

GENETSKI ALGOTIRMI

Genetski algoritmi, kao jedni od evolucijskih algoritama, heuristička su metoda optimiranja koja imitira prirodni evolucijski proces. Predstavljaju model strojnog učenja čije ponašanje potječe iz procesa evolucije koji se neprekidno odvija u prirodi. Kao algoritmi koji se oslanjaju na ideje iz prirode, pogodni su za pretraživanje velikog prostora mogućnosti s ciljem pronalaska optimalnog rješenja.

Osnovna karakteristika evolucije u prirodi je prilagođavanje živih bića uvjetima u prirodi. Analogija evolucije kao prirodnog procesa i genetskog algoritma kao metode optimiranja očituje se u procesu selekcije i genetskim operatorima. Selekcijom se odabiru dobre jedinke (rješenja) koje se prenose u sljedeću populaciju, a manipulacijom genetskog materijala stvaraju se nove jedinke. Takav ciklus reprodukcije, selekcije i manipulacije ponavlja se sve dok nije zadovoljen uvjet zaustavljanja genetskog algoritma.

Po načinu djelovanja, genetski algoritmi ubrajaju se u metode slučajnog pretraživanja prostora rješenja u potrazi za globalnim optimumom. U tu skupinu možemo ubrojiti još i evolucijske strategije, simulirano kaljenje te genetsko programiranje. Snaga tih metoda, a ponajviše genetskih algoritama, leži u tome što su u stanju odrediti položaj globalnog optimuma u prostoru s više lokalnih ekstrema, u tzv. višedimenzionalnom prostoru.

1. Pozivanje funkcija za genetske algoritme iz linije naredbi (engl. command line)

Za pokretanje genetskih algoritama iz linije naredbi treba pozvati funkciju za genetske algoritme sa sintaksom:

```
[x fval] = ga(@fitnessfun, nvars, options),
```

gdje je:

- **@fitnessfun** – handle za funkciju cilja,
- **nvars** – broj nezavisnih ulaznih varijabli u funkciju cilja, a
- **options** – struktura koja sadrži postavke za GA. Ako se ovaj argument izostavi, koriste se pretpostavljene postavke.

S ciljem dobivanja detaljnijih informacija, možete koristiti sljedeću sintaksu:

```
x fval exitflag output population scores]=ga(@fitnessfcn, nvars, options)
```

pri čemu je:

- **x** – trenutak kada je dostignuta konačna vrijednost funkcije cilja,
- **fval** – konačna vrijednost funkcije cilja,
- **exitflag** – opisuje razlog zašto je GA prekinut,

- **output** – struktura sadrži informacije o obavljanju GA u svakoj generaciji,
- **population** – konačna populacija,
- **scores** – konačna rješenja.

1.1. Pokretanje genetskih algoritama s pretpostavljenim postavkama

Za pokretanje genetskih algoritama s pretpostavljenim mogućnostima, funkciju za genetske algoritme možemo pozvati u kraćem obliku, tj.

```
[x fval] = ga(@fitnessfun, nvars) .
```

Prije korištenja strukture **options** kao argumenta funkcije **ga**, strukturu treba stvoriti. To se radi s naredbom:

```
options = gaoptimset(@ga) .
```

Postavke crtanja

Postavke crtanja omogućuju vam crtanje podataka iz genetskog algoritma za vrijeme njegova izvođenja. Kada izaberete funkciju crtanja i pokrenete genetski algoritam, u prozoru za crtanje prikazuje se graf s dvije osi. Klikom na subplot otvara se veća inačica grafa u novom prozoru. Pritiskom na gumb **stop** u prozoru crtanja možete prekinuti izvođenje algoritma.

PlotInterval određuje svakih koliko generacija se vrši ispis.

Postavka **OutputFcn** prihvaća polje koje se sastoji od 0, 1 ili svih sljedećih funkcija:

@gplotbestf crta najbolje vrijednosti po generacijama,

@gplotexpectation crta očekivani broj djece s obzirom na rezultat pojedine generacije,

@gplotscorediversity crta histogram rezultata pojedine generacije,

@plotstopping crta kriterije zaustavljanja,

@gplotbestindiv crta karakteristike najboljih jedinki u generacijama,

@gplotgenealogy crte genealogiju jedinki. Crte iz jedne generacije u drugu su označene različitim bojama s različitim značenjima:

- Crvene crte označavaju djecu nastalu mutacijom, Plave crte označavaju djecu nastalu križanjem, Crne crte označavaju elitne jedinke,

@gplotscores crta rezultate jedinki u pojedinim generacijama,

@gplotmaxconstr crta najveće nelinearno prekoračenje ograničenja u pojedinoj generaciji,

@gplotdistance crta prosječnu udaljenost jedinki u pojedinoj generaciji,

@gplotrange crta minimum, maksimum i prosječnu vrijednost funkcije cilja u pojedinim generacijama,

@gplotselection crta histogram roditelja.

Možete napisati i vlastitu funkciju za crtanje.

Prvi redak funkcije za crtanje koju pišete mora imati oblik:

```
function state = plotfun(options, state, flag) .
```

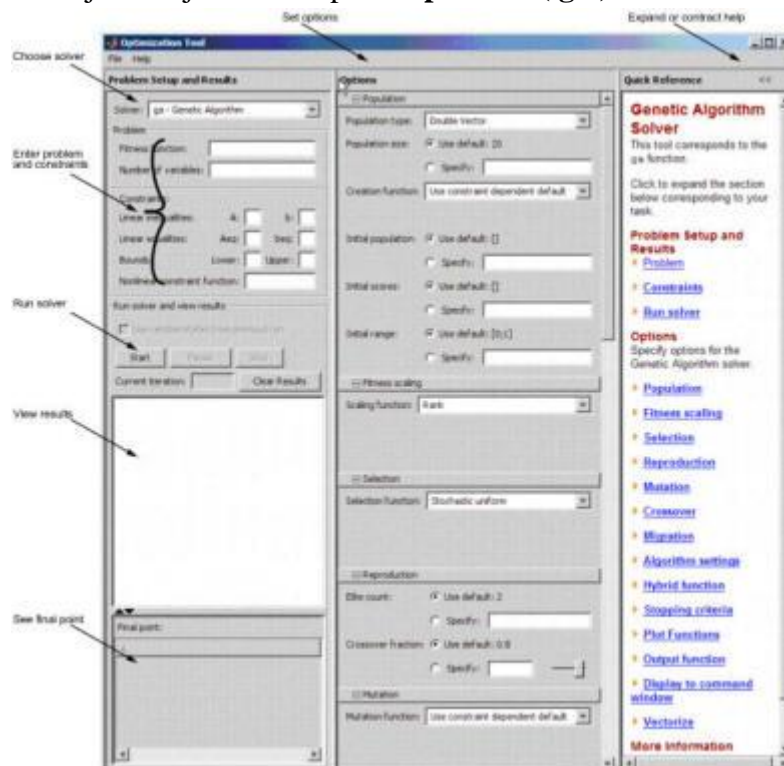
Ulazni argumenti u funkciju su:

- **options** – struktura sadrži sve trenutne postavke
- **state** – struktura sadrži informacije o trenutnoj populaciji
- **flag** – string koji nam govori koji se dio algoritma trenutno izvodi.

2. Korištenje Optimization toola

Optimization toolom možete napraviti sve što možete napraviti i pozivom funkcije `ga` iz linije naredbi. Postavke koje možete postavljati u *Optimization toolu* istovjetne su onima koje se u liniji naredbi postavljaju u strukturi *options*. Za pokretanje

Optimization toola valja u liniji naredbi upisati **`optimtool('ga')`**.



Slika 1: Izgled Optimization toola

Prije početka korištenja Optimization toola trebate unijeti sljedeće podatke:

Funkciju cilja (Fitness function) – funkcija koju želimo minimizirati. Funkciju cilja treba unijeti u formatu `@ime_funkcije_cilja`, gdje je `fitnessfun.m` M-funkcija koja računa funkciju cilja.

Broj varijabli (Number of variables) – broj elemenata ulaznog vektora, tj. broj ulaznih varijabli u funkciju cilja.

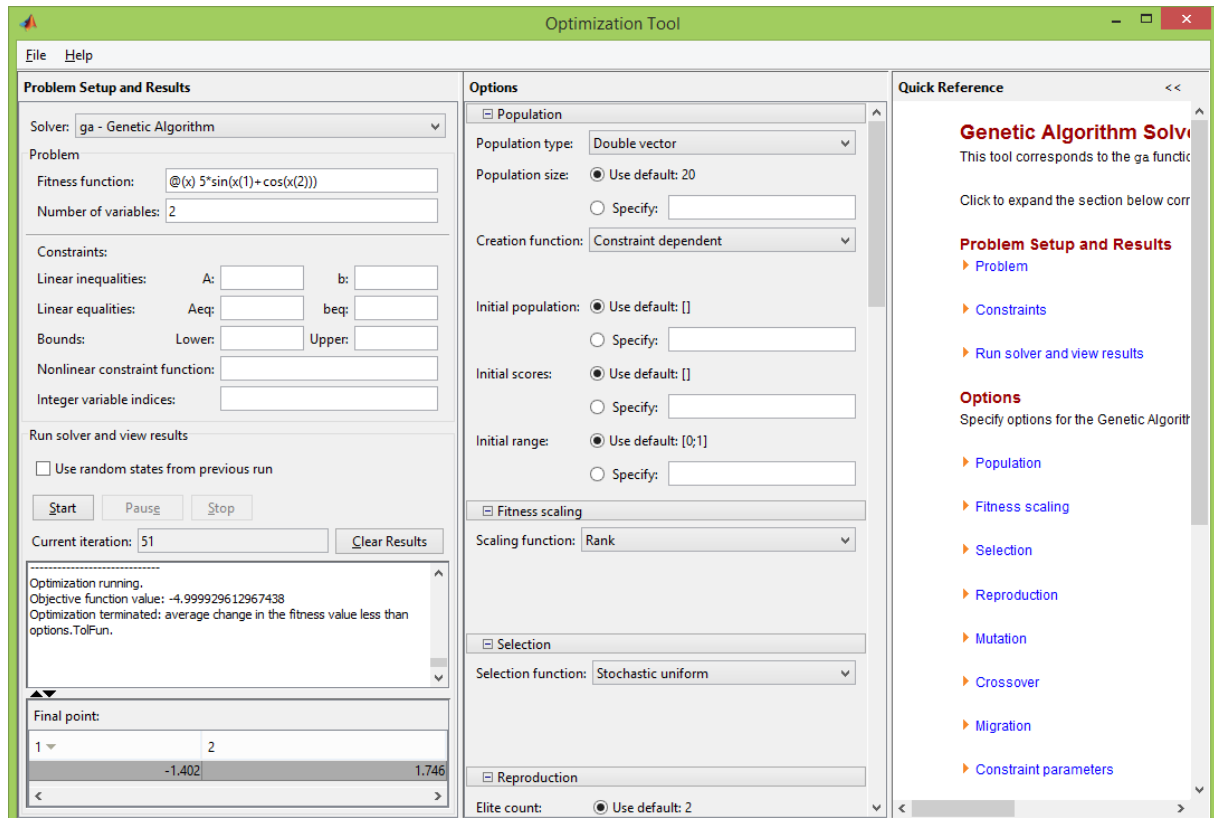
U okviru **Constraints** moguće je unositi ograničenja ili linearnu funkciju ograničenja. Ako problem nije ograničen, polja navedenog okvira mogu se ostaviti prazna. Za pokretanje genetskog algoritma valja pritisnuti gumb **Start**. Rezultati optimizacije ispisuju se u okviru **Run solver and view results**.

U okviru Options možete mijenjati postavke.

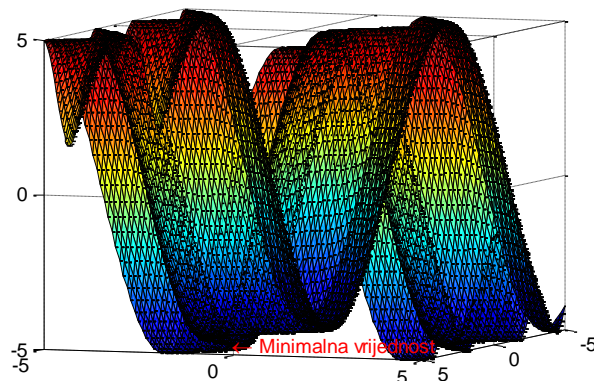
PRIMJER 1: Koristeći genetski algoritam naći minimum sljedeće funkcije:

$$z = 5 \cdot \sin(x + \cos(y))$$

ukoliko je x i y na intervalu od -5 do 5.



```
[x,y]=meshgrid(-5:0.1:5, -5:0.1:5);
z=5*sin(x+cos(y));
surf(x,y,z)
[x,f]=ga(@(x) 5*sin(x(1)+cos(x(2))),2);
text(-1.402, 1.746, -4.999929612967438, '\leftarrow Minimalna vrijednost',
'FontSize',10,'Color','red','LineWidth',3)
r=min(z)
h=min(r)
```

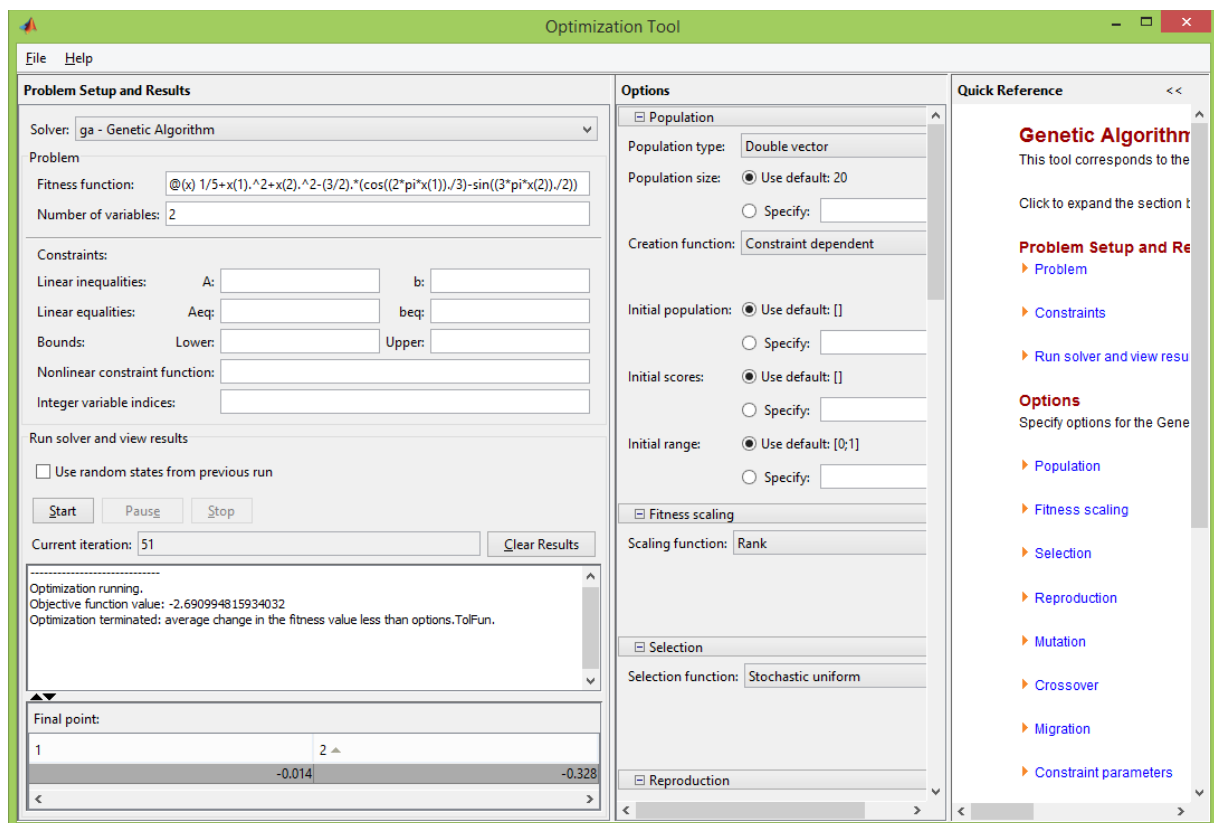


Kod određivanja minimuma ove funkcije je korišten Optimization tool, gdje se za solver bira opcija **ga-Genetic Algorithm**. Zatim se u polje **Fitness function** unosi željena funkcija, a za broj parametara stavljamo 2 (jer su korištene dvije varijable, x i y). Naravno, klikom ga gumb **Start** dobivamo rezultate po x, y i z osi, te upravo njih unosimo kao tačke za ispis teksta. Funkciju crtamo standardno pomoću meshgrid-a i funkcije surf. Sada, ako nam se poklopi da strelica tačno bude na najnižoj vrijednosti (minimumu), tj. tamo gdje je graf tamno-plav, onda smo dobili dobar rezultat. Dakle, pokredemo Start u Optimization tool-u sve dok nam se ne poklope rezultati.

PRIMJER 2: Koristeći genetski algoritam naći minimum sljedeće funkcije:

$$z = \frac{1}{5} + x^2 + y^2 - \frac{3}{2} \cdot \left(\cos\left(\frac{2\pi}{3}x\right) - \sin\left(\frac{3\pi}{2}y\right) \right)$$

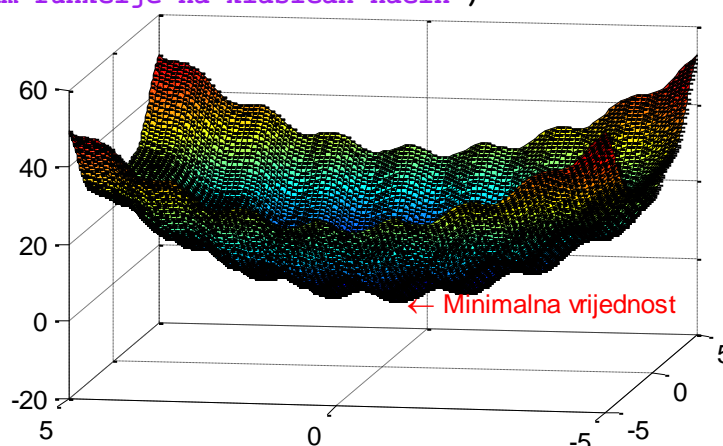
ukoliko je x i y na intervalu od -5 do 5.



```
[x,y]=meshgrid(-5:0.1: 5,-5:0.1: 5);
z=1/5+x.^2+y.^2-(3/2).*(cos((2*pi*x)./3)-sin((3*pi*y)./2))
surf(x,y,z)
```

```
[x,f,m,n,k,v] = ga(@(x) 1/5+x(1).^2+x(2).^2-(3/2).*(cos((2*pi*x(1))./3)-
sin((3*pi*x(2))./2)),2)
text(-0.014 , -0.328, -2.69099, ' \leftarrow Minimalna
vrijednost', 'FontSize',10, 'Color', 'red', 'LineWidth',4)
```

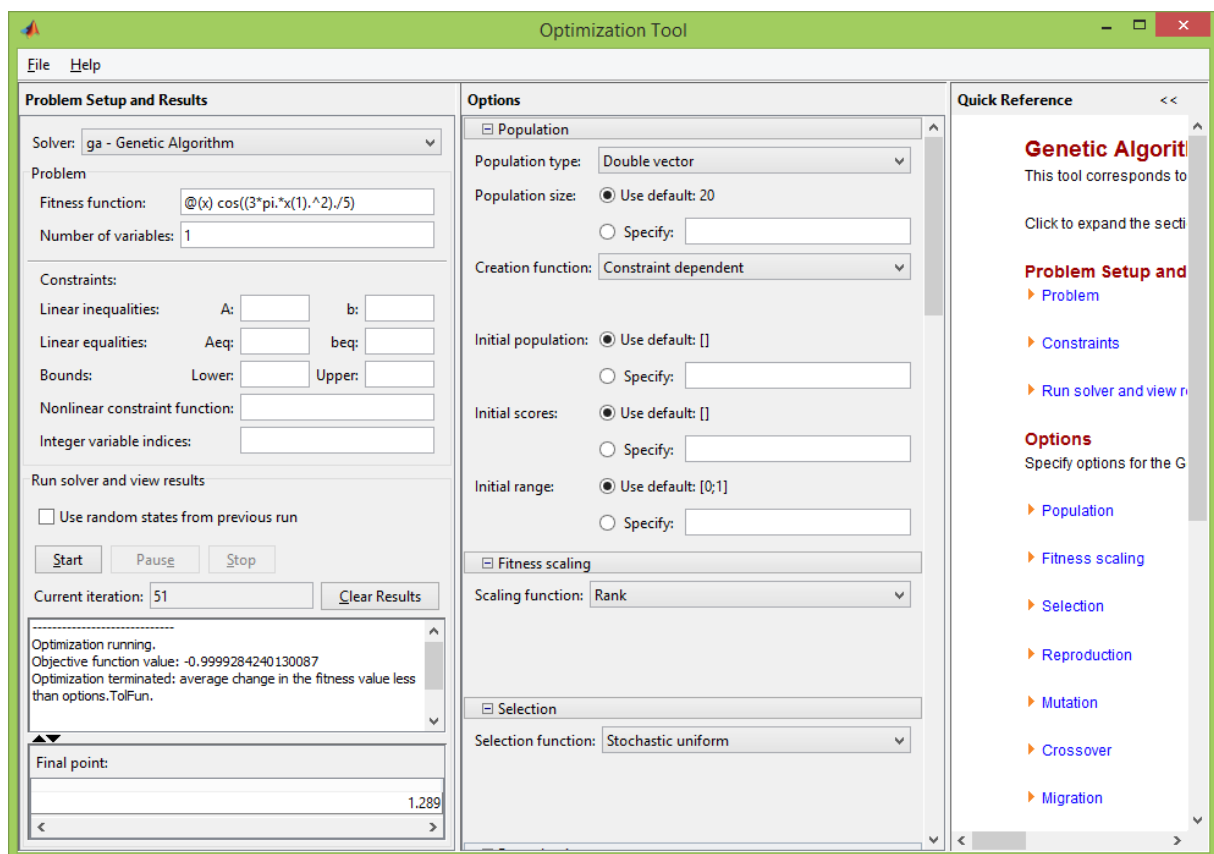
```
r=min(z)%klasican nacin traženja minimuma
disp('Minimum funkcije na klasican nacin')
h=min(r)
```



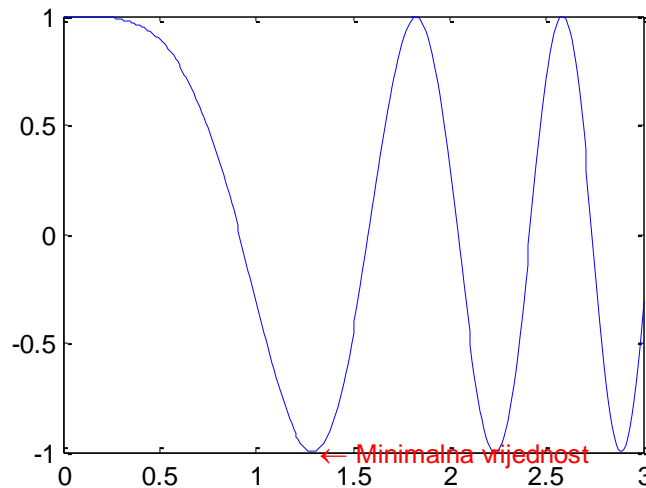
PRIMJER 3: Koristeći genetski algoritam naći minimum sljedeće funkcije:

$$y = \cos\left(\frac{3\pi \cdot x^2}{5}\right)$$

ukoliko je x na intervalu od 0 do 3.



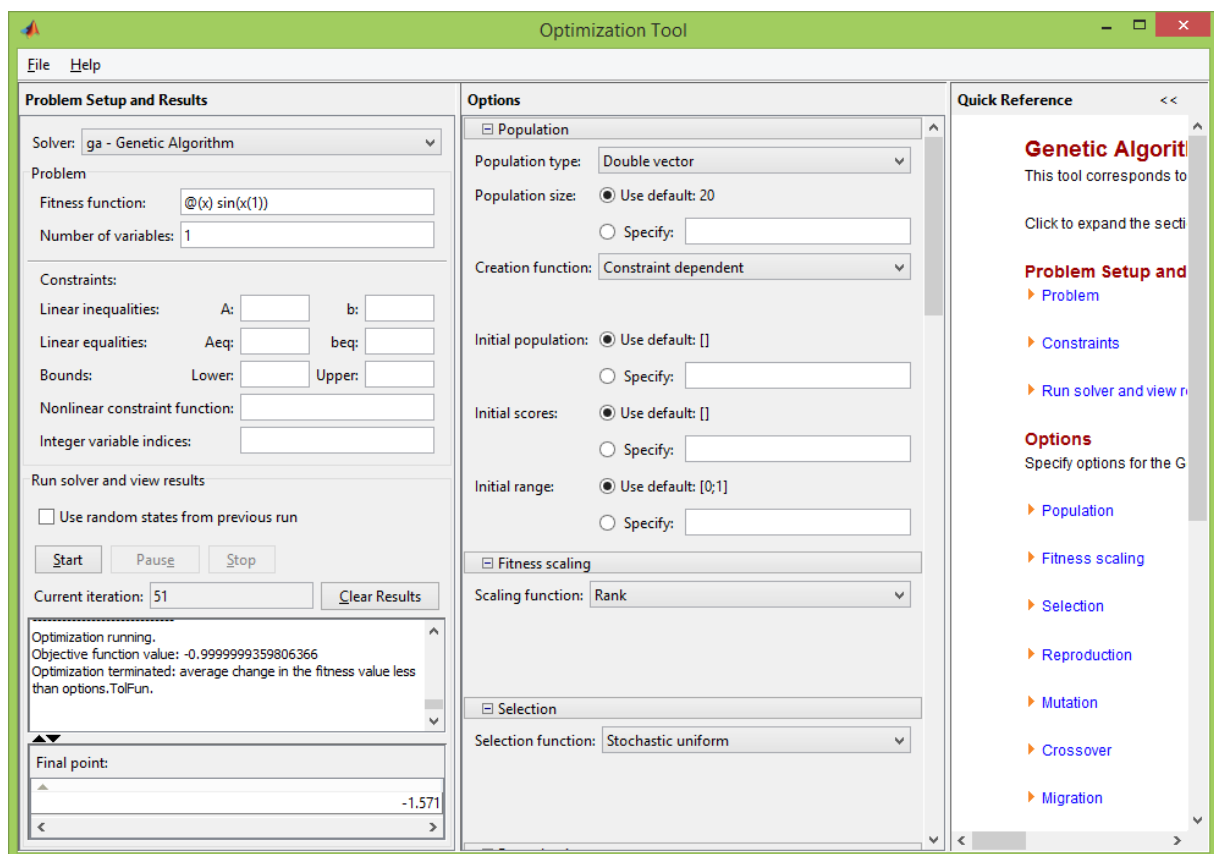
```
x=0:0.01:3;
y=cos((3*pi*x.^2)./5);
plot(x,y)
[x,f]=ga(@(x) cos((3*pi*x.^2)./5),1);
text(1.289, -0.9999284240130087, '\leftarrow Minimalna
vrijednost', 'FontSize',10, 'Color', 'red', 'LineWidth',4)
```



PRIMJER 4: Koristeći genetski algoritam naći minimum sljedeće funkcije:

$$y = \sin(\theta)$$

ukoliko je θ na intervalu od $-\pi$ do π .



```
x = -pi:.1:pi;
y = sin(x);
p = plot(x,y)
set(gca, 'XTick', -pi:pi/2:pi)
set(gca, 'XTickLabel', {'-pi', '-pi/2', '0', 'pi/2', 'pi'})
xlabel('-\pi \leq \Theta \leq \pi')
ylabel('sin(\Theta)')
title('sin(\Theta)')
```

```

text(-pi/2, sin(3*pi/2), '\leftarrow Minimalna
vrijednost', 'HorizontalAlignment', 'left')
set(p, 'Color', 'red', 'LineWidth', 2)

```

