



Univerzitet u Novom Sadu  
Fakultet tehničkih nauka



## Dokumentacija za projektni zadatak

Studenti: Olivera Radovanović, SV46/2021  
Marko Mitošević, SV56/2021

Predmet: Nelinearno programiranje i evolutivni algoritmi

Broj projektnog zadatka: 14

Tema projektnog zadatka: Ant colony optimization algoritam, problem putujućeg trgovca

## Opis problema

Problem putujućeg trgovca – dat je skup gradova u kom je svaki grad određen pomoću dve koordinate  $x$  i  $y$ . Kojim redom treba obilaziti gradove tako da ukupan pređeni put bude najmanji mogući, a da se obiđu svi gradovi i da se putovanje završi u onom gradu odakle je počelo? Pređeni put se računa kao Euklidska udaljenost između dve tačke u ravni. Dakle, ako posmatramo gradove  $g_1$  i  $g_2$  određene svojim koordinatama  $(x_1, y_1)$  i  $(x_2, y_2)$ , pređeni put od jednog grada do drugog predstavlja udaljenost te dve tačke:

$$d(g_1, g_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

## Uvod

Da bi rešili problem putujućeg trgovca, gradove i rastojanja između gradova ćemo prikazati preko ponderisanog (težinski) neusmerenog grafa. Ovaj problem bi se mogao rešiti i uz pomoć teorije grafa koristeći jedan od algoritama za traženje najkraće razdaljine tačaka u grafu, ali bi se onda performanse rešavanja eksponentalno pogoršavale sa povećanjem broja gradova.

Zbog toga koristimo algoritam optimizacije kolonije mrava (ant colony optimization algorithm) čije performanse su znatno bolje za veći broj gradova. Algoritam simulira prirodno ponašanje kolonije mrava prilikom traženja hrane u njihovoj okolini. Naime, svaki mrav nezavisno izabere svoju putanju. Kada nadje hranu, mrav se vrati u koloniju i ispušta feromone - izlučenje hemijskih faktora koje drugim mravima govori kojom putanjom se dati mrav kretao da bi stigao do hrane. Što je put do hrane kraći, mrav će češće donositi hranu u koloniju i time lučiti veću količinu feromona. Drugi mravi će primetiti povećanu količinu feromona i krenuti istom putanjom kao dati mrav, čime će se povećati efikasnost nabavljanja hrane za celu koloniju.

## Implementacija

Podaci gradova su smešteni u tekstualnoj datoteci u uređenoj trojci:

(redni broj grada,  $x$  koordinata,  $y$  koordinata)

Radi lakšeg obrađivanja podataka, kreirana je klasa Grad koja ima 4 atributa:

- 1) redni – broj linije u kojoj se grad nalazi u samoj datoteci (potrebno za pravilno formiranje matrice grafa)
- 2) broj – redni broj grada
- 3) x – x koordinata
- 4) y – y koordinata

Prilikom čitanja podataka iz datoteke sve gradove smestamo u listu gradova `svi_gradovi` a zatim pravimo matricu rastojanja `matrice_gradova` koja predstavlja ponderisani neusmereni graf, gde su vrste i kolone gradovi a vrednost grana polja u matrici. Glavna dijagonala je jednaka beskonačnosti da ne bi remetila algoritam jer je rastojanje između grada i samog sebe 0.

Algoritam optimizacije mrava pozivamo funkcijom `odredi_najbolju_putanju`.

Parametri algoritma su:

- 1) `n` – broj iteracija
- 2) `stopa_isparivanja` – stopa isparivanja feromona
- 3) `broj_mrava` – broj mrava u jednoj generaciji
- 4) `alfa` – stepen vrednovanja istraživanja
- 5) `beta` – stepen vrednosvanja eksploatacije

Više o optimalnim vrednostima parametara će biti diskutovano u zaključku.

Feromoni se simuliraju preko matrice feromona, koja je istih dimenzija kao i `matrica_gradova`. Početna vrednost svih polja u matrici feromona je ista i iznosi recipročnu vrednosti veličine matrice.

## Iterativni postupak

Put kojim mrav prolazi se simulira funkcijom `sve_mogucnosti`, koja svakom mravu dodeljuje jednu putanju koja prolazi kroz sve gradove. Putanja je određena nasumično koristeći funkciju `choice` biblioteke `numpy`, koja bira sledeći grad koji će mrav posetiti na osnovu verovatnoća izračunatih iz funkcije `verovatnoća_izbora`.

Verovatnoća izbora  $v$  grane matrice  $i$ -tog reda i kolone  $j$  se računa na osnovu formule:

$$v_{i,j} = \frac{p_{i,j}^{\alpha} * d_{i,j}^{\beta}}{\sum_z p_{i,z}^{\alpha} * d_{i,z}^{\beta}}$$

$p$  – matrica feromona

$d$  – matrica rastojanja

$z$  – uzima vrednosti preostalih gradova,

$\alpha$  – alfa

$\beta$  – beta

Nakon što svi mravi izaberu svoj put, matrica feromona se ažurira funkcijom `azuriraj_feromona` gde se svakom polju matrice feromona dodaje recipročna vrednost razdaljine kojom je prešao mrav koji je prošao kroz granu koja odgovara datom polju matrice. Za trenutno najkraću rutu uzimamo najkraću rutu u ovoj generaciji mrava, a zatim proveravamo da li je ona najkraća ruta ikada dobijena.

Na kraju iteracije vrednosti matrice feromona smanjimo za stopu isparivanja, koja predstavlja procenat za koji ćemo smanjiti vrednost i simulira isparivanje čestica feromona vremenom u prirodi.

Iterativni postupak se ponavlja  $n$  puta, gde  $n$  predstavlja broj generacija mrava.

## Rezultati

Efikasnost i tačnost algoritma veoma zavisi od vrednosti parametara. Testiranjem algoritma sa različitim vrednostima parametara utvrdili smo optimalne vrednosti za sve parametre koje smo koristili.

## Vremensko ograničenje

Na vreme izvršavanja algoritma utiče broj iteracija i broj mrava unutar jedne generacije. Za vremensko ograničenje izvršavanja programa smo proizvoljno uzeli 10 sekundi. Za što preciznije

rešenje koje takodje zadovoljava dato ograničenje, dobili smo da broj iteracija ne treba da premašuje 100, kao i da broj mrava ne treba da premašuje 45. Ove granične vrednosti su uzete za optimalne.

## Pospešivanje tačnosti

Za što tačniji rezultat, nakon broja iteracija i broja mrava, je najbitniji odnos promenljivih alfa i beta. Alfa predstavlja koliko algoritam vrednuje istraživanje novih neotkrivenih ruta, dok beta predstavlja vrednovanje eksploatacije već pronadjenih efikasnih ruta. Za najoptimalniju razmeru alfa prema beta smo dobili 1:12, što pokazuje da je korišćenje dobrih ruta ključno na početku algoritma da bi kroz generacije algoritam bolje konvergirao ka optimumu. Za datu razmeru smo dobili konstanto poboljšanje u konačnom rezultatu za više od 100%, naime najmanja razdijelina nam se smanjila za pola u odnosu na početan odnos parametara 1:1, za datu test datoteku (sa 18000 na 8000).

Stopa isparivanja feromona reflektuje prirodnu pojavu isparivanja čestica feromona kroz vreme i ne igra preveliku ulogu u optimizaciji i može da utiče na promenu krajneg rešenja za svega 5-10%. Imavši to u vidu, za najoptimalniju vrednost ovog parametra smo dobili 10%. Sve vrednosti preko 20% mogu da pogoršaju krajnje rešenje u nezanemarljivoj količini.

## Zaključak

Upotrebom algoritma optimizacije kolonije mrava uspešno smo rešili problem putujućeg trgovca sa zadovoljavajućim rezultatom pri čemu se algoritam izvršava u prihvatljivo kratkom vremenskom roku.

Algoritam može biti još precizniji povećanjem broja mrava u generaciji i broja iteracije, međutim time kompromitujemo vreme izvršavanja algoritma.