

Genetski algoritmi

Minimizacija funkcije Trogrba kamila

ALEKSANDAR STOJANOVIC RN97/2018

April 9, 2020

Chapter 1

Objašnjenje rada

1.1 Jezik i tehnicki detalji

Za rešavanja datog problema sam se odlučio za python programski jezik zbog lagane obrade podataka. Od biblioteka sam koristio sledeće:

- random
- sys
- decimal
- numpy
- pprint
- configparser - parsiranje fajlova
- matplotlib - grafik

Instaliranje ovih biblioteka vrši se tako sto iz istog foldera pozovemo sledeću komandu: *pip install -r requirements.txt*

Za pokretanje programa neophodno je sve fajlove staviti u isti folder.

Struktura fajlova je sledeća:

- config.properties
- output.txt
- main.py

U *config.properties* fajlu se nalaze default vrednosti koje ce algoritam koristiti ukoliko korisnik ne unese svoj fajl. Nakon sto algoritam završi sa radom u folderu će biti kreiran fajl koji sadrži rezultate pokretanja sa nazivom *output.txt*.

1.2 Algoritma

Korišćen je kontinualan genetski algoritam. Ovaj algoritam sadrži sledeće komponente:

- Funkcija prilagodjenosti,
- funkcija ukrštanja i
- funkcija mutacije.

One će biti objasnjene kasnije u posebnim sekcijama, za sada ću se fokusirati na parametre.

U folderu se nalazi *config.properties* fajl koji sadrži polja za konfiguraciju algoritma. Pri pokretanju programa, od korisnika će biti zatraženo da unese putanju do fajla za konfiguraciju. Ukoliko korisnik ne unese fajl, program će učitati vrednosti iz ovog fajla.

Ako pogledamo sastav fajla možemo primetiti dva poglavlja. Poglavlje *population* odnosi se na konfiguraciju se populacije i ovde se može podesiti:

- Veličina populacije (3 polja jer nemam skilla),
- Donja granica intervala,
- Gornja granica intervala,
- Koeficijent koji određuje mutaciju i
- Veličina test objekta.

Poglavlje *general* sadrži polja za konfiguraciju algoritma:

- Maksimalnog broja iteracija,
- Lokaciju output fajla,
- Broj ponavljanja po veličini populacije,
- Prikaz grafika i
- Čuvanje grafika u *.png* obliku

1.3 Metoda prilagodjenosti

Funkcija prilagodjenosti predstavlja način za rangiranje populacije uz pomoć fitness-a. Budući da mi radimo minimizaciju visedimenzionalne funkcije možemo nju koristiti kao funkciju troška. Ovo u prevodu znači da konkretno u našem slučaju hromozomi imaju u sebi dva gena $[x, y]$ koje ubacujemo u funkciju. Njihov fitness je zapravo vrednost funkcije za parametre koje smo prosledili.

Nama je poznat interval za koji funkcija radi i znamo da se očekivana nula nalazi u $f(0, 0) = 0$. Sa ovim možemo rangirati jedinke tako što ćemo na osnovu fitnessa sortirati populaciju u rastućem poretku. Jedinke koje su bliže nuli će imati najveći fitness.

1.4 Metoda ukrštanja - Redklif ukrštanje

Za metodu ukrštanja koristio sam Redklifovo jednostavno ukrštanje. U jednom od svojih radova Redklif je predložio jednostavnu formulu za izračunavanje vrednosti jednog parametra potomka iz vrednosti istog parametara kod njegova dva roditelja. Ova formula izgleda ovako:

$$\rho_{pot} = \beta * \rho_1 + (1 - \beta) * \rho_2$$

U ovoj formuli ρ_{pot} predstavlja parametar potomka, ρ_1 i ρ_2 su parametri prvog i drugog roditelja. β je slučajni broj kojim se određuje udeo vrednosti parametra prvog i drugog roditelja u vrednost parametra dobijenog potomka. Ovo se postiže tako što je β fiksiran između vrednosti $[0, 1]$. Kada je $\beta = 0$, vrednost parametra se prepisuje od drugog roditelja, kada je $\beta = 1$, vrednost se uzima od prvog roditelja, a kada je $\beta = 0.5$, vrednost je aritmetička sredina vrednosti parametra oba roditelja.

1.5 Metoda mutacije - Tačkasta normalna mutacija

Jednostavna metoda kod koje se na gen iz normalne raspodele dodaje slučajna vrednost. Bilo bi lepo definisati optimalan interval iz kojeg ćemo birati istu radi bržeg konvergiranja ka cilju ali ja sam se odlučio za vrednosti između 0 i 1.

Chapter 2

Izvršavanje

2.1 Grafik





