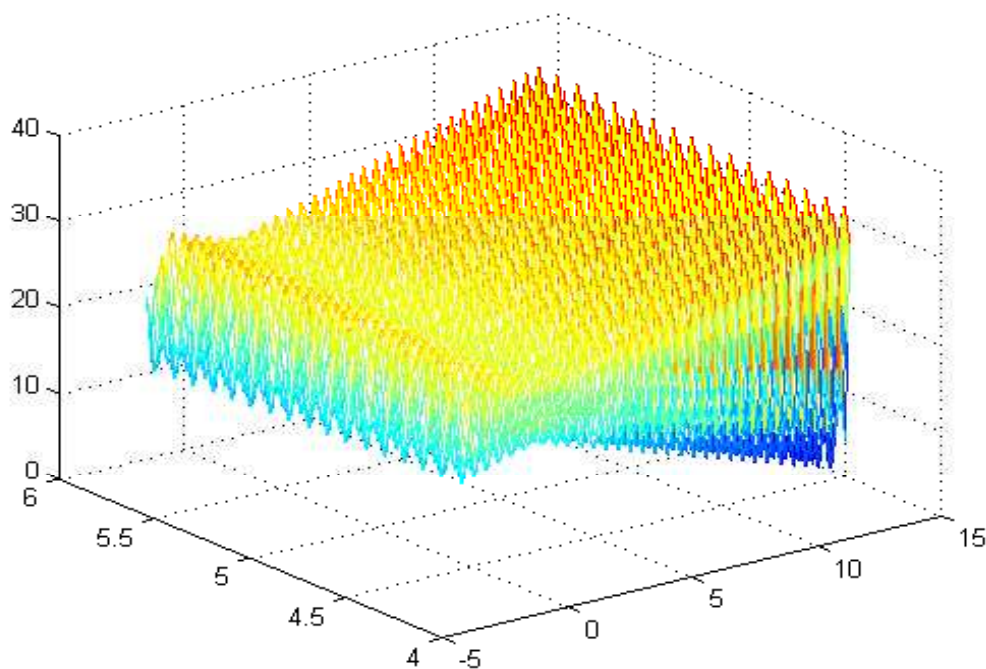
val = 21.5 + x(1)*sin(4*pi*x(1)) + x(2)*sin(20*pi*x(2));
```

Jako parametry przekazujemy do funkcji wektor zmiennych **x** i wektor opcji **options** (zwykle nie jest wykorzystywany). Funkcja musi zwracać wektor zmiennych **x** i wynik **val**.

Jeżeli badana funkcja jest funkcją 2 zmiennych można wykreślić jej powierzchnię. Służą do tego polecenia (zilustrowane na przykładzie):

```
[xx,yy]=meshgrid(-3:0.1:12.1, 4.1:0.1:5.8);
rys2D('michal',xx,yy);
```

Przykładowy wykres funkcji jaki uzyskamy:



3. Poszukiwanie maksimum funkcji

Przy poszukiwaniu maksimum funkcji najwygodniej utworzyć sobie skrypt (M-plik) MATLABA (lub zrobić kopię i przerabiać już istniejące np.: `example_real.m`).

W skrypcie należy podać szereg poleceń określających rodzaj algorytmu maksymalizacji i jego parametry. Są one kolejno wymienione.

Inicjujemy generator liczb losowych, jeśli chcemy mieć powtarzalne wyniki.

```
% Setting the seed to the same for binary
% rand('seed',156789)
```

Definiujemy zakres zmiennych (zmienna musi być globalna).

```
global bounds
% Bounds on the variables
bounds = [-3 12.1; 4.1 5.8];
```

Generujemy populację początkową

```
% Generate an initialize population of size 20 for binary version
startPop = initializega(20,bounds,'michal',[],[1e-6 1]);
```

Jako parametry przekazujemy: liczebność populacji, zdefiniowane wcześniej zakresy zmiennych, nazwę funkcji przystosowania, ewentualne parametry dodatkowe przekazywane do funkcji przystosowania i wektor z dwoma wartościami. Pierwsza – określa dokładność dla zmiennych, druga – określa typ algorytmu (0 – populacja binarna, 1 – populacja w formie liczb rzeczywistych).

Definiujemy typ i parametry krzyżowania

```
% Binary Crossover Operators
xFns = 'simpleXover';
xOpts = [0.8];

% Crossover Operators
% xFns = 'arithXover heuristicXover simpleXover';
% xOpts = [1 0; 1 3; 1 0];
```

Dla wersji binarnej możliwe jest tylko krzyżowanie proste, a jako parametr przekazujemy prawdopodobieństwo jego wystąpienia. Dla wersji rzeczywistej możliwe są 3 rodzaje krzyżowania: arytmetyczne, heurystyczne i proste. Każde z nich może mieć osobne parametry. Pierwszy określa ile razy dany rodzaj krzyżowania należy zastosować dla populacji, drugi parametr ma znaczenie tylko dla krzyżowania heurystycznego i określa ilość powtórzeń (prób) zrealizowania algorytmu.

Definiujemy typ i parametry mutacji

```
% Binary Mutation Operators
mFns = 'binaryMutation';
mOpts = [0.005];

% Mutation Operators
% mFns = 'boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation';
% mOpts = [2 0 0; 3 200 3; 2 200 3; 2 0 0];
```

Dla wersji binarnej możliwy jest tylko jeden rodzaj mutacji, a jako parametr przekazujemy prawdopodobieństwo zmutowania dla każdego genu. Dla wersji rzeczywistej możliwe są 4 rodzaje mutacji: brzegowa, wielokrotna-niejednolita, niejednolita i jednolita. Każdy rodzaj mutacji wymaga określenia ile razy dla danej populacji należy go zastosować. Dodatkowo dla mutacji niejednolitej określamy maksymalną liczbę populacji i parametr określający tzw. stopień niejednorodności.

Definiujemy sposób zatrzymania algorytmu

```
% Termination Operators
termFns = 'maxGenTerm';
termOps = [200]; % 200 Generations
```

Definiujemy rodzaj selekcji i jej parametry

<pre>% Selection Function selectFn = 'roulette'; selectOps = [];</pre>	metoda koła ruletki (bez parametrów)
<pre>% selectFn = 'normGeomSelect'; % selectOps = [0.08];</pre>	metoda rankingu chromosomów prawdopodobieństwo wyboru najlepszego osobnika
<pre>% selectFn = 'tourn'; % selectOps = [2];</pre>	metoda stochastycznej rozgrywki ilość chromosomów biorących udział w rozgrywce

Określamy funkcję przystosowania i jej ewentualne parametry

```
% Evaluation Function
evalFn = 'michal';
evalOps = [];
```

Określamy parametry algorytmu

```
% GA Options [epsilon float/binar display]
gaOpts=[1e-6 1 1];
```

Podajemy parametry: epsilon – najmniejsza zmiana potrzebna do tego aby uznać dwa rozwiązania za różne, float/binar – 1 lub 0 dla wersji z kodowaniem rzeczywistym lub binarnym, display – 1 lub 0 gdy na ekranie mają być wyświetlane informacje w trakcie obliczeń lub nie.

Uruchamiamy algorytm

```
% Lets run the GA
[x endPop bestPop trace]=ga(bounds,evalFn,evalOps,startPop,gaOpts,...
    termFns,termOps,selectFn,selectOps,xFns,xOpts,mFns,mOpts);
```

Wyświetlamy wyniki obliczeń

```
% x is the best solution found
disp('Best solution')
x

% endPop is the ending population
disp('Ending population')
endPop

% bestPop is the best solution tracked over generations
disp('Best solution tracked over generations')
bestPop

% trace is a trace of the best value and average value of generations
% trace

% Plot the best over time
clf
plot(trace(:,1),trace(:,2));
% Add the average to the graph
hold on
plot(trace(:,1),trace(:,3));
hold off
```