Bevezetés a lágy számítás módszereibe
Genetikus algoritmusok
Evolúció, genetika, kódolás, szelekció

Werner Ágnes

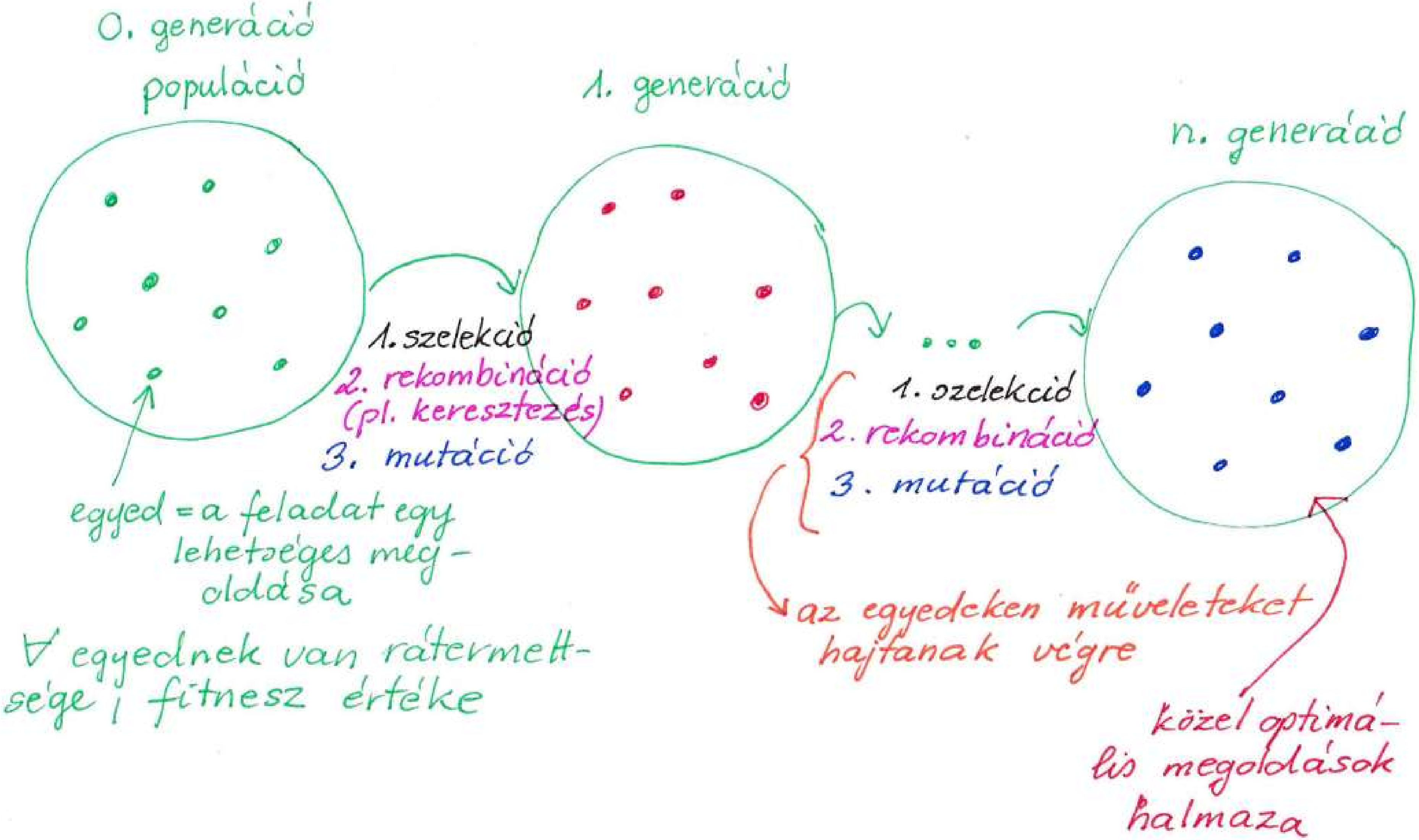
Villamosmérnöki és Információs Rendszerek Tanszék

Evolúció

- darwini evolúció, tőrzsfejlődés ismerete
- egyedek közötti versengés
  - a táplálékszerzés ügyessége
  - a táplálékká válás elkerülésének képessége
  - fajon belüli harc a nőstényekért
  - tűrőképesség mértéke
  - alkalmazkodás képessége
- minden irányítottság nélkül

Genetika

- azonos fajhoz tartozó élőlények nem egyformák
- tökéletesebb utódok
- "két fekete kecskének tarka utódja"
- gének
- szaporodás
- mutáció


0. generáció
populáció
1. generáció
n. generáció
1. szelekció
2. rekombináció
(pl. keresztezés)
3. mutáció
1. szelekció
2. rekombináció
3. mutáció
...
egyed = a feladat egy lehetséges meg-oldása
∀ egyednek van rátermett-sége / fitnesz értéke
az egyedeken műveleteket hajtanak végre
közel optimá-lis megoldások halmaza

1. Gének

- az öröklött tulajdonságokat a gének határozzák meg
- két fontos jellemző:
  1. funkció
  2. lókusz (hely)
- allélok
- kromoszóma
- kromoszómaszelvény
- a gén egyik allélje domináns
- fenotípus
- genotípus

2. Szaporodás

- ivartalan (utódnak egy szülője van, megegyezik a szülővel)
- ivaros (utódnak két szülője van, genetikai anyag keveredik)

3. Mutáció

- megváltozhat egy gén értéke
- kromoszóma részek maradhatnak el
- kromoszóma részek kettőződhetnek meg
- kromoszóma részek fordulhatnak meg
- a mutációt lehet pozitív értelemben is tekinteni –> elősegíti a genetikai változatosságot

Általános evolúciós algoritmus pszeudó-kódja

- $t := 0$ {kezdeti idő beállítása}
- $initpopulacio \;\; P_t$ {kezdeti populáció létrehozása}
- $fitneszszamit \;\; P_t$ {fitneszértékek kiszámítása}
- **while** amíg nincs kész **do**
- $P_t' := szulokivalasztas \;\; P_t$ {szülők választása}
- $kereszetez \;\; P_t'$ {a szülők génjeinek keresztezése}
- $mutacio \;\; P_t'$ {véletlen mutáció}
- $fitneszszamit \;\; P_t'$ {az új fitnesz kiszámítása}
- $P_{t+1} := tulelo(P_t, P_t')$ {az új populációba kerülnek az egyedek}
- $t := t + 1$
- **end while**

Az algoritmus konvergál.

Evolúciós algoritmus alaptechnikák

1. Hogyan ábrázolható az egyed
2. Milyen rekombinációs és mutációs műveletet alkalmazzunk
3. Milyen szelekciós és visszahelyező művelet jöhet szóba
4. Fitnesz függvény definiálása
5. Egyes feladatoknál a lehetséges megoldásokat feltételekkel korlátozzuk (pl. büntető függvény)
6. Általános paraméterek, az egyes műveletek paramétereinek meghatározása

Az egyedek ábrázolása, kódolása

- **Valós (egész) vektor**
  Fenotípus formát jelent
  Az egyed tulajdonságait mint valós (egész) értékű változókat adjuk meg és az egyedeket vektorként ábrázoljuk:
  $E = (x_1, x_2, \ldots, x_n)$, ahol x_i az i-edik tulajdonsághoz tartozó változó.
- **Permutáció**
  Adott számú objektumot valamilyen sorrendben el kell helyezni
  Permutáció: $(\pi_1, \pi_2, \ldots, \pi_n)$ az első n pozitív egész szám permutációja
  A sorrend rögzített
  Más ábrázolási forma:
  $\pi_1, \pi_2, \ldots, \pi_n$ sorrend
  $x_1, x_2, \ldots, x_n$ pozícióhoz rendelt objektum
  Fenotípus forma

Az egyedek ábrázolási formája

- **Bináris vektor**
  Genotípus formát jelent
  Jelölje az $(x_1, x_2, \ldots, x_n)$ valós (egész) vektor az egyed tulajdonságait.
  Bináris ábrázolásban az egyed egy sztringként jelenik meg:

$$x_1, x_2, \ldots, x_i, \ldots, x_n$$

$\ldots 00|11010111|01 \ldots$ az x_i változó kódolt értékei

Bináris kódoló és dekódoló függvény

Standard bináris kódolás

az x_i válzozót kettes számrendszerbeli számmá konvertálja és a kapott bitsorozatot egy véges hosszúságú rész sztringre képezi le;

csak egy $[b_i, c_i]$ intervallumba eső, adott pontosságú számot tud ábrázolni;
a másik irányú transzformációhoz jelölje D a dekódoló függvényt,
h_i az i. rész szring hosszát,
a_{iz} a rész sztring z. pozícióján lévő bit értékét,
az i. rész sztring dekódolt értéke:

$$D(a_{i1}, \cdots, a_{in}) = b_i + \frac{c_i - b_i}{2^{h_i} - 1}\left(\sum_{z=1}^{h_i} a_{i(h_i - z + 1)} \cdot 2^{z-1}\right) = x_i$$

Példa: $[b_i,c_i]=[4,9]$ $h_i=4$ (i. rész sztring hossza)

$$\boldsymbol{D}(\mathbf{1},\mathbf{0},\mathbf{0},\mathbf{1}) = 4 + \frac{9-4}{2^4 - 1} * (a_{i4}2^0 + a_{i3}2^1 + a_{i2}2^2 + a_{i1}2^3) = 4 + \frac{5}{15}(1+0+0+8) = \mathbf{7}$$

Bináris kódoló és dekódoló függvény

Gray-kódolás

A standard kódolásnál az egymás melletti számok Hamming távolsága különböző számoknál más és más. Ez a rekombinációnál hibákat okozhat.

A Gray-kódolás kijavítja a Hamming távolságot: egységesen, bármely egymás melletti szám távolsága 1 lesz.

A standard bináris kódból a Gray-kód:

$$g_z = \begin{cases} a_z & ha \quad z = 1 \\ a_{z-1} \bigoplus a_z & kulonben \end{cases}$$

A Gray-kódból a bináris kód:

$$a_z = \bigoplus_{k=1}^{z} g_k$$

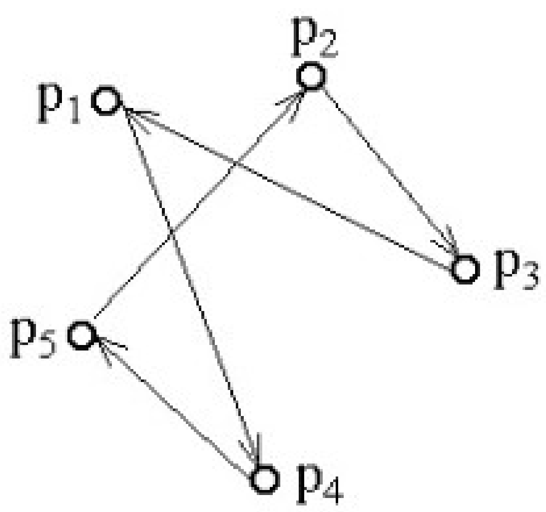
$\bigoplus$ a kettes számrendszerben a modulo 2 összeadást jelöli

Szám	Gray-kód	H. t.	Bináris kód	H.t.
0	0000		0000	
1	0001	1	0001	1
2	0011	1	0010	2
3	0010	1	0011	1
4	0110	1	0100	3
5	0111	1	0101	1
6	0101	1	0110	2
7	0100	1	0111	1
8	1100	1	1000	4
9	1101	1	1001	1

Permutációs ábrázolási mód

Például bejárandó települések megadására, azonosításra:

1	4	5	2	3



Hagyományos órarend reprezentáció

A kétdimenziós mátrix (i, j) eleme azokat a tanárokat tartalmazza, akik a j-edik órában az i-edik osztályban tartanak órát.

	1. nap				…	n. nap			…
	1. óra	2. óra	…	n. óra	…	…	j. óra	…	…
1. oszt	Tóth L.	Biró I.							
2. oszt	Szabó L.	Kovács G.							
⋮									
i. oszt							Kiss I.		
⋮									

Ebben könnyű megállapítani, hogy egy tanár elfoglalt-e egy adott időben vagy ki tart órát egy bizonyos időpontban és osztályban.

Nem támogatja annak ábrázolását, amikor több osztálynak egyidejűleg több tanár tart órát. Például bontott nyelvóráknál 2 osztályt 4 tanár is taníthat.

Halmazos reprezentáció

A legkisebb **adategység a halmaz**, egy olyan struktúra, amely tetszőleges számú osztályt, tanárt és tantermet tartalmaz.
Pl. két osztályhoz egy tanárt rendelünk (például összevont testnevelés óra esetén) a két osztályt és az egy tanárt felvesszük a halmazba.
Csak **egy dimenzió**ban dolgozunk, amely nem más mint az idő.
Az időtengelyen lévő időrésekbe, melyek a lehetséges órákat jelentik-kell halmazainkat elhelyezni.
Egy időrésbe több halmaz is kerülhet, itt ügyelni kell arra, hogy ne legyen ütközés, ne kerüljön egy időrésbe két olyan halmaz, melyben közös tanár vagy közös osztály szerepel.

Halmazos reprezentáció

Az $o_{i,j}$, $t_{i,j}$, $te_{i,j}$ értékekkel az i-edik halmazban szereplő osztályok, tanárok, termek sorszámát jelöljük, j pedig a halmazon belüli sorszámok indexe.

1. nap			2. nap			…
1. óra	2. óra	…	…	…	…	…
$Halmaz_1$: $\text{osztály}_{o_{1,1}}, \text{osztály}_{o_{1,2}}, \ldots$ $\text{tanár}_{t_{1,1}}, \text{tanár}_{t_{1,2}}, \ldots$ $\text{terem}_{te_{1,1}}, \text{terem}_{te_{1,2}}, \ldots$	$Halmaz_{N+1}$: $\text{osztály}_{o_{N+1,1}}, \text{osztály}_{o_{N+1,2}}, \ldots$ $\text{tanár}_{t_{N+1,1}}, \text{tanár}_{t_{N+1,2}}, \ldots$ $\text{terem}_{te_{N+1,1}}, \text{terem}_{te_{N+1,2}}, \ldots$					
$Halmaz_2$	$Halmaz_{N+2}$					
…	…	…				
$Halmaz_N$	$Halmaz_{2N}$					

Halmazos reprezentáció

Függőleges és vízszintes linearizálás

1. időrés	2. időrés	...	k. időrés
Halmaz 1	Halmaz N+1		Halmaz (k-1)*N+1
Halmaz 2	Halmaz N+2		Halmaz (k-1*N+2
...	...		
Halmaz N	Halmaz 2N		Halmaz k*N

Függőleges:

Halmaz 1	Halmaz 2	...	Halmaz N	Halmaz N+1	...	Halmaz k*N

Vízszintes:

Halmaz 1	Halmaz N+1	...	Halmaz (k-1)*N+1	Halmaz 2	...	Halmaz k*N

Szelekció

A populáció átlagos minőségét hivatott javítani. A minőséget a fitnesz függvénnyel mérjük. A jobb minőségű egyedeket nagyobb valószínűséggel használja a GA az új populácó kialakításához.
A szelekciós műveletek összehasonlítása:

- **szelekciós intenzitás**: a szelekció hatására bekövetkező, a populációk átlagos fitnesz értékeinek változását mutatja
$Int = (M^* - M)/\sigma$ ahol M^* és M a szelekció előtti és utáni átlagos fitnesz értékek, σ a fitnesz értékek szórását jelöli az új populációban
- **változatosság elvesztése**: a populáció azon egyedeinek D aránya, amelyeket nem választott ki a szelekciós művelet
- **szelekciós variancia**: a populációbeli fitnesz értékek varianciájának változása a szelekció hatására
$V = (\sigma^*)^2/\sigma^2$ σ és σ^* a fitnesz értékek szórása a szelekció előtt és után

Rulett szelekció

- Fitnesz arányos szelekció, amely az egyedeket fitnesz értékük abszolut értékének arányában választja ki a szelekciós állományból.
- Visszatevéses művelet
- Egy egyed kiválasztását a szelekciós valószínűség határozza meg:

$$P(E_i) = \frac{f(E_i)}{\sum_{j=1}^{n} f(E_j)}$$

f a fitnesz függvény, E_i $(i = 1, \ldots, n)$ az egyedek

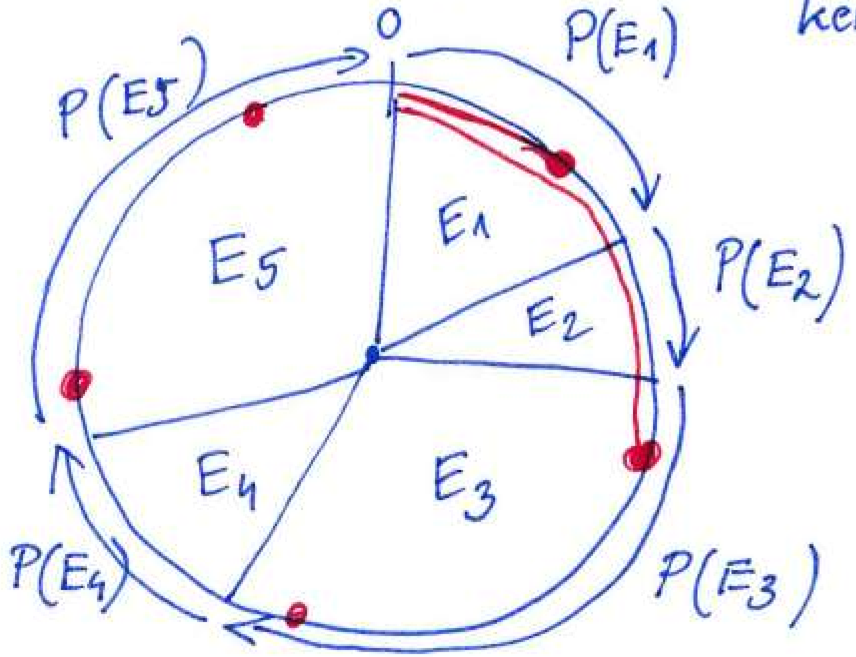
Rulett szelekció

- veszünk egy rulettet
- feleltessünk meg minden E_i egyednek valamely kiindulási pontból folyamatosan egy-egy körszeletet
- generálunk egy $[0, 1]$-beli véletlen számot, a véletlen számot ívhossznak tekintjük
- azt az egyedet választjuk, amelynek körszeletében az ív végződik
- egy μ elemű szelekciós halmazból a választást μ-ször kell megismételni, amíg kialakul a szülők állománya

A kiválasztott egyedek közt $\mu * p(E_i)$ $(i = 1, \ldots, n)$ várható számú másolata lesz az E_i egyednek.

Rulett kerék

kerület = 1



A körszelet ívének hossza feleljen meg az egyed szelekciós valószínűségének.

Sztochasztikus univerzális mintavétel

- Fitnesz arányos szelekció
- Minimalizálja a szelekció során kapott duplikációk számát
- Minden E_i egyedhez a várható másolatok számával azonos hosszúságú körívet rendel:

  $V(E_i) = \mu * p(E_i)$

Sztochasztikus univerzális mintavétel

Lépések:

1. **Input**: A szelekciós állomány E_i elemei és a hozzá tartozó $V(E_i)$ $(i = 1, \ldots, n)$ várható másolatok száma.
2. **Output**: A populáció a szelekció után (szülők állománya): E_i' $(i = 1, \ldots, n)$
3. $s = 0; \quad j = 1$
4. $mutato = Rnd$ (véletlen szám a $[0, 1]$ intervallumból)
5. **for** $i = 1$ **to** μ **do**
6. $\quad s = s + V(E_i)$
7. $\quad$ **while** $(s > mutato)$ **do**
8. $\quad\quad E_j' = E_i; \quad j = j + 1; \quad mutato = mutato + 2r\pi/\mu$
9. $\quad\quad$ **od**
10. **od**

A kiválasztott egyedek közt $\mu * p(E_i)$ $(i = 1, \ldots, n)$ várható számú másolata lesz az E_i egyednek.

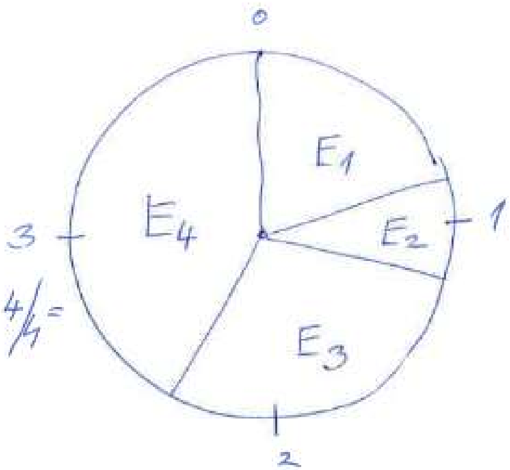
	E_1	E_2	E_3	E_4
$f(E_i)$	2	1	3	4
$p(E_i)$	0,2	0,1	0,3	0,4
$v(E_i)$	0,8	0,4	1,2	1,6

$$\sum_{j=1}^{4} f(E_j) = 10$$

$\mu = 4$

$$P(E_i) = \frac{f(E_i)}{\sum_{j=1}^{4} f(E_j)}$$

$$V(E_1) + V(E_2) + V(E_3) + V(E_4) =$$
$$= 2r\pi = 4$$



mutató $= 0,4$ $\quad s := 0 \quad j := 1$

$\boxed{i=1}$ $\quad s := s + 0,8 = 0,8$

$V(E_1) \overset{?}{>} 0,4$ ✓

$ = 0,8$

$\boxed{E_1' := \boxed{E_1}}$, $\quad j := 2 \quad$ mutató $= 0,4 + 4/4 =$
$= 1,4$

$0,8 \overset{?}{>} 1,4$ ✗

$\boxed{i=2}$

$s := 0,8 + 0,4 = 1,2$

$1,2 \overset{?}{>} 1,4$ ✗

$\boxed{i=3}$ $\quad s := 1,2 + 1,2 = 2,4$

$2,4 \overset{?}{>} 1,4$ ✓

$\boxed{E_2' := \boxed{E_3}}$ $\quad j := 3 \quad$ mutató $= 1,4 + 1 =$
$= 2,4$

$2,4 \overset{?}{>} 2,4$ ✗

$\boxed{i=4}$

$s := 2,4 + 1,6 = 4$

$4 \overset{?}{>} 2,4$

$\boxed{E_3' := \boxed{E_4}}$ $\quad j := 4$

mutató $:= 2,4 + 1 = 3,4$

$4 \overset{?}{>} 3,4$ ✓

$\boxed{E_4' := \boxed{E_4}}$. $\quad j := 5$

mutató $:= 4,4$

$4 \overset{?}{>} 4,4$ ✗

Versengő szelekció

- Az egyedek fitnesz értékeinek sorrendjét használja fel.
- Nem fog növekedni az egyed duplikációk száma.
- Több egyedből a legjobb fitnesz értékű egyedet választja ki. (Biológiai szelekciót modellezi.)

Versengő szelekció

Lépések:

1. **Input**: A szelekciós állomány E_i elemei és $f(E_i)$ fitnesz értékei $(i = 1, \ldots, n)$, $tour$ paraméter
2. **Output**: A populáció a szelekció után (szülők állománya): $E_i' \quad (i = 1, \ldots, n)$
3. **for** $i = 1$ **to** μ **do**
4.   **for** $k = 1$ **to** $tour$ **do**
5.   válasszunk egy $j \in \{1, \ldots, n\}$ indexet véletlenszerűen
6. $T_k = E_j$
7.   **od**
8. $E_i' = T_j$ ha $f(T_j) = \max(f(T_1), \ldots, f(T_{tour}))$
9. **od**

A kiválasztott egyedek közt $\mu * p(E_i)$ $(i = 1, \ldots, n)$ várható számú másolata lesz az E_i egyednek.

Versengő szelekció (valós életből származik)

tour = 3

	E_1	E_2	E_3	E_4	E_5	E_6
értéke:	2	7	4	8	6	9
fitnesz:	0,2	0,4	0,8	0,5	0,7	0,1

$i = 1$

$k=1 \quad j=3 \quad T_1 = E_3 = 4 \quad \boxed{0{,}8}$

$k=2 \quad j=5 \quad T_2 = E_5 = 6 \quad 0{,}7$

$k=3 \quad j=1 \quad T_3 = E_1 = 2 \quad 0{,}2$

→ max

$E_1' = T_1 = E_3 = 4$

véletlenszerűen választjuk

$i = 2$

$k=1 \quad j=2 \quad T_1 = E_2 = 7 \quad 0{,}4$

$k=2 \quad j=5 \quad T_2 = E_5 = 6 \quad 0{,}7$

$k=3 \quad j=3 \quad T_3 = E_3 = 4 \quad \boxed{0{,}8}$

→ max

$E_2' = T_3 = E_3 = 4$

⋮

$i = 6$ -ig

Csonkolásos szelekció

- Csak a legjobb egyedeket választjuk ki.
- A fitnesz értékek sorrendjét használja fel. (Mesterséges eljárás.)
- Lépések:

1. **Input**: A szelekciós állomány E_i elemei és $f(E_i)$ fitnesz értékei $(i = 1, \ldots, n)$, a $T \in [0, 1]$ korlát
2. **Output**: A populáció a szelekció után (szülők állománya): E'_i $(i = 1, \ldots, n)$
3. Legyen J a fitnesz értékek alapján növekvőbe rendezett szelekciós halmaz.
4. **for** $i = 1$ **to** μ **do**
5. válasszunk egy $k \in \{[(1 - T) * \mu], \ldots, \mu\}$ indexet véletlenszerűen
6. $E'_i = J_k$
7. **od**

Csonkolásos szelekció

Szelekciós halmaz:

$$
\begin{array}{lcccccc}
 & E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \\
f(E_i) & 0{,}2 & 0{,}4 & 0{,}8 & 0{,}5 & 0{,}7 & 0{,}1
\end{array}
$$

$\mu := 6 \quad T := 0{,}4$

$k \in \{[0{,}6 \cdot 6], \ldots, 6\} = \{3, 4, 5, 6\}$

$$
\mathcal{J} = (\underset{0{,}1}{E_6}, \underset{0{,}2}{E_1}, \underset{0{,}4}{E_2}, \underset{0{,}5}{E_4}, \underset{0{,}7}{E_5}, \underset{0{,}8}{E_3})
$$

Szülők állománya:

$E_1' = \mathcal{J}_4 = \boxed{E_4} \quad f = 0{,}5$

$E_2' = \mathcal{J}_3 = \boxed{E_2} \quad f = 0{,}4$

$E_3' = \mathcal{J}_5 = \boxed{E_5} \quad f = 0{,}7$

$E_4' = \mathcal{J}_4 = \boxed{E_4} \quad f = 0{,}5$

$E_5' = \mathcal{J}_5 = \boxed{E_5} \quad f = 0{,}7$

$E_6' = \mathcal{J}_6 = \boxed{E_3} \quad f = 0{,}8$

Lineáris sorrend alapú szelekció

- Minden egyedet a fitnesz értékeik alapján sorba rendezünk, és sorszámot rendelünk hozzájuk (1 a legrosszabb egyed sorszáma).
- Az egyedek kiválasztását a szelekciós valószínűség határozza meg, amely lineárisan függ az egyed sorszámától.
- $P_1 = \eta^- / \mu$ a legrosszabb egyed szelekciós valószínűsége $\eta^- \in [0, 1]$
- μ egyedet kell választani
- $P_\mu = (2 - \eta^-)/\mu$
- η^- / μ és η^+ / μ a legrosszabb és a legjobb egyed kiválasztásának valószínűsége $\eta^+ = 2 - \eta^-$

Lineáris sorrend alapú szelekció

Lépések:

1. **Input**: A szelekciós állomány E_i elemei és $f(E_i)$ fitnesz értékei $(i = 1, \ldots, n)$, $\eta^- \in [0, 1]$
2. **Output**: A populáció a szelekció után (szülők állománya): E_i' $(i = 1, \ldots, n)$
3. Legyen J a fitnesz értékek alapján növekvőbe rendezett szelekciós halmaz.
4. $S_0 = 0$
5. **for** $i = 1$ **to** μ **do**
6. $\quad S_i = S_{i-1} + p_i$
7. **od**
8. **for** $i = 1$ **to** μ **do**
9. $\quad r = Rnd;\ \ E_i' = J_z,$ ha $S_{z-1} \leq r < S_z$
10. **od**

Lineáris sorrend alapú szelekció

	E_1	E_2	E_3	E_4
$f(E_i)$	0,2	0,5	0,3	0,1
	$\mathcal{J}_1$	$\mathcal{J}_2$	$\mathcal{J}_3$	$\mathcal{J}_4$
	E_4	E_1	E_3	E_2
	↑	↑ 0,2	↑ 0,3	↑

$\mu = 4$

$\eta^- = 0{,}4$ $\quad (\eta^- \in [0,1])$

$p_1 = \eta^-/\mu$ $\quad \eta^+ = 2 - \eta^-$

$p_4 = \eta^+/\mu$

$p_1 = 0{,}4/4 = 0{,}1 \qquad p_4 = 1{,}6/4 = 0{,}4$

$s_0 = 0$ $\quad \boxed{i=1}$ $\quad s_1 = s_0 + p_1 = 0{,}1$ $\quad \boxed{i=2}$ $\quad s_2 = s_1 + p_2 = 0{,}1 + 0{,}2 = 0{,}3$

$\boxed{i=3}$ $\quad s_3 = s_2 + p_3 = 0{,}3 + 0{,}3 = 0{,}6$ $\quad \boxed{i=4}$ $\quad s_4 = s_3 + p_4 = 0{,}6 + 0{,}4 = 1$

$\boxed{i=1}$ $\quad r = 0{,}8 \quad E_1' = \mathcal{J}_4 = \boxed{E_2} \Leftarrow$ $\quad s_{z-1} \le r < s_z \Rightarrow (z=4)$

$0{,}6 \le 0{,}8 < 1$

$\boxed{i=2}$ $\quad r = 0{,}3 \quad E_2' = \mathcal{J}_3 = \boxed{E_3} \Leftarrow \quad 0{,}3 \le 0{,}3 < 0{,}6 \Rightarrow z = 3$

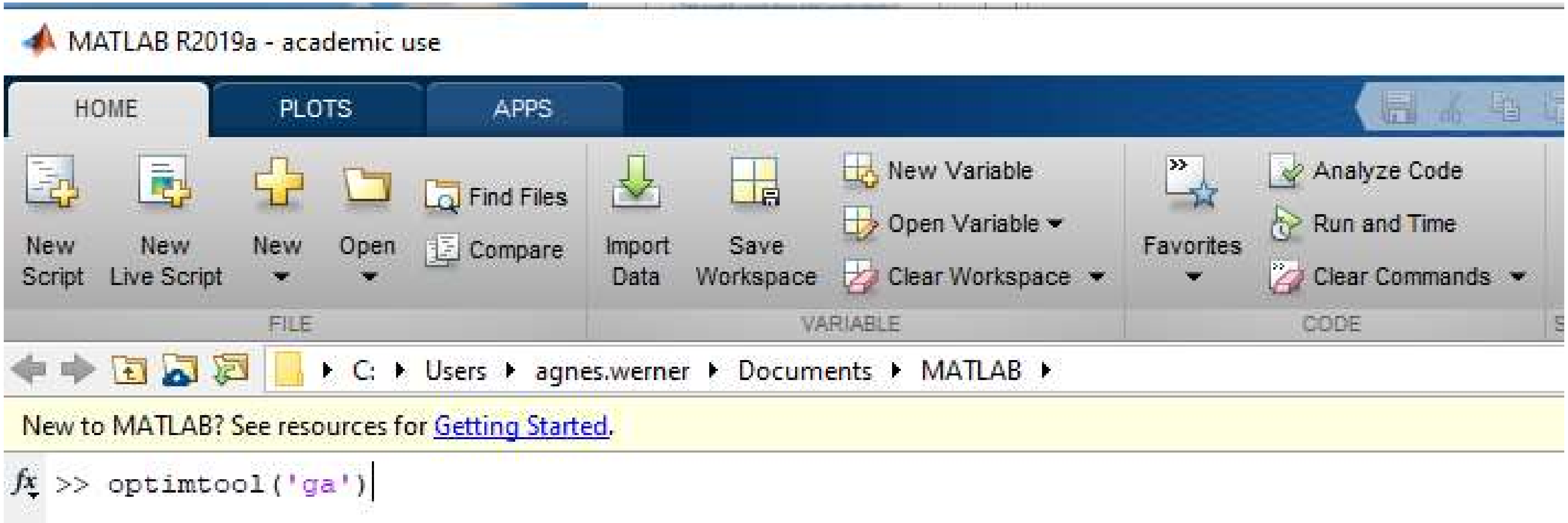
$\boxed{i=3}$ $\quad r = 0{,}4 \quad E_3' = \mathcal{J}_3 = \boxed{E_3} \Leftarrow \quad 0{,}3 \le 0{,}4 < 0{,}6 \Rightarrow z = 3$

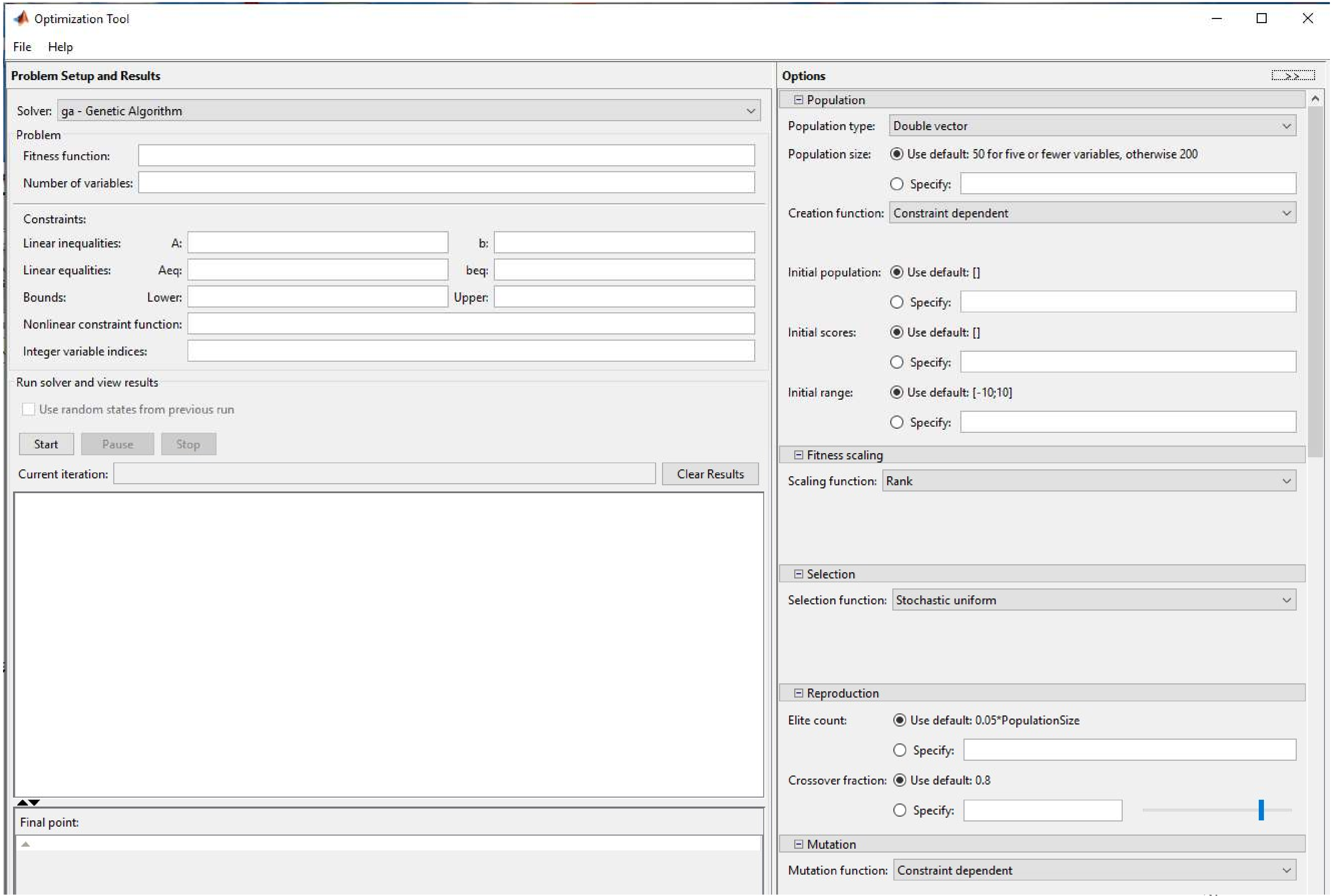
$\boxed{i=4}$ $\quad r = 0{,}7 \quad E_4' = \mathcal{J}_4 = \boxed{E_2} \Leftarrow \quad 0{,}6 \le 0{,}7 < 1 \Rightarrow z = 4$

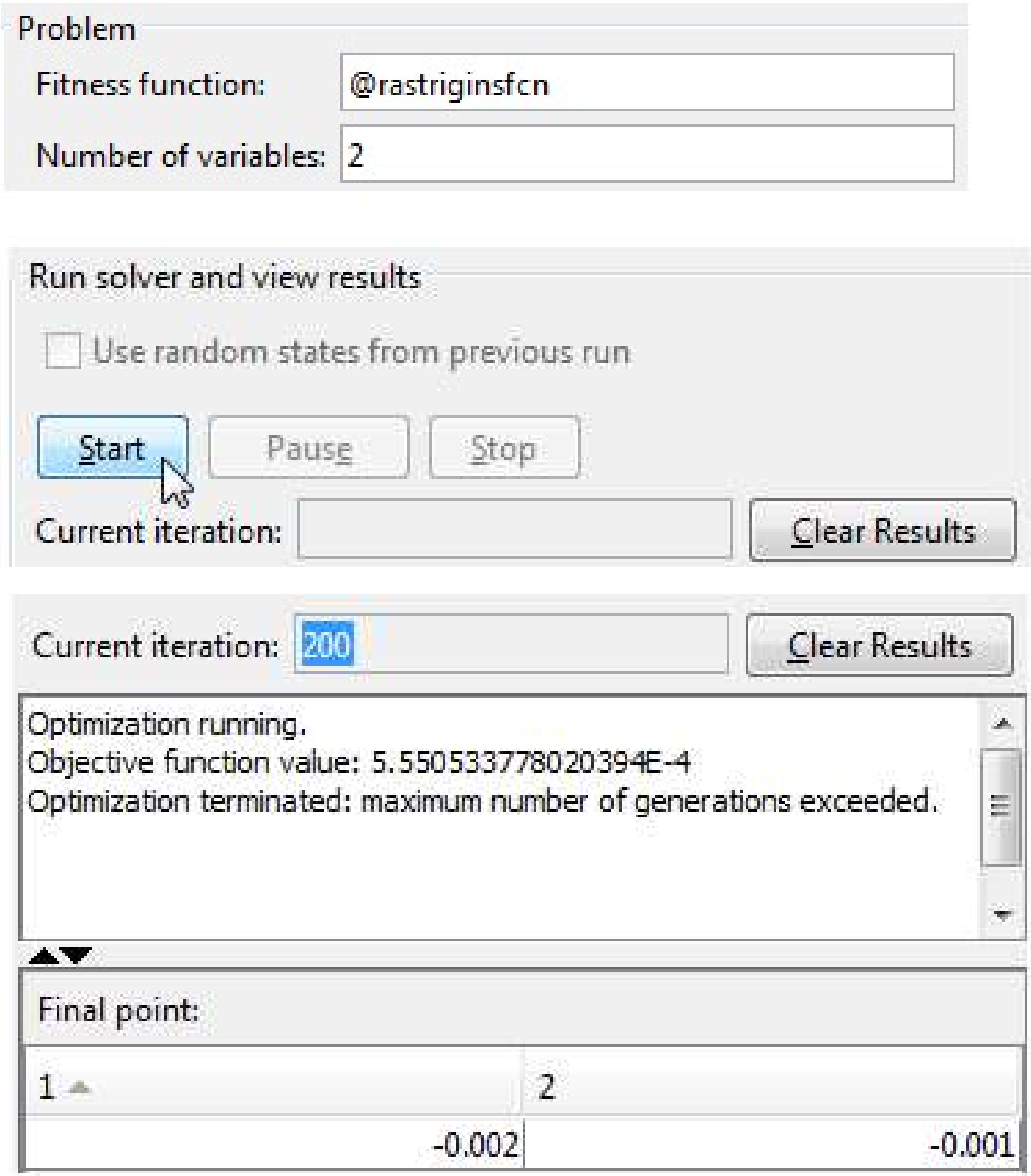
MATLAB

Global Optimization Toolbox

Használjuk az optimtool('ga'):




Optimization Tool
File
Help
Problem Setup and Results
Solver:
ga - Genetic Algorithm
Problem
Fitness function:
Number of variables:
Constraints:
Linear inequalities:
A:
b:
Linear equalities:
Aeq:
beq:
Bounds:
Lower:
Upper:
Nonlinear constraint function:
Integer variable indices:
Run solver and view results
Use random states from previous run
Start
Pause
Stop
Current iteration:
Clear Results
Final point:
Options
Population
Population type:
Double vector
Population size:
Use default: 50 for five or fewer variables, otherwise 200
Specify:
Creation function:
Constraint dependent
Initial population:
Use default: []
Specify:
Initial scores:
Use default: []
Specify:
Initial range:
Use default: [-10;10]
Specify:
Fitness scaling
Scaling function:
Rank
Selection
Selection function:
Stochastic uniform
Reproduction
Elite count:
Use default: 0.05*PopulationSize
Specify:
Crossover fraction:
Use default: 0.8
Specify:
Mutation
Mutation function:
Constraint dependent


Problem
Fitness function: @rastriginsfcn
Number of variables: 2
Run solver and view results
Use random states from previous run
Start
Pause
Stop
Current iteration:
Clear Results
Current iteration: 200
Clear Results
Optimization running.
Objective function value: 5.550533778020394E-4
Optimization terminated: maximum number of generations exceeded.
Final point:
1
2
-0.002
-0.001