

Problem trgovačkog putnika

Stefan Dragičević

E9-4-2022

1. Projektni zadatak

Cilj ovog projekta je pronaći najkraći mogući put koji povezuje skup gradova, pri čemu se svaki grad posećuje samo jednom, a putovanje se završava u početnom gradu. Traži se najkraće rastojanje između 2 grada i redosled obilazaka gradova (kombinatorni problem).

Problem je iz stvarnog života, a na njega se nailazi prilikom pronalaženja optimalne putanje za dostavna vozila ili prilikom planiranja putovanja.

Problem je potrebno rešiti upotrebom genetskog algoritma (GA).

Testirati različite metode selekcije, ukrštanja i mutacije i ponoviti GA više puta sa različitim parametrima. Prikazati dobijene rezultate i prokomentarisati koji su parametri dali najbolje rešenje.

2. Teorijski deo

Genetski algoritmi spadaju u grupu stohastičkih optimizacionih metoda koje ne koriste derivacije odnosno izvode funkcija već su bazirani na konceptu prirodne selekcije i evolucije. Ovi algoritmi imaju veliki stepen prilagodljivosti i manje su podložni zapadanju u lokalni minimum.

Genetski algoritmi rade na principu pravljenja populacije koja se sastoji od jedinki, a svaka jedinka predstavlja jedno od rešenja datog problema. Svako rešenje je predstavljeno kao binarni niz koji se naziva hromozom. Do najboljeg rešenja se dolazi posle određenog broja generacija, tako što se jedinke menjaju primenom genetskih operatora. Osnovni genetski operatori su selekcija, ukrštanje i mutacija. Jedinke koje imaju bolje rešenje imaju veće šanse da prežive i da se ukrštaju.

Nakon formiranja populacije izračunava se stepen prilagođenosti svake jedinke. Nova populacija se formira na osnovu postojeće generacije tako što se vrši selekcija jedinki. Najprilagođeniji članovi populacije se automatski prebacuju u sledeću generaciju i taj proces se naziva elitizam. Druge jedinke sa većim stepenom prilagođenosti postaju roditelji i na taj način se slabije jedinke izbacuju iz populacije. Kada je određeno koje jedinke će postati roditelji vrši se njihovo ukrštanje tako što se hromozomi jednog roditelja kombinuju sa hromozomima drugog roditelja. Ukrštanjem se dobijaju 2 potomka koji imaju delove hromozoma od oba roditelja. Pored ukrštanja prisutne su i mutacije. Na slučajan način se bira mali broj jedinki čiji hromozomi će se mutirati. Na taj način se unosi novi genetski materijal i sprečava se da populacija konvergira ka lokalnom minimumu. Stepenn mutacije određuje koji procenat populacije će

biti mutiran i on se obično drži na niskom nivou da se dobri hromozomi dobijeni ukrštanjem ne bi izgubili. Ako je stepen mutacije visok algoritam se približava slučajnom pretraživanju.

3. Praktični deo

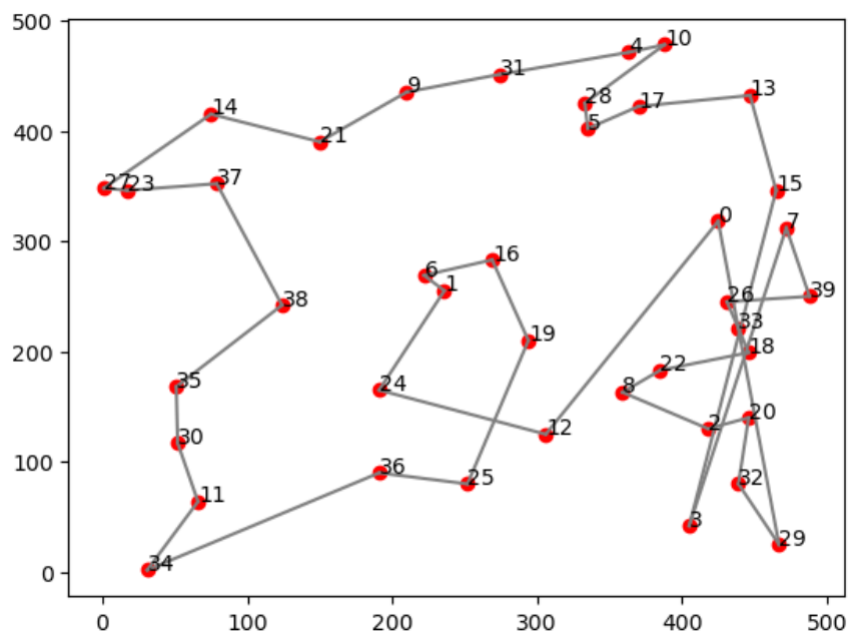
Ovaj projekat je rađen u python programskom jeziku u Jupyter Notebook-u. Na samom početku bilo je potrebno definisati gradove. Svaki grad je generisan na slučajan način u 2D prostoru ali tako da se svaki put generišu isto kako bi se rezultati mogli uporediti. Svaki grad je predstavljen kao niz od 3 elementa, gde je prvi element broj grada odnosno njegov naziv, a drugi i treći element su njegove x i y koordinate. Posle toga su svi gradovi stavljeni u jednu listu, a ta lista označava jednu jedinku u populaciji. Redosled gradova u listi se izmeša i tako se dobijaju različite jedinke. Nakon toga bilo je potrebno izračunati rastojanje između gradova. To je urađeno tako što je sumirano euklidsko rastojanje između svaka dva susedna grada u listi.

Na kraju je definisan genetski algoritam koji prima čitavu populaciju i različite parametre vezane za mutacije i ukrštanje. Genetski algoritam pravi novu generaciju koja ima isti broj jedinki kao prethodna generacija, on u startu uzima dve najbolje jedinke iz prethodne generacije i prebacuje ih u sledeću, takozvani elitizam. Ostatak populacije se dobija ukrštanjem, mutacijom ili slučajnim biranjem jedinki iz prethodne generacije. Definiše se verovatnoća ukrštanja i mutacije i u slučaju da dođe do ukrštanja radi se ukrštanje u jednoj tački. Na slučajan način se bira 10% populacije i taj deo će da učestvuje u turniru, pobednik jednog turnira postaje jedan roditelj, a pobednik drugog turnira postaje drugi roditelj. Tačka preseka hromozoma se bira nasumično i oba deteta dobijaju deo genetskog materijala od oba roditelja. U slučaju mutacije na slučajan način se biraju dva hromozoma koja će zameniti mesta i to se dešava na nivou jedne jedinke. Ako nije došlo ni do mutacije ni do ukrštanja bira se nasumično jedinka koja se prebacuje u sledeću generaciju.

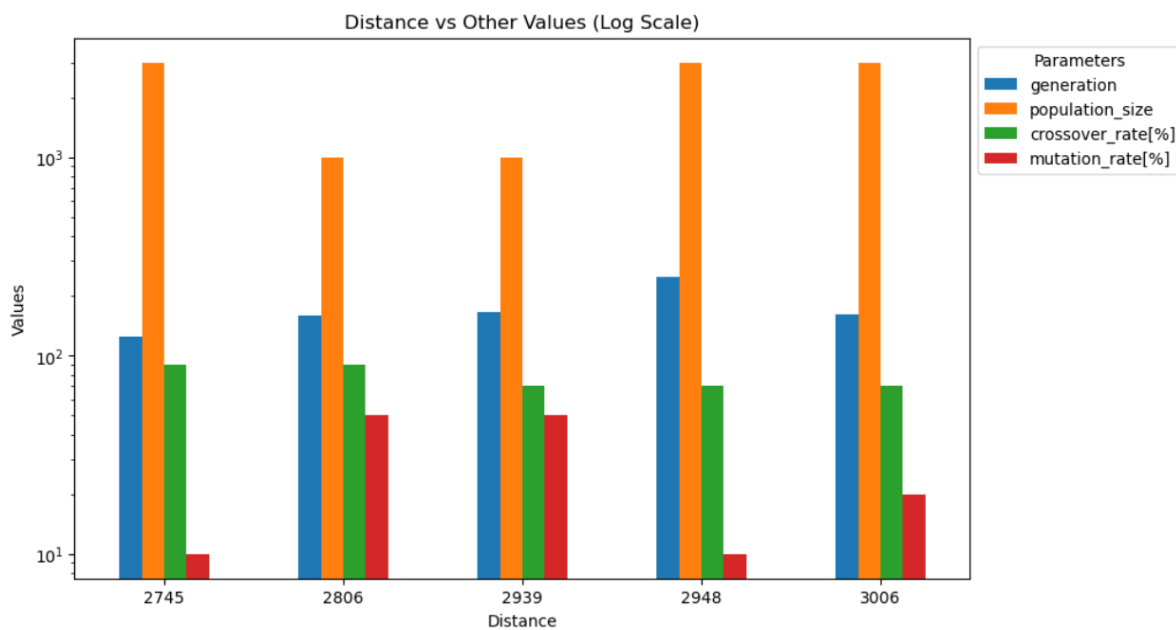
Kako bi se dobilo što bolje rešenje za problem trgovačkog putnika isprobane su različite vrednosti parametara za veličinu populacije [100, 500, 1000, 3000], verovatnoću ukrštanja [0.5, 0.7, 0.9] i verovatnoću mutacije [0.1, 0.2, 0.5]. Najbolji rezultat za 40 gradova je dala kombinacija parametara gde je veličina populacije 3000, verovatnoća ukrštanja 90% i verovatnoća mutacije 10%, putanja koja je dobijena je prikazana na slici 1. Do ovog rezultata se došlo posle 125 generacija. Što je populacija veća broj rešenja problema se povećava tako da je i logično da je najveća populacija dala najbolje rešenje. Ona je imala veliki stepen ukrštanja tako da je u svakoj novoj generaciji postojao veliki broj novih rešenja koja su sadržala delove najboljih rešenja iz prethodnih generacija. Stepen mutacije je nizak i to omogućava da se sačuvaju potomci koji su dobijeni ukrštanjem. Zavisnost pređenog puta od ostalih parametara za 5 najboljih rešenja je data na slici 2.

Najbolji rezultat za 50 gradova je dobijen kada je veličina populacije bila 1000, verovatnoća ukrštanja 90% i verovatnoća mutacije 50%, broj generacija koji je bio potreban da bi se došlo do ovog rezultata je 209 i putanja koja je dobijena se nalazi na slici 3. Objašnjenje za ove parametre je da je zbog velikog stepena mutacije bila potrebna manja populacija jer visok stepen mutacije omogućava bolju pretragu prostora. Zavisnost pređenog puta od ostalih parametara za 5 najboljih rešenja je data na slici 4.

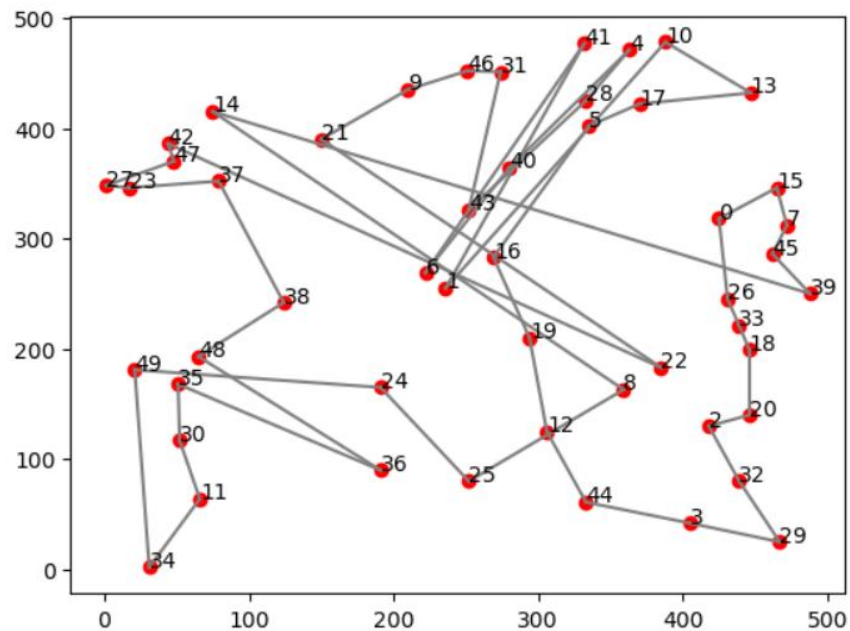
Genetski algoritam je definisan tako da će pokušati da nađe najbolje rešenje tokom 1000 generacija ali u slučaju da se najbolja putanja ne promeni tokom 50 generacija algoritam će se automatski zaustaviti kako ne bi trošio resurse.



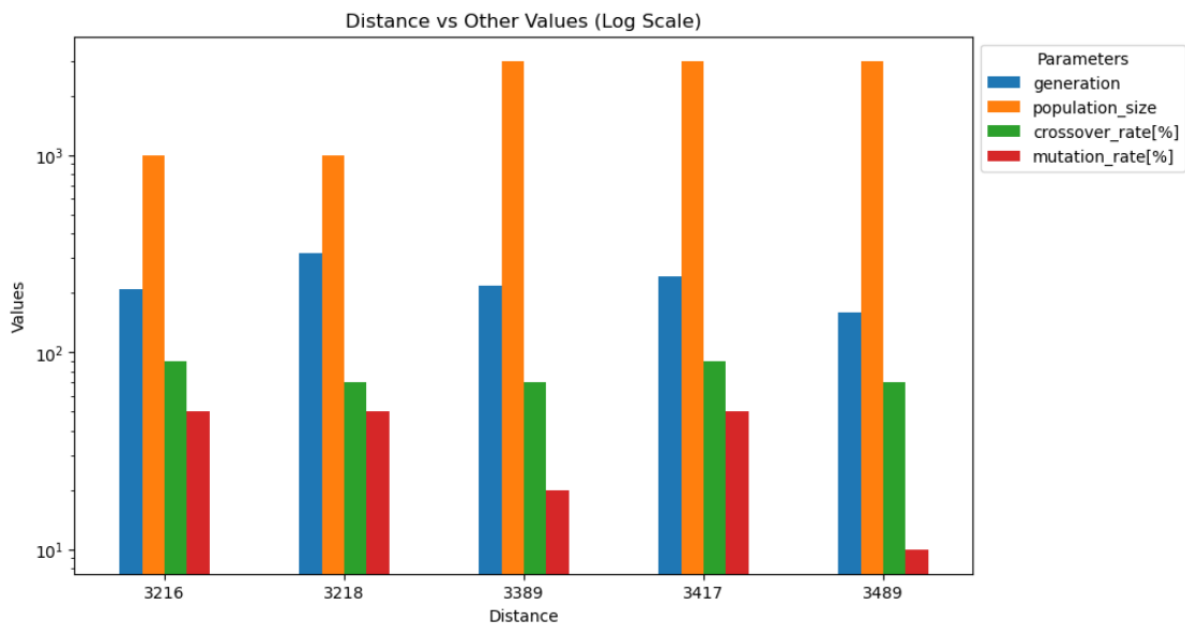
Slika 1 Najbolja putanja za 40 gradova



Slika 1 Zavisnost predenog puta od drugih parametara



Slika 3 Najbolja putanja za 50 gradova



Slika 4 Zavisnost predenog puta od drugih parametara

Python kod ovog projekta se može pronaći na [github-u](#).