

Vizualizacija genetskog algoritma za pronalazak globalnog minimuma funkcija dvije varijable

Ljudevit Jelečević
3. laboratorijska vježba

1. Uvod

Genetski algoritmi je primjer algoritama evolucijskog računanja koji se primarno sastoji od dva operatora – križanja i mutacije. Ovakvi algoritmi se koriste za pronalazak rješenja problema optimizacije i pretraživanja.

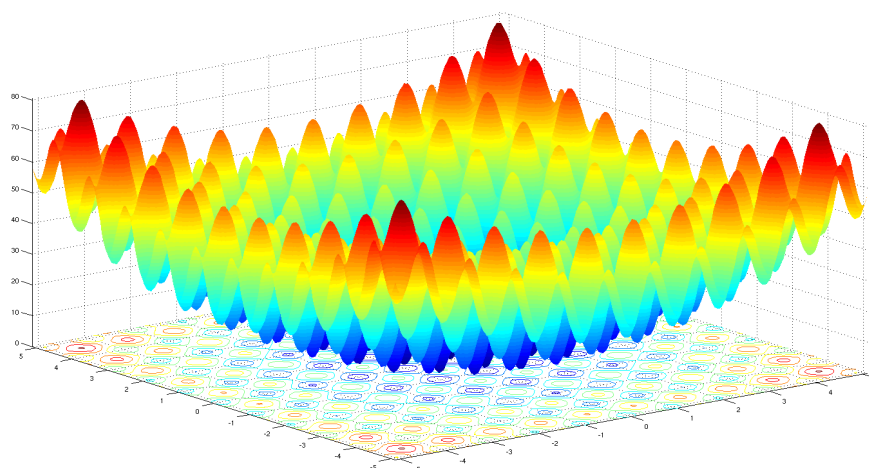
U sklopu ove laboratorijske vježbe implementiran je jednostavni eliminacijski genetski algoritam s ciljem pronalaska globalnog minimuma funkcije dvije varijable, tj. u ovome slučaju, za primjer, Rastriginove funkcije.

2. Rastriginova funkcija

Rastriginova funkcija je višemodalna funkcija koja se često koristi za testiranje optimizacijskih algoritama. Pronalazak globalnog minimuma ove funkcije je relativno težak problem zbog velikog broja lokalnih minimuma. Rastriginova funkcija glasi:

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

n – dimenzionalnost ulaznog prostora



Slika 1.

Rastriginova funkcija

3. Genetski algoritam

U svakoj generaciji ovoga algoritma populaciju rješenja predstavlja struktura fiksne veličine koja se sastoji od x , y i evaluacije rješenja funkcijom kazne koju želimo minimizirati. Kao funkcija kazne korišten je kvadrat vrijednosti funkcije.

Prilikom početka rada punimo populaciju prve generacije nasumično generiranim rješenjima, nakon čega ponavljamo korake selekcije, križanja i mutacije. U iduću generaciju prelazi određeni postotak najboljih jedinki i potreban broj novih rješenja nastalih križanjem kako bi se popunila generacija. Jedinke koje ćemo križati biramo obrnuto proporcionalno njihovoj funkciji kazne tako da bolja rješenja imaju veću vjerojatnost križanja. Nakon što smo popunili generaciju, sve jedinke mutiramo i evaluiramo nakon čega prelazimo u iduću iteraciju.

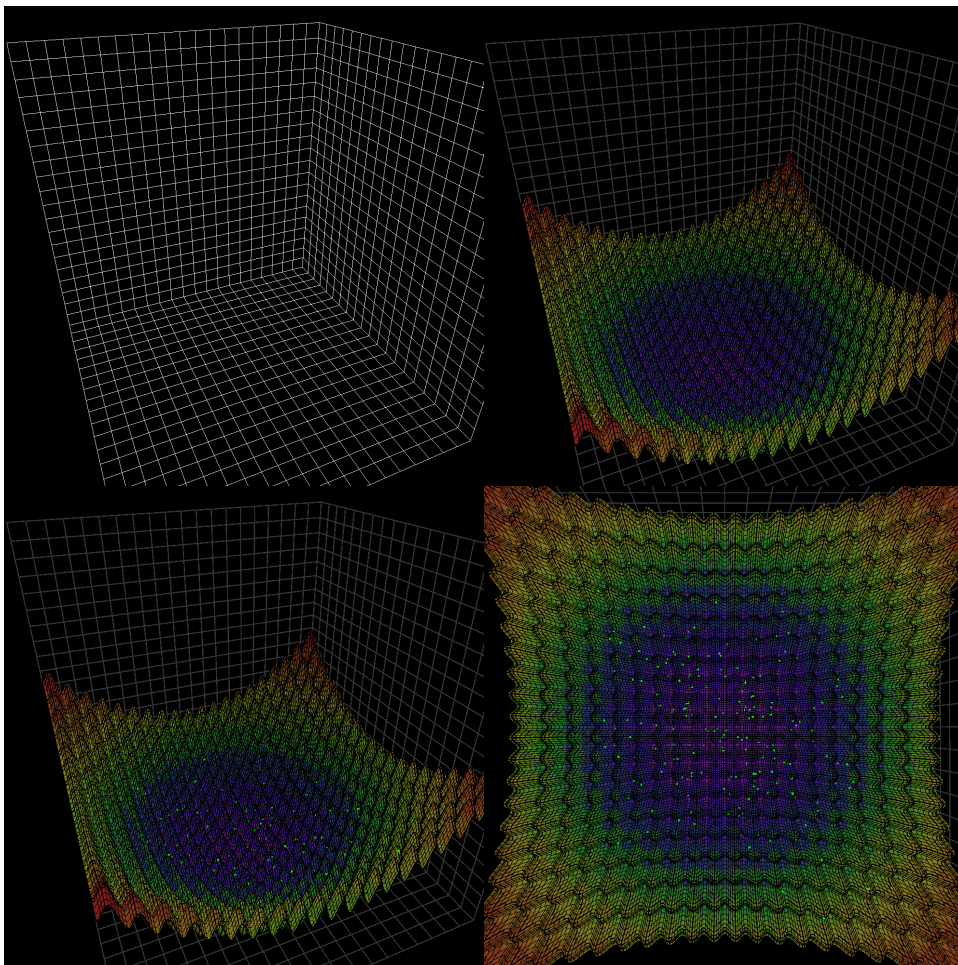
Kao operator križanja koristimo težinsku sumu vrijednosti odabranih roditelja dok prilikom mutiranja dodajemo povećavamo ili smanjujemo malih vrijednostima parametre roditelja. Ovako definiran operator mutiranja nije idealan i u određenim situacijama algoritam radi bolje ako ga isključimo. `drawGrid`, `drawIndividuals`, `draw3dFunction`.

4. Vizualizacija

Vizualizacija je napravljena pomoću OpenGL-a te je animacija temeljena na metodi `idle` u kojoj se interpolira između dvije generacije rješenja ili generirana nova generacija, ovisno o tome koliko je vremena prošlo od prethodnog crtanja.

Za crtanje su definirane funkcije:

- `drawGrid` – koristi pomoćnu metodu `drawWireframePlane` kako bi nacrtala koordinatni sustav
- `draw3dFunction` – crta površinu funkcije čiji minimum tražimo
- `drawIndividuals` – crta rješenje trenutne generacije evolucijskog algoritma reprezentirane kockama na površini funkcije čiji minimum tražimo



Slika 2. Izgled nakon `drawGrid`, nakon `drawGrid` + `draw3dFunction` i izgled nakon sve tri funkcije

5. Upute

Prije korištenja ovaj program je potrebno ili prevesti koristeći CMake ili na GNU/Linux samo pokrenuti izvršnu datoteku *3lab* dostupnu zajedno s programskim kodom na repozitoriju: <https://github.com/thelcrys1s/racani>.

Tijekom izvršavanja programa moguće je koristiti:

<i>space</i>	pokretanje/pauziranje
<i>esc</i>	izlaz iz programa
<i>u</i>	micanje za + 0.5 na x osi
<i>j</i>	micanje za - 0.5 na x osi
<i>i</i>	micanje za + 0.5 na y osi
<i>k</i>	micanje za - 0.5 na y osi
<i>o</i>	micanje za + 0.5 na z osi
<i>l</i>	micanje za - 0.5 na z osi
<i>r</i>	pogled od gore
<i>z</i>	zumirani pogled od gore