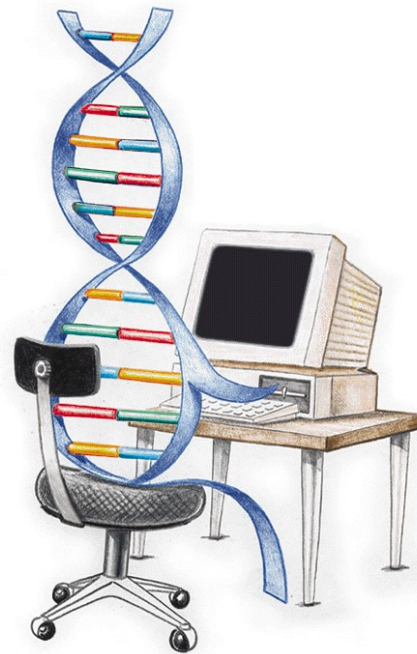


# Bevezetés a lágyszámítás módszereibe

## MATLAB - GA



# MATLAB GA használata

- Pl. tesztmérnökök tesztelési eljárást dolgoztak ki rendszerek (pl. hardver komponensek) helyességének tesztelésére: evolúciós teszt
- A tesztet optimalizálási problémaként értelmezik
- Genetikus algoritmus használata
- A GA teljesítménye és kimeneti hatékonysága több futtatásban változhat
- MATLAB környezet képes biztosítani ezt a bonyolultságot
- Interaktív használat, kevesebb idő a kódoláshoz és a hibakereséshez

# Parancssori lehetőségek

Fő függvénye a **ga** függvény, amely  $F(X)$  minimumát próbálja meg meghatározni, megadott feltételeket figyelembe véve.

$F$  a fitness függvény,  $X$  egy tetszőleges egyed ekkor az optimális megoldás egy olyan  $X$ , ahol

- 1  $A_{eq} * X = b_{eq}$  (lineáris egyenletek)
- 2  $A * X \leq b$  (lineáris egyenlőtlenségek)
- 3  $C_{eq}(X) = 0$  (nemlineáris egyenletek)
- 4  $C(X) \leq 0$  (nemlineáris egyenlőtlenségek)
- 5  $LB \leq X \leq UB$ , azaz  $X$  egy adott intervallumban keresendő

Feltételek teljesülése mellett  $F(X)$  értéke minimális.

Ezen jelölések mellett a **ga** függvény általános alakja:

$\gg [X, FVAL, REASON, OUTPUT, POPULATION, SCORES] = GA(F, NVAR, A, b, Aeq, beq, LB, UB, NONLCON, Options)$

A fent nem definiált jelölések jelentése:

- **NVAR**:  $F$  függvény függvényváltozóinak száma
- **NONLCON**:  $|C(X)$  és  $C_{eq}(X)$  függvényeket megvalósító Matlab függvény (ezeket ált. magunknak kell implementálnunk)
- **FVAL**:  $F(X)$ , a kimenő megoldásegységre
- **REASON**: a kilépés okának leírása
- **OUTPUT**: a futás körülményeiről ad néhány információt ez a struktúra
- **POPULATION**: a kilépéskor meglévő populáció
- **SCORES**: a kilépéskor meglévő populáció fitness értékei
- **options**: az algoritmus paramétereit tartalmazó struktúra

# Egyszerű példa

A sikeres futtatáshoz elegendő  $F$  és  $NVARS$  megadása is:

```
>> ga(@(x)x * x, 1)
```

Ez az  $f(x) = x^2$  függvény abszolút minimumát keresi, ami  $x = 0$  helyen található.

```
>> ga(@(x) x*x, 1)
```

```
Optimization terminated: maximum number of generations exceeded.
```

```
ans =
```

```
-4.7080e-005
```

A futás egészen közeli eredményt talált.

A leállás a generációszám maximális értékének túllépése miatt történt.



# Bonyolultabb példa

Matlabék (és talán mások) kedvence a **Rastrigin-függvény**, ha a genetikus algoritmusokról van szó.

Ez egy  $n$ -változós valós függvény, amely  $n$  függvényében a következő formulával adható meg:

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$$

Ez a függvény folytonos, differenciálható, és könnyen észrevehető, hogy minimuma  $x = 0$  pontban van, értéke  $f(x) = 0$ .

Ezen kívül hála a  $\cos$  függvény periodicitásának, a függvény tele van lokális minimumhelyekkel, ami igen rossz hatással lehet a kereső algoritmusokra.

A függvény megtalálható **rastriginsfcn** néven a Matlab tárházában.

```
function scores = rastriginsfcn(pop)
%RASTRIGINFCN Compute the "Rastrigin" function.
% Copyright 2003-2004 The MathWorks, Inc.
% pop = max(-5.12,min(5.12,pop));
scores = 10.0 * size(pop,2) + sum(pop .^2 - 10.0 * cos(2 * pi .* pop),2);
```

## GA futtatása a Rastrigin-függvényre:

```
>> [x,FVAL]=ga(@rastriginsfcn,2)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
```

```
x =
-0.0202 -0.0637
```

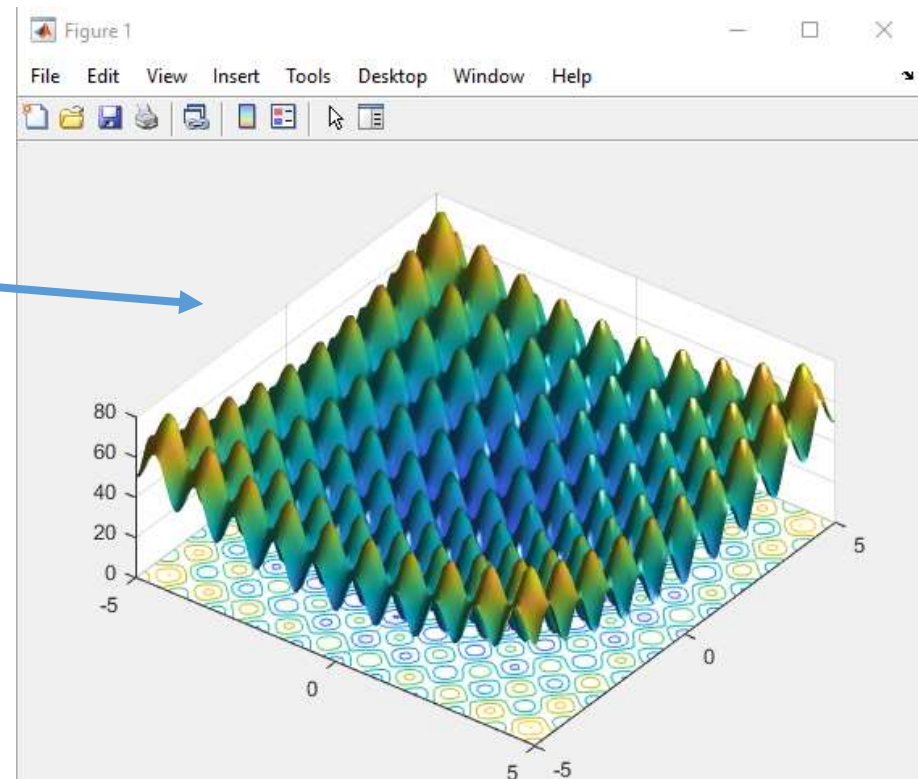
Közel optimális megoldást  
kaptunk (globális optimum)

```
FVAL =
0.8743
```

**fsurf(@(x,y)reshape(rastriginsfcn([x(:),y(:)]),size(x)))**

A Rastrigin-függvény képe  
( $n=2$ ,  $-5 \leq x_1, x_2 \leq 5$ ):

Számos lokális optimum hely van  
a globális optimum hely mellett



# Hasonlítsuk össze a genetikus algoritmus és egy kereső algoritmus működését!

```
>> [x,FVAL]=ga(@rastriginsfcn,2)
Optimization terminated: maximum number of generations exceeded.

x =

    0.0051   -0.0071

FVAL =

    0.0151
```

Az eredmény elfogadható első neki futásra.  
Érdeemes megnézni egy konkurens kereső algoritmust. Vegyük a Matlab **fminsearch** függvényét, nézzük mit tud kezdeni vele:

```
>> fminsearch(@rastriginsfcn, rand(1,2))

ans =

    0.9950    0.9950
```

A keresés beleesett egy lokális minimum körüli gödörbe.



## Egy kétváltozós valós fg. - Shufcn fg.

```
function f = shufcn(y)
%SHU objective function.
%
% L.C.W. Dixon and G.P. Szegö (eds.), Towards Global Optimisation 2,
% North-Holland, Amsterdam 1978.

% Copyright 2004 The MathWorks, Inc.
% $Revision: 1.1 $ $Date: 2004/01/14 15:35:06 $

for j = 1: size(y,1)
    f(j) = 0.0;
    x = y(j,:);
    temp1 = 0;
    temp2 = 0;
    x1 = x(1);
    x2 = x(2);
    for i = 1:5
        temp1 = temp1 + i.*cos((i+1).*x1+i);
        temp2 = temp2 + i.*cos((i+1).*x2+i);
    end
    f(j) = temp1.*temp2;
end
```

# Shufcn függvény vizsgálata (GA használata)

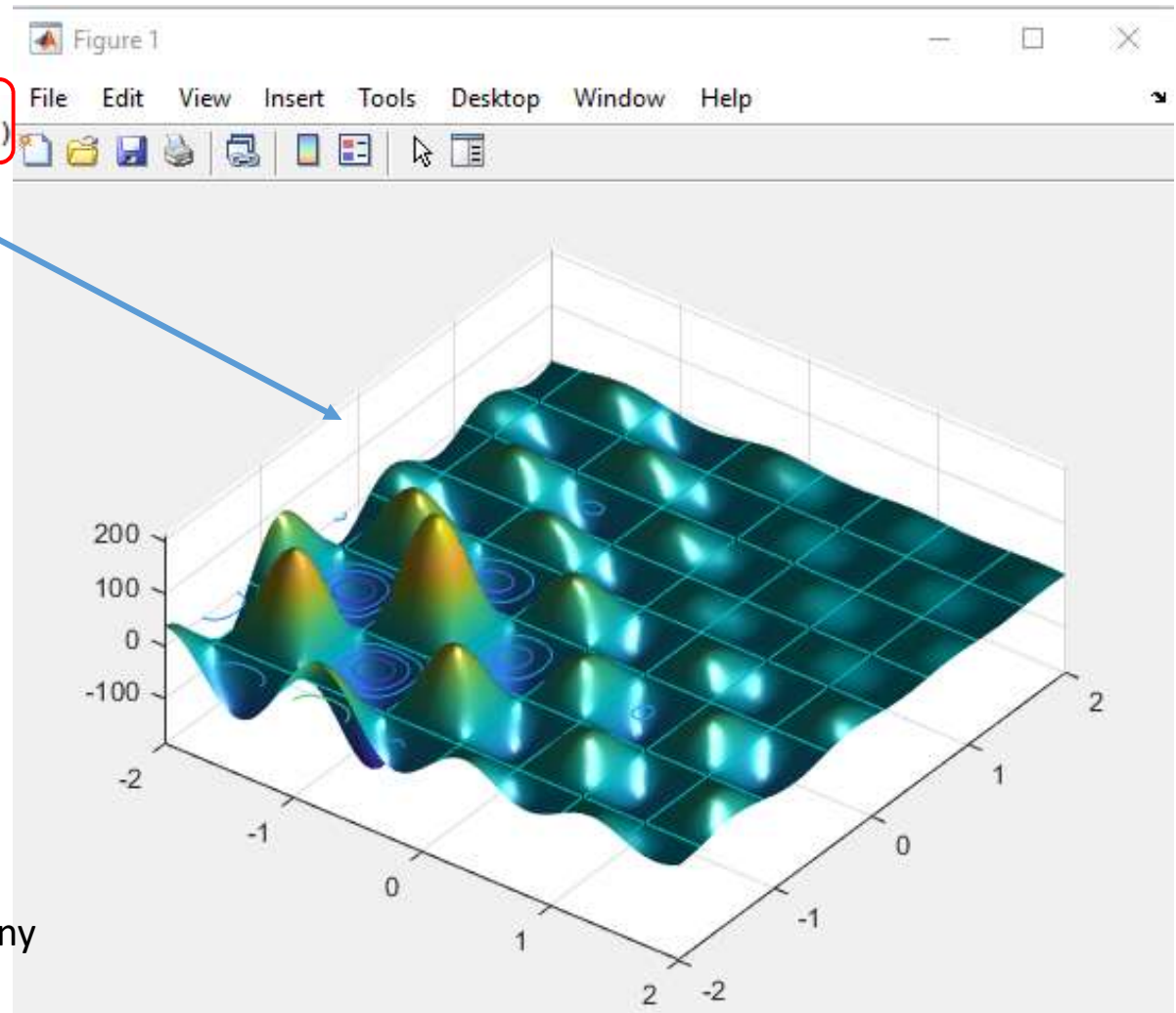
```
>> x=ga(@shufcn,2)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

-51.0625 -32.8263
```

```
>> plotobjective(@shufcn,[-2 2;-2 2])
```

```
>> FitnessFunction=@shufcn
FitnessFunction =
function_handle with value:
@shufcn
>> numberofvariables=2
numberofvariables =
2
```



A GA függvény futtatása előtt a függvény változóit definiáljuk

# Shufcn függvény - futtatás

```
>> [x,Fval,exitFlag,Output]=ga(FitnessFunction,numberofvariables)
```

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
```

```
x =
```

```
-13.9945  36.9044
```

```
Fval =
```

```
-186.6390
```

```
exitFlag =
```

```
1
```

```
Output =
```

```
struct with fields:
```

```
    problemtype: 'unconstrained'
```

```
        rngstate: [1×1 struct]
```

```
    generations: 59
```

```
        funccount: 3000
```

```
        message: 'Optimization terminated: average change in the fitness value less than options.FunctionTolerance.'
```

```
    maxconstraint: []
```

```
>> FitnessFunction=@shufcn
```

```
FitnessFunction =
```

```
function_handle with value:
```

```
@shufcn
```

```
>> numberofvariables=2
```

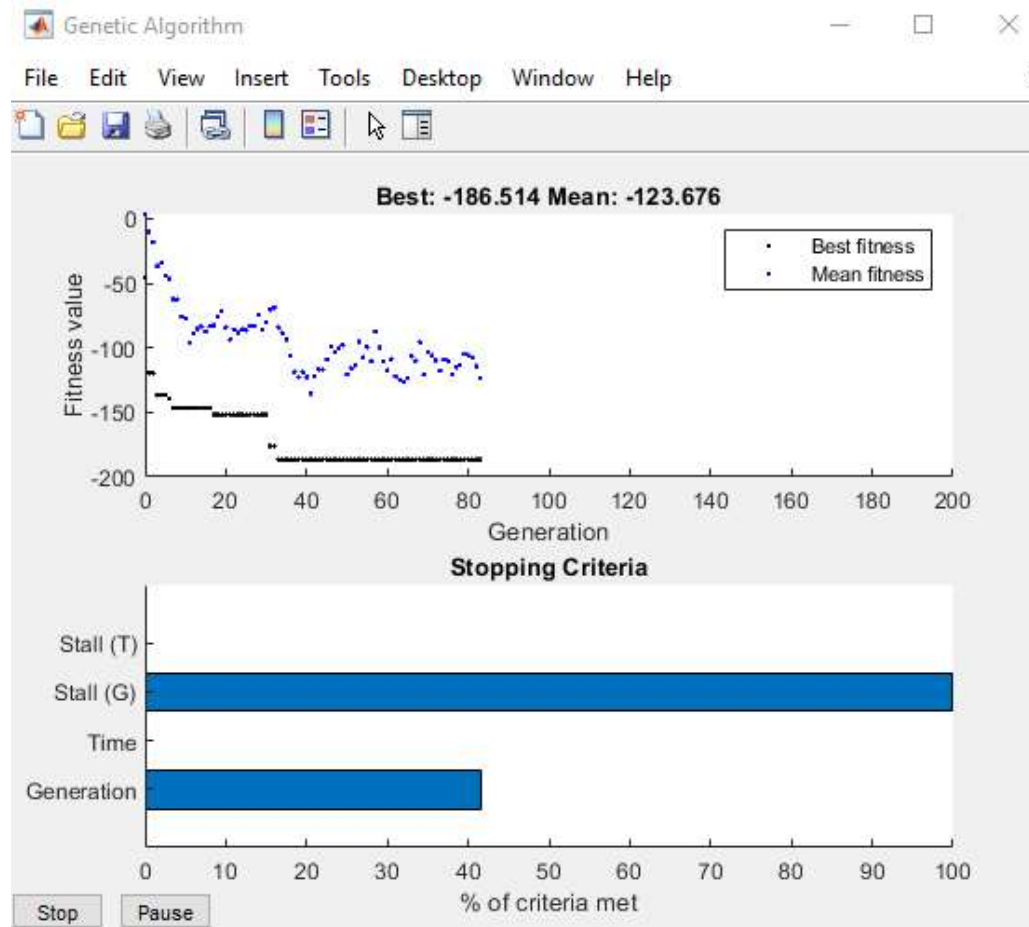
```
numberofvariables =
```

```
2
```

# Paraméterek módosítási lehetőségei

```
>> opts=gaoptimset('PlotFcns',{@gaplotbestf,@gaplotstopping})
```

```
>> [x,Fval,exitFlag,Output]=ga(FitnessFunction,numberofvariables,opts)  
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
```



Az algoritmus paramétereit tartalmazó struktúra ('opts' felépítése):

```
>> opts=gaoptimset('PlotFcns',{@gaplotbestf,@gaplotstopping})

opts =

  struct with fields:

    PopulationType: []
    PopInitRange: []
    PopulationSize: []
    EliteCount: []
    CrossoverFraction: []
    ParetoFraction: []
    MigrationDirection: []
    MigrationInterval: []
    MigrationFraction: []
    Generations: []
    TimeLimit: []
    FitnessLimit: []
    StallGenLimit: []
    StallTest: []
    StallTimeLimit: []
    TolFun: []
    TolCon: []
    InitialPopulation: []
    InitialScores: []
    NonlinConAlgorithm: []
    InitialPenalty: []
    PenaltyFactor: []
    PlotInterval: []
    CreationFcn: []
    FitnessScalingFcn: []
    SelectionFcn: []
    CrossoverFcn: []
    MutationFcn: []
    DistanceMeasureFcn: []
    HybridFcn: []
    Display: []
    PlotFcns: {@gaplotbestf @gaplotstopping}
    OutputFcns: []
    Vectorized: []
    UseParallel: []
```





# GA futtatása és a közbülső változások figyelemmel kísérése

```
>> [x,Fval,exitFlag,Output]=ga(FitnessFunction,numberofvariables,opts)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
```

```
x =
    4.8456   -0.7991
```

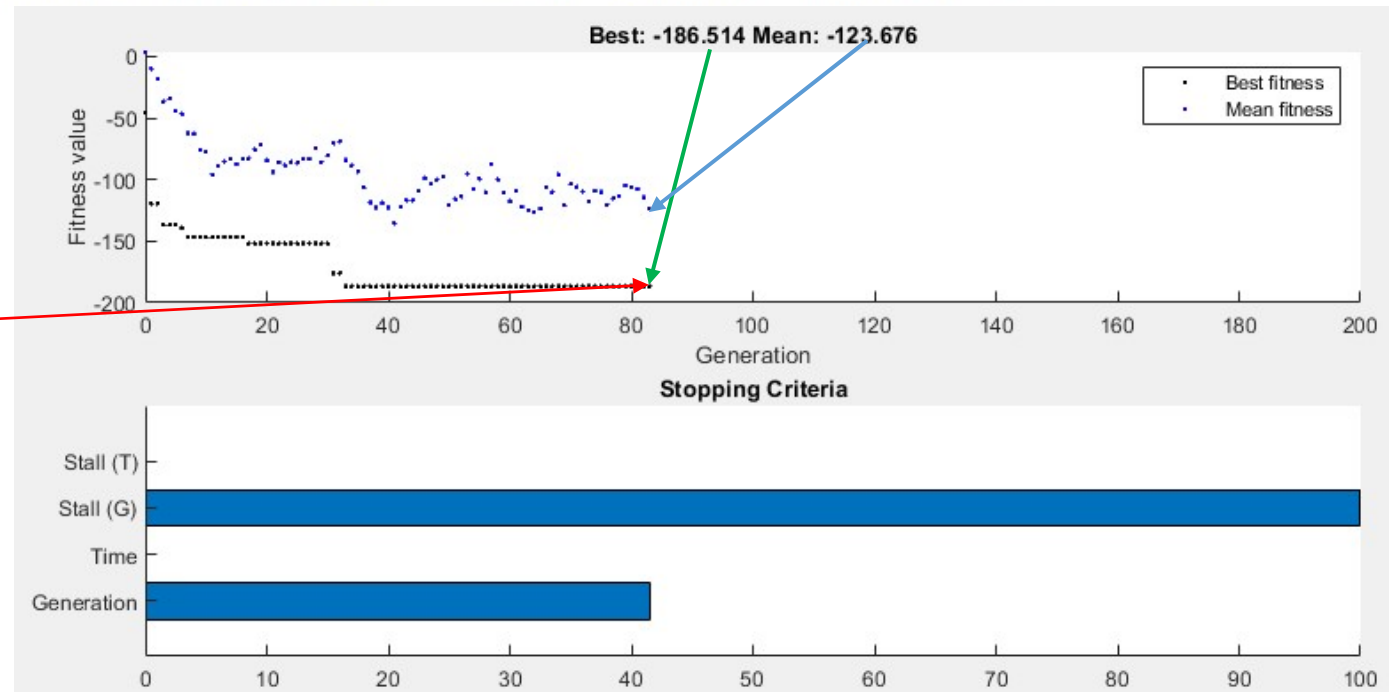
```
Fval =
-186.3652
```

```
exitFlag =
    1
```

```
Output =
```

struct with fields:

```
    problemtype: 'unconstrained'
         rngstate: [1x1 struct]
    generations: 88
         funccount: 4450
         message: 'Optimization terminated: average change in the fitness value less than options.FunctionTolerance.'
    maxconstraint: []
```

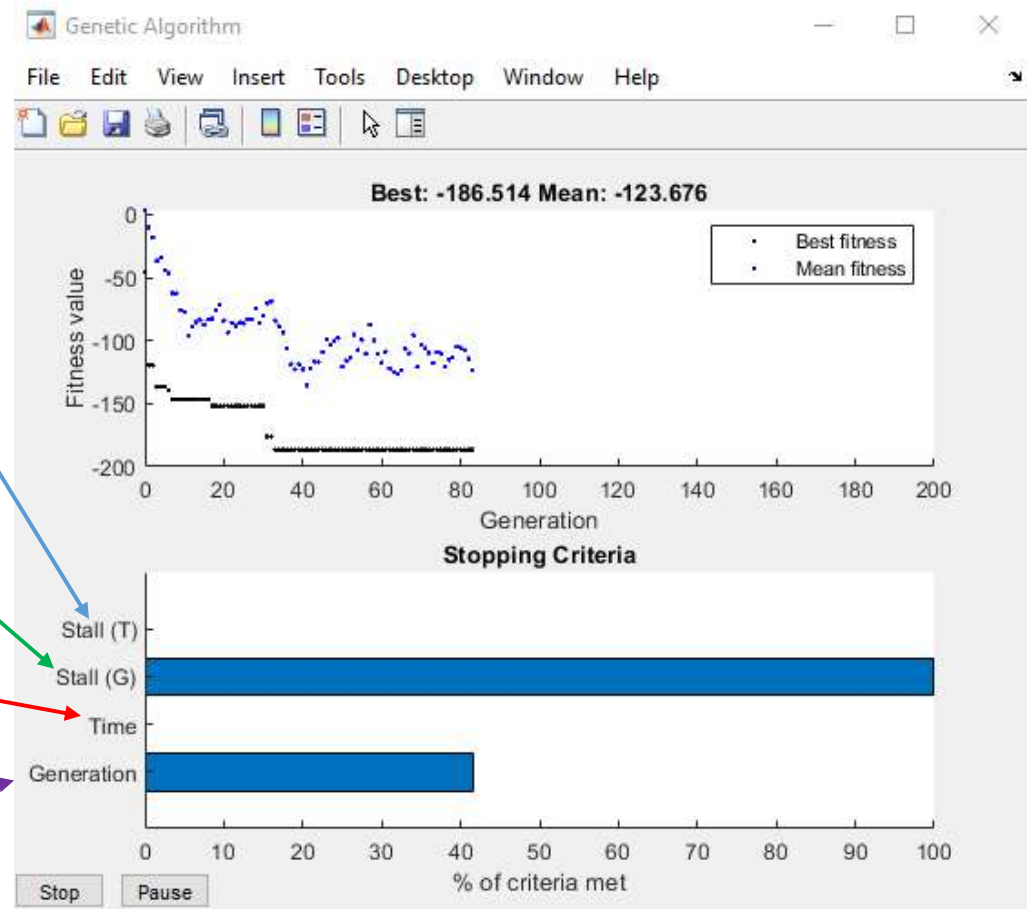


'Stall time limit': Az algoritmus leáll, ha a célfüggvény nem javul egy adott időkorláton belül

'Stall generations': Az algoritmus leáll, ha a fitness függvény értékének átlagos relatív változása a függvény tűrés határa alatt van

'Time limit': Az algoritmus leáll, ha a futás elért egy megadott időkorlátot

'Generations': Az algoritmus leáll, ha a futás elért egy megadott generáció számot

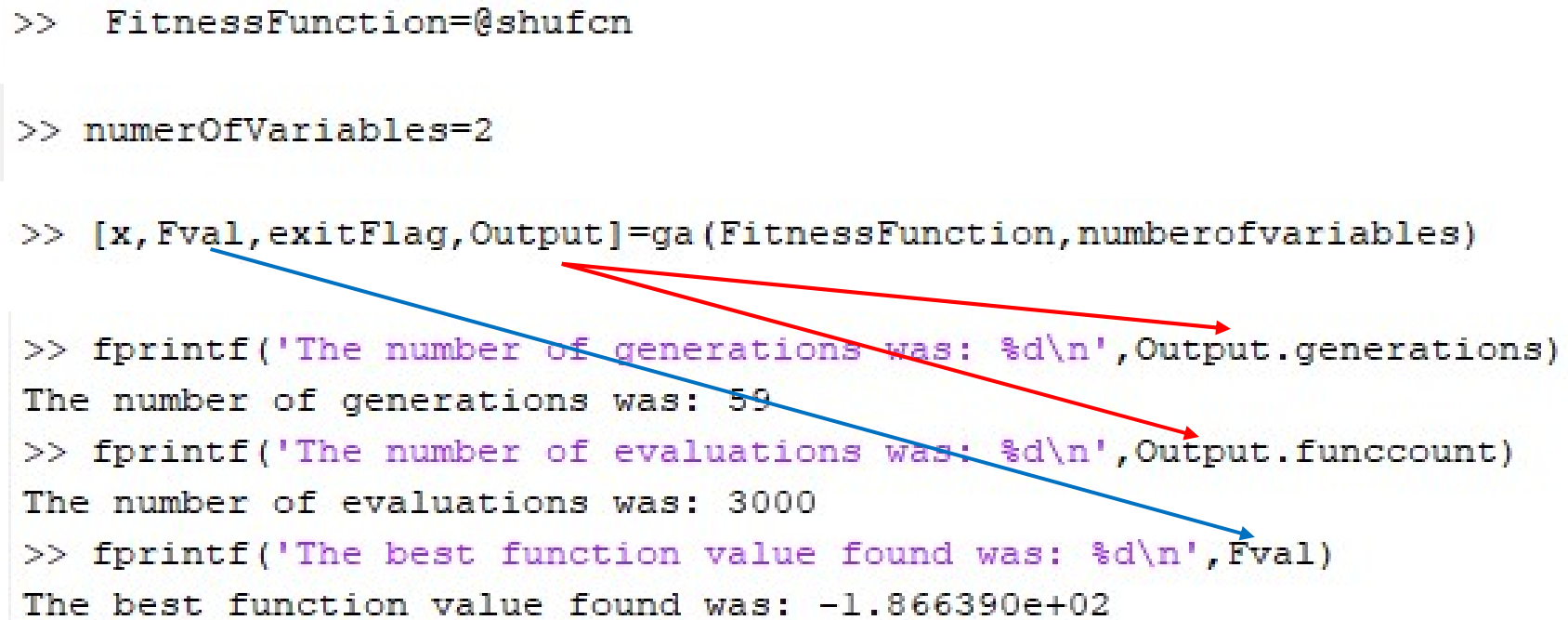


```
>> FitnessFunction=@shufcn

>> numberOfVariables=2

>> [x,Fval,exitFlag,Output]=ga(FitnessFunction,numberOfVariables)

>> fprintf('The number of generations was: %d\n',Output.generations)
The number of generations was: 59
>> fprintf('The number of evaluations was: %d\n',Output.funccount)
The number of evaluations was: 3000
>> fprintf('The best function value found was: %d\n',Fval)
The best function value found was: -1.866390e+02
```



# Víztisztító berendezés irányítása

- Egy víztisztítónak tiszta vizet kellett nyerni egy folyó zavaros vizéből.
  - A folyóvizet egymás után 3 tartályban kémiai anyagokkal kezelik, szűrik.
    1. tartály: kémiai anyagokkal pl. klór
    2. tartály: kémiai anyagokkal megkötik és üleptik a szennyeződések
    3. tartály: tovább szűrik a vizet
- Tartályonként 3-5 órás kezelés.  
A maradandó szennyezettség foka függ
- a folyóvíz korábbi szennyezettségi fokától
  - az első tartályba kerülő víz  $T_1$  mennyiségétől

# Víz tisztító berendezés

**Szabályozó működése:** a víz  $SZ1$  szennyezettségfokának függvényében meghatározza azt a  $T1$  mennyiséget, amelynél a kezelt víz szennyezettségfoka  $SZ2$  alatt marad

- A szabályozót mérési adatok elemzésével határozták meg: a víz szennyezettségfoka  $SZ1$ ,  $SZ2$  mellett figyelembe vették
  1. a víz lugosságát:  $AL$
  2. ph-értékét:  $PH$
  3. hőmérsékletét:  $TE$

Adatgyűjtésből származó adatok elemzése.

Függvényillesztéssel meghatározták a Sugano-modell keresett paramétereit.



# Vízisztító tesztadatai

SZ1	PH	TE	AL	T1	SZ2
10	7.1	18.8	53	1300	1.0
17	7.0	18.6	50	1300	1.0
22	7.3	19.4	46	1400	2.0
50	7.1	19.5	40	1400	1.0
9	7.3	23.3	48	900	40
11	7.1	20.7	50	900	1.0
12	7.2	21.3	50	900	3.0
14	7.2	23.6	53	900	40
35	7.0	17.8	35	1200	1.0
20	7.0	16.6	40	1100	1.0
20	6.9	17.8	42	1100	1.0
18	7.1	17.3	40	1100	1.0
12	7.2	18.8	55	900	3.0
8	7.2	18.0	50	1000	1.5
11	7.1	19.2	49	1000	2.0
50	7.0	18.0	37	1200	1.5
35	7.0	17.7	42	1200	1.5
30	7.0	17.3	41	1100	1.5
16	7.1	19.3	42	1100	3.0

$$6,9 \leq x_1 \leq 7,3$$

$$35 \leq x_2 \leq 55$$

$$16,6 \leq x_3 \leq 23,6$$

$$8 \leq x_4 \leq 50$$

$$1 \leq x_5 \leq 4$$

$x_1 \rightarrow \text{PH}$ ,  $x_2 \rightarrow \text{AL}$ ,  $x_3 \rightarrow \text{TE}$ ,  $x_4 \rightarrow \text{SZ1}$ ,  
 $x_5 \rightarrow \text{SZ2}$

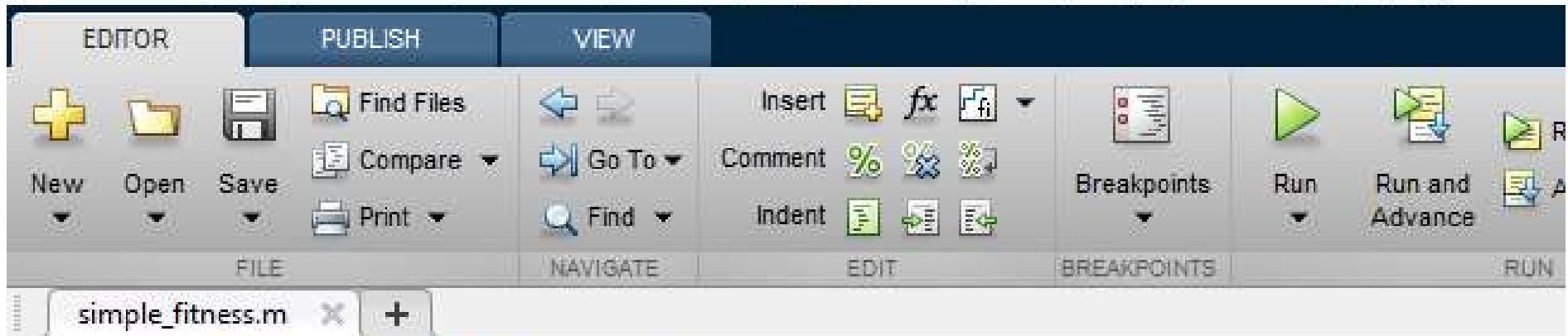
$$T_1 = p_0 + p_1 x_4 + p_2 x_5 + p_3 x_1 + p_4 x_2 + p_5 x_3$$

# Lehetséges paraméter változatok

$p_0, p_1, p_2, p_3, p_4, p_5$  paraméterek, amelyek változhatnak a PH, AL, TE értékeinek függvényében

(függvény illesztéssel meghatározzák a modell hiányzó paramétereit)

$R_i$	PH	AL	TE	$p_0$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$
1	K	K	K	8858	2664	-8093	11230	-1147	-2218
2	K	K	N	-7484	124	-427	761	52	-17
3	K	N	K	7270	42	-54	-1368	10	158
4	K	N	N	2202	5	-34	-221	-8	40
5	N	K	K	-13918	3	-6	2110	-3	2
6	N	K	N	770	22	11	64	-8	-9
7	N	N	K	-14819	159	-14	2337	-25	-68
8	N	N	N	-317	-13	-16	29	6	41



The image shows the MATLAB Editor interface. At the top, there are three tabs: EDITOR, PUBLISH, and VIEW. Below the tabs is a toolbar with various icons and labels. The labels are: FILE (New, Open, Save), NAVIGATE (Find Files, Compare, Print, Go To, Find), EDIT (Insert, Comment, Indent), BREAKPOINTS (Breakpoints), and RUN (Run, Run and Advance). The main window shows a single tab for 'simple\_fitness.m'. The code editor contains the following MATLAB code:

```
1 function y = simple_fitness(x)
2 - y = 2202+5*x(4)-34*x(5)-221*x(1)-8*x(2)+40*x(3);

>> ObjectiveFunction = @simple_fitness
>> nvars = 5
>> LB = [6.9 35 16.6 8 1]
>> UB = [7.3 55 23.6 50 4]
>> [x,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance

x =

    7.3000    55.0000    16.6000     8.0009     4.0000

fval =

    716.7048
```

```
>> [x,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    7.3000    55.0000    16.6000     8.0009     4.0000

fval =

    716.7048
```

---

```
>> [x,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    7.3000    55.0000    16.6000     8.0007     4.0000

fval =

    716.7042
```

---

```
>> [x,fval] = ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB)
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

    7.3000    55.0000    16.6000     8.0011     4.0000

fval =

    716.7060
```

# Készítsünk el egy tetszőleges szöveg kitenyésztésére szolgáló genetikus algoritmust!

- Megpróbáljuk genetikusan „kitenyészteni” sztringek egy sorozatából a kívánt mondatot.
- A populáció egyedei azonos hosszúságú sztringek, amiket genotípusosan ábrázolunk. Minden sztringhez hozzárendelünk egy sorvektort, amelynek elemei (azaz a gének) a sztring egyes karaktereinek ASCII kódjai.
- A fitness érték az egyed és a keresett sztring távolsága, azaz az egyes betűk közötti távolságok összege.
- Minél kisebb a fitness érték, annál rátermettebb az adott egyed. Nyilvánvaló, hogy ha ez az érték nulla, akkor elértük a keresett sztringet.



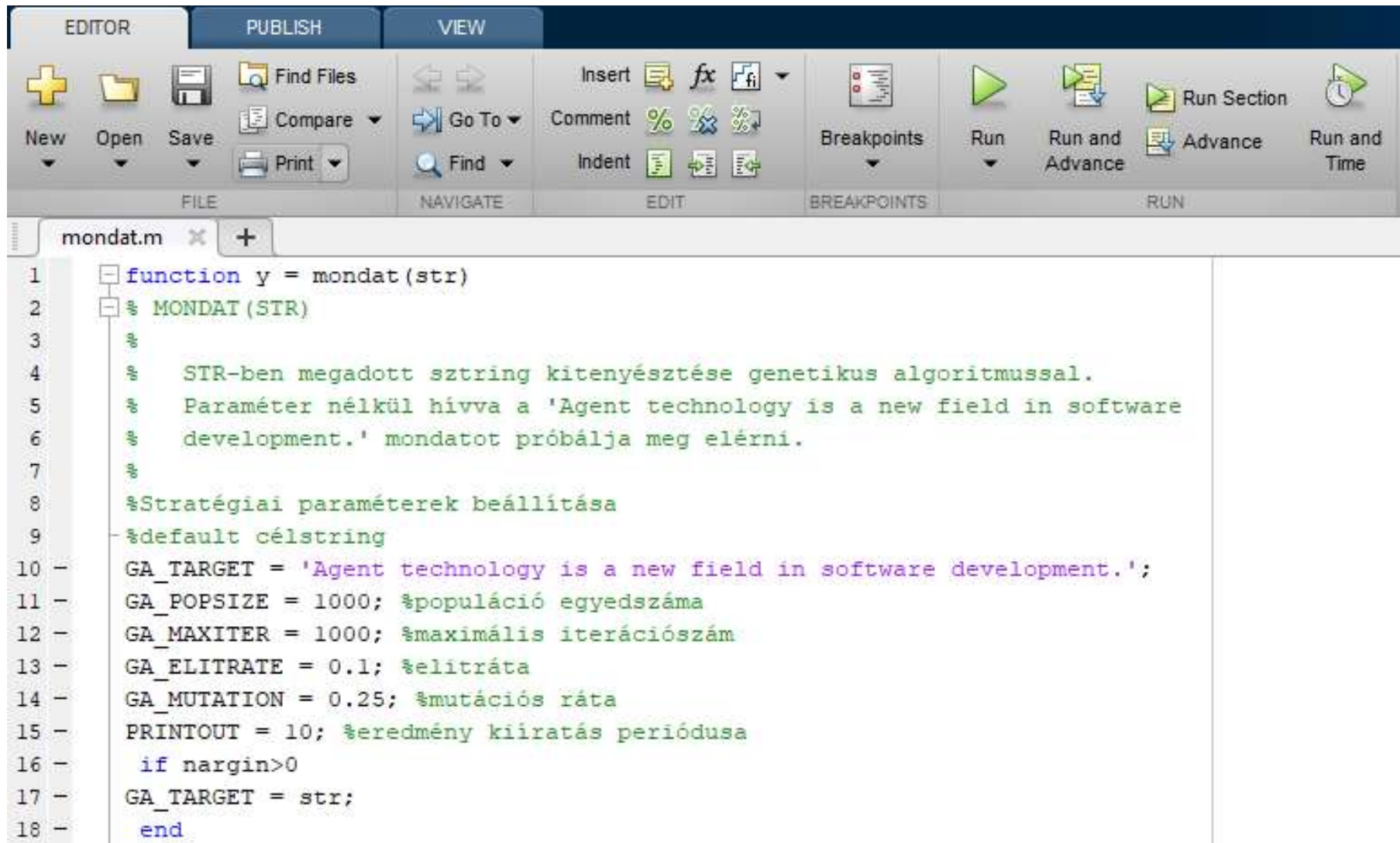
# Az algoritmushoz

- Az algoritmus fő ciklusában sorba rendezzük a populáció egyedeit növekvő fitness érték alapján.
- A **legjobbkat automatikusan beválogatjuk** a következő populációba (elitizmus).
- A maradék helyek feltöltéséhez szelekció során kiválogatjuk a szülőket. A **szelekció** ebben az esetben egyszerű **véletlen kiválasztás**, ami nem foglalkozik az önreprodukcióval (mindkét szülő ugyanaz az egyed).
- A kiválasztott szülőket **egyponos keresztezéssel rekombináljuk**, a keletkező egyedeket behelyezzük az új populációba.
- Legvégül egy előre beállított mutációs ráta szerint módosítjuk a sztringeket. A **mutáció** megvalósítása **véletlen génmutáció**, egyetlen génre.

# Az algoritmushoz

- A kapott új populációra kiértékeljük a rátermettségi függvényt, az egyedeket behelyettesítjük a régi populáció helyére, növeljük a generációs számot és ezzel le is zárul a fő ciklus.
- Kérdés még, hogy milyen **kilépési feltételt** lehet alkalmazni? Két dolgot veszünk figyelembe: **egy előre megadott generációs szám ill. az optimális megoldás elérését** (fitness érték nulla).
- Az alapbeállítás **1000 példányos populáció**kkal dolgozik.
- A populáció legjobb **10%**-át válogatjuk az **elitek** közé minden lépésben.
- A **mutáció esélye 0,25** egyedenként.
- Ha elérjük a **100 generációt** optimális eredmény nélkül, a függvény leáll, és az adott pillanatban legjobb egyed adja vissza közelítő megoldásként.

A függvény definiálása, alapértékek beállítása, a szöveg helyére tetszőleges karaktersorozat írható



The screenshot shows the MATLAB editor interface. The top toolbar includes tabs for EDITOR, PUBLISH, and VIEW. Below the tabs are icons for file operations (New, Open, Save, Find Files, Compare, Print), navigation (Go To, Find), editing (Insert, Comment, Indent), breakpoints, and running (Run, Run and Advance, Run Section, Advance, Run and Time). The main window displays a file named 'mondat.m' with the following code:

```
1 function y = mondat(str)
2 % MONDAT(STR)
3 %
4 %   STR-ben megadott sztring kitenyésztése genetikussal.
5 %   Paraméter nélkül hívva a 'Agent technology is a new field in software
6 %   development.' mondatot próbálja meg elérni.
7 %
8 %Stratégiai paraméterek beállítása
9 %default célstring
10 - GA_TARGET = 'Agent technology is a new field in software development.';
11 - GA_POPSIZE = 1000; %populáció egyedszáma
12 - GA_MAXITER = 1000; %maximális iterációszám
13 - GA_ELITRATE = 0.1; %elitráta
14 - GA_MUTATION = 0.25; %mutációs ráta
15 - PRINTOUT = 10; %eredmény kiiratás periódusa
16 -     if nargin>0
17 - GA_TARGET = str;
18 -     end
```

```

19     % t := 0, populációk számának inicializálása
20 -   t = 0;
21     % P kezdeti populáció létrehozása
22 -   P = floor (rand(GA_POPSIZE, length(GA_TARGET)) * 96 + 32);
23 -   F = sum(abs(P-ones(GA_POPSIZE,1)*GA_TARGET), 2);
24     % WHILE NOT Kilépési_feltétel(P)
25 -   while t < GA_MAXITER & F(1) ~= 0 % kilépési feltétel: iterációk száma vagy megoldás
26     % Aktuális populáció sorbarendezése
27 -     [F, I] = sort(F);
28 -     for i = 1 : length(I)
29 -     TMP(i, :) = P(I(i), :);
30 -     end
31 -     P = TMP;
32 -     y = char( P(1, :)); % legjobb egyed
33     % Az algoritmus futásáról tájékoztatjuk a felhasználót
34 -     if rem (t, PRINTOUT) == 0
35 -     str = sprintf('%d. generáció legjobb egyede: %s fitness: %d', t, y, F(1));
36 -     disp (str);
37 -     end
38     % Elitek beválogatása
39 -     elitek = floor (GA_ELITRATE * GA_POPSIZE); Puj(1:elitek,:) = P(1:elitek,:);

```



HOME PLOTS APPS

New Script New Live Script New Open Compare Import Data Save Workspace New Variable Open Variable Clear Workspace Favorites Analyze Code Run and Time Clear Commands Simulink Layout

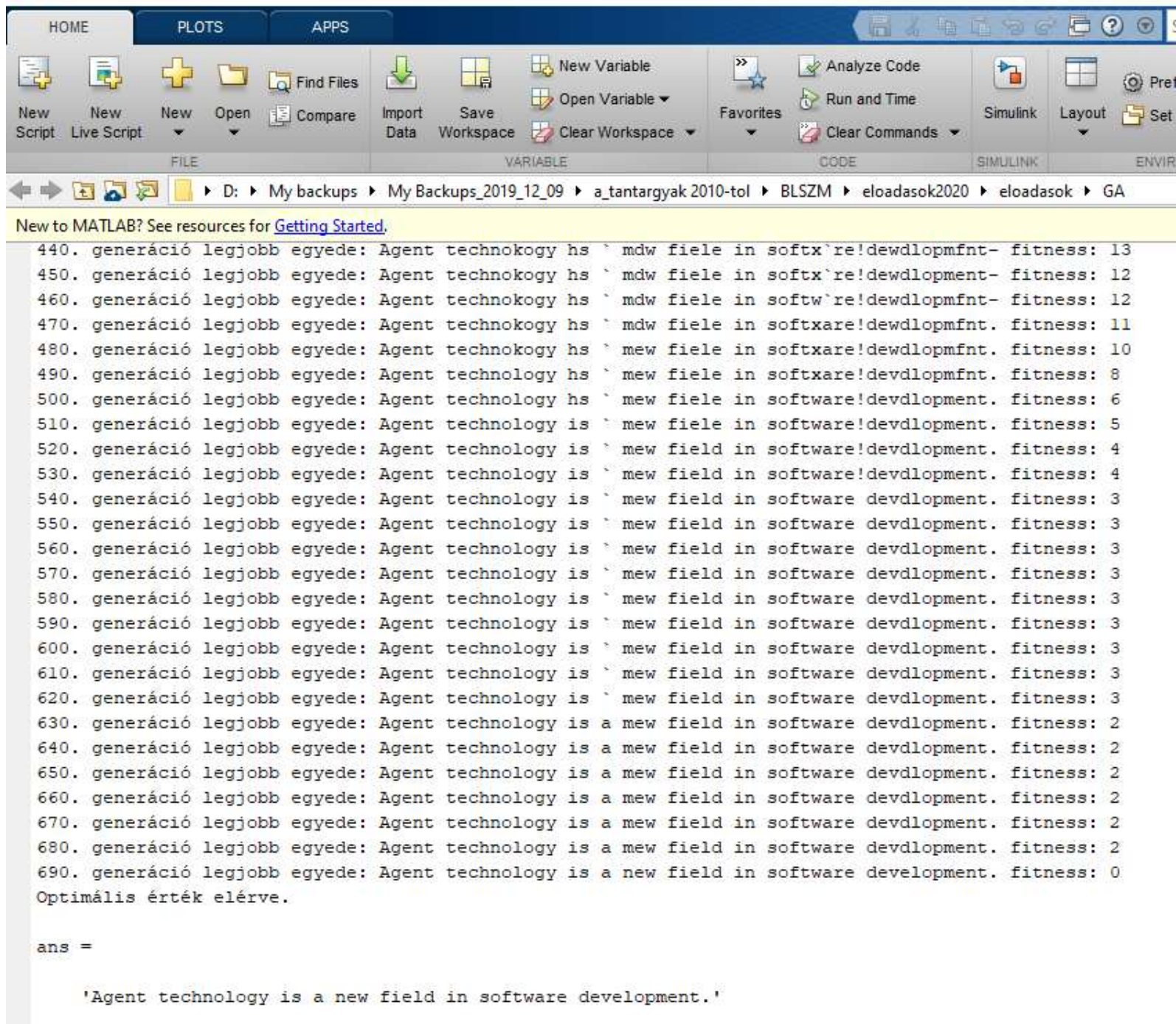
FILE VARIABLE CODE SIMULINK ENVIRONMENT

D:\My backups\My Backups\_2019\_12\_09\ a\_tantargyak 2010-tol \BLSZM \eloadasok2020 \eloadasok \GA

New to MATLAB? See resources for [Getting Started](#).

```
>> mondat
0. generáció legjobb egyede: 5rxCj>;C*Zy;&2]S'Uv'_gWla^_3]eu-gs@w)<=W(Lw!<sxa^yl>\~.5 fitness: 1435
10. generáció legjobb egyede: &\_QX5Zn d^pho']*Ca&ze[d^)\g(sr5vg%W(rPv(V[:ib{qWn:ntvt^ fitness: 1084
20. generáció legjobb egyede: .iXjQD{\`ZYDkofm)azTn~mz | *sr0j19kqaM\ji`:=i{qGhpcEjvW fitness: 917
30. generáció legjobb egyede: &\_Q#-bHih^phofm)azKn8~m~ ^SJua/|y'tY9om`ip&PsxibrvoLcm- fitness: 725
40. generáció legjobb egyede: AlQ}j-`Hih^phofm)az!h4[d] ^a@se5i^,tY|ay{\`!a\jpDinoLcm- fitness: 554
50. generáció legjobb egyede: 8t^kx'}\ih^phofm)az/n8lmz yi`ui2|y'tYcom`ip&Psxibrdofvt- fitness: 428
60. generáció legjobb egyede: AmYkx'}\ih^phofm)az!e"pRu yi`ui2|y'tYcom`gx!a\xibrvohcm- fitness: 371
70. generáció legjobb egyede: 8i^kx'}\ihlphofm)ax!e"pdz [i`xu(m!kqhom`gp!g[wigrwZfvt- fitness: 300
80. generáció legjobb egyede: Alkjr rbih^phofm)ax!e"pdz [i`xu!m!kqhom`mU![]wiqnwofvt- fitness: 263
90. generáció legjobb egyede: Alkjr rbih^pkof})ax!h"pdz [i`xu(mo!kqhom`ji!g[wiqnpoflt- fitness: 219
100. generáció legjobb egyede: Alkjr rbihlphofm)ax!e"pdz [i`je!mk!kqct(`il!^[wiqnwoflt- fitness: 186
110. generáció legjobb egyede: Flkjr rbihlphoft)ax!h"pdz [i`je!mk!kqct(`ji!g[wiqnpoflt- fitness: 173
120. generáció legjobb egyede: Aikjr rbihlpkof})ax!e"pdz di`je!mk!kqct(`il!gewiqnpoflt- fitness: 143
130. generáció legjobb egyede: Agkjr rbihlpkofw aq!e"pdz [i`je!mo!kqht(`ji!gewiqnpoflt- fitness: 129
140. generáció legjobb egyede: Aikjr rbihlpkofw aq!e"pdz dicje!mk!qqct(`ji!gewiqnpoflt- fitness: 116
150. generáció legjobb egyede: Agkjr rbihlpkofw aq!e"pdz di`je!ik!kqct(`pi!gewiqnmpflt- fitness: 111
160. generáció legjobb egyede: Agknr rb_hlnhofw aq!e"pdz ficje!mk!qqct(`pi!gewiqnpofmt- fitness: 102
170. generáció legjobb egyede: Agfjr rfihlpkofw aq!e"pdz ficje!it!qqct(`pi!gewiqnmpflt- fitness: 96
180. generáció legjobb egyede: Agfnr rfihlpkofw as!e"pdz dicje!ik!qqct(`pi!gwdqnpoflt- fitness: 88
190. generáció legjobb egyede: Agfnr rf_hnpkofw pq!e"pdz dicje!ik!qqct(`pi!gwdqnpoflt- fitness: 83
200. generáció legjobb egyede: Agfnr rf_hlpkofw mq!`"pdz ficje!ip!qqct(`pi!gwdqnpoflt- fitness: 76
210. generáció legjobb egyede: Agfnt rf_hnpkofw ks!e"pdz ficje!ik!qqctx`pi!gwdqnpoflt- fitness: 71
220. generáció legjobb egyede: Agfnt rb_hnpkofw hq!e pdz ficje!ik!qqctx`pi!gwdmnpoflt- fitness: 66
230. generáció legjobb egyede: Agfnt rf_gnpkofw hq!e"pdz ficke!in!qqctx`pi!gwdmnpofmt- fitness: 62
240. generáció legjobb egyede: Agfnt rf_hnpkofw ks!`"pdz ficje!ik!qqctx`pi!dewdnpofmt- fitness: 58
250. generáció legjobb egyede: Agfnt re_hnpkofw ks!`"pdz ficke!in!qqctx`pi!dewdnpoflt- fitness: 54
260. generáció legjobb egyede: Agfnt rfchnpkofw ks!`"pdz ficke!in!qqctx`pi!dewdnpofmt- fitness: 50
270. generáció legjobb egyede: Agfnt rfchnpkofw ks!`"pdz ficke!in!qqctx`pc!dewdnpofmt- fitness: 48
280. generáció legjobb egyede: Agfnt rfchnpkofw hs!`"pdz ficke!in!qpctx`pd!dewdnpofmt- fitness: 45
290. generáció legjobb egyede: Agfnt rfchnpkofy hs!`"pdz ficke!in qpctx`pd!dewdnpofmt- fitness: 42
300. generáció legjobb egyede: Agfnt rfchnpkofy hs!`"pdw ficke!in!qodtx`pd!dewdnpofmt- fitness: 38
```





```

HOME PLOTS APPS
New Script New Live Script New Open Find Files Import Data Save Workspace New Variable Open Variable Clear Workspace Favorites Analyze Code Run and Time Clear Commands Simulink Layout Set
FILE VARIABLE CODE SIMULINK ENVIR
D:\My backups\My Backups_2019_12_09\a_tantargyak 2010-tol\BLSZM\eloadasok2020\eloadasok\GA
New to MATLAB? See resources for Getting Started.
440. generáció legjobb egyede: Agent technokogy hs ^ mdw filele in softx`re!dewdlopmpfnt- fitness: 13
450. generáció legjobb egyede: Agent technokogy hs ^ mdw filele in softx`re!dewdlopment- fitness: 12
460. generáció legjobb egyede: Agent technokogy hs ^ mdw filele in softw`re!dewdlopmpfnt- fitness: 12
470. generáció legjobb egyede: Agent technokogy hs ^ mdw filele in softxare!dewdlopmpfnt. fitness: 11
480. generáció legjobb egyede: Agent technokogy hs ^ mew filele in softxare!dewdlopmpfnt. fitness: 10
490. generáció legjobb egyede: Agent technology hs ^ mew filele in softxare!devdlopmpfnt. fitness: 8
500. generáció legjobb egyede: Agent technology hs ^ mew filele in software!devdlopment. fitness: 6
510. generáció legjobb egyede: Agent technology is ^ mew filele in software!devdlopment. fitness: 5
520. generáció legjobb egyede: Agent technology is ^ mew field in software!devdlopment. fitness: 4
530. generáció legjobb egyede: Agent technology is ^ mew field in software!devdlopment. fitness: 4
540. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
550. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
560. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
570. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
580. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
590. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
600. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
610. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
620. generáció legjobb egyede: Agent technology is ^ mew field in software devdlopment. fitness: 3
630. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
640. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
650. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
660. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
670. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
680. generáció legjobb egyede: Agent technology is a mew field in software devdlopment. fitness: 2
690. generáció legjobb egyede: Agent technology is a new field in software development. fitness: 0
Optimális érték elérve.

ans =

'Agent technology is a new field in software development.'

```