Improving metaheuristics by using learning algorithms to predict good regions of search space

Projekat u okviru kursa Računarska inteligencija
Matematički fakultet
Univerzitet u Beogradu

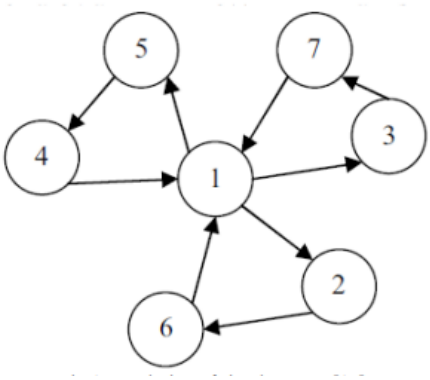
Vladimir Knežević
mi17206@alas.matf.bg.ac.rs

September 2023

Sadržaj

1 Opis problema

Rešavanje problema poboljšanja metaheuristike korišćenjem algoritama učenja za predviđanje dobrih regiona pretrage ćemo prikazati na primeru **mTSP-a** (Multiple traveling salesman problem). TSP je poznat problem kombinatorne optimizacije gde prodavac mora da pronađe najkraći put do **n** gradova i povratak u početni grad. Dok se TSP ograničava na problem sa jednim trgovcem, **mTSP** je problem koji posmatra više od jednog trgovca. Pristup koji ćemo koristiti je da grupišemo susedne gradove u klastere pomoću **K-means** algoritma i zatim unutar tih klastera koristiti **ACO** (Ant colony optimization) algoritam. U mTSP problemu imamo neusmereni povezan graf koji sadrži E grana koje povezuju V čvorova. Svaki trgovac će obići neki podskup čvorova i vratiće se u početni čvor. Ukupan zbir cena putanja svih trgovaca treba biti minimalna.



Slika 1: Jednostavan primer mTSP-a

2 Implementacija

Prvo ćemo opisati implementaciju K-means algoritma za klasterovanje čvorova,a zatim ACO algoritam koji primenjujemo nad tim klasterima.

2.1 K-means algoritam

K-means algoritam je jedan od najpopularnijih algoritama za klasterovanje. Ovo klasterovanje je zasnovano na centroidu, odnosno elementi se postavljaju u onu grupu čiji im je centroid najbliži.

1. Na početku odaberemo željeni broj klastera **K** na koje želimo da podelimo naš skup.
2. Zatim na nasumičan način inicijalizujemo centroide, gde ćemo imati onoliko centroida koliko i klastera.
3. Zatim za svaki element skupa računamo euklidsko rastojanje između njega i centroida i određujemo kom centroidu pripada odnosno kom klasteru.
4. Zatim ćemo da reinicijalizujemo centroide računajući prosek koordinata elemenata datog klastera.

5. Nakon toga ćemo da ponavljamo 3. i 4. korak sve dok ne dobijemo optimalne centroide, odnosno dok se čvorovi ne grupišu u odgovarajuće klastere.

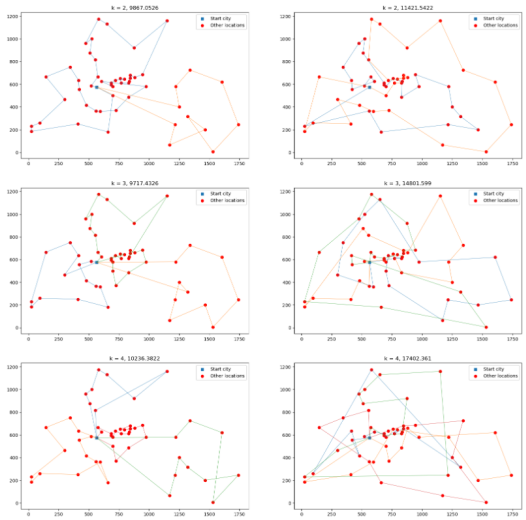
Glavna mana ovog pristupa je što kvalitet algoritma značajno zavisi od inicijalnog odabira centroida.

2.2 ACO algoritam

ACO algoritam je metod inteligencije roja koji koristi metaheurističke optimizacije. Princip na kom se zasniva ovaj algoritam je prirodno ponašanje mrava kada traže put od njihove kolonije do izvora hrane. Dok mravi traže hranu oni otpuštaju hemijsku supstancu koja se naziva **feromon**. Ova supstanca im pomaže da pronađu put nazad, i pomaže ostalim mravima da znaju kojim putem je neko već išao. Glavna ideja je da je jačina feromona veća na onim putevima koji su kraći, jer zbog dužine puta i intenzivnosti prolaska mrava ne stigne da ispari, i onda je i prolaznost mrava veća jer prate tu jačinu feromona. Takođe i isparivanje feromona pomaže protiv konvergencije u lokalno optimalnim rešenjima.
ACO algoritam se izvršava tako što se prvo događa određeno isparivanje feromona sa svih putanja, zatim se za svaki pređeni put dodaje određena količina feromona koja zavisi od parametra **q** i dužine datog puta.

3 Rezultati modela

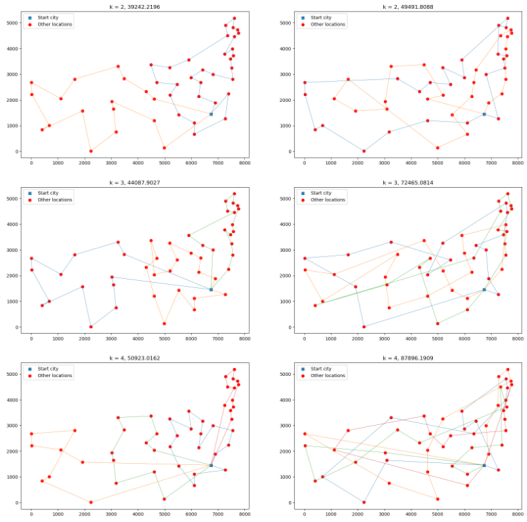
3.1 Berlin53: KMeans vs Random initialization



Slika 2: Berlin53: KMeans vs Random initialization

Sa leve strane grafikona se nalaze rezultati dobijeni klasterovanjem lokacija primenom K-means algoritma, dok se sa desne strane nalaze lokacije klasterovane nasumičnim odabirom gde u svakom klasteru ima otprilike jednak broj lokacija. Kao što možemo videti, dužine putanja koje trgovci pređu su nezanemarljivo manje kada koristimo K-means da klasterujemo lokacije. Takođe, putanje izgledaju prirodnije, nema škakanjaša jednog kraja na drugi kao što je to slučaj na podgrafikonima sa desne strane

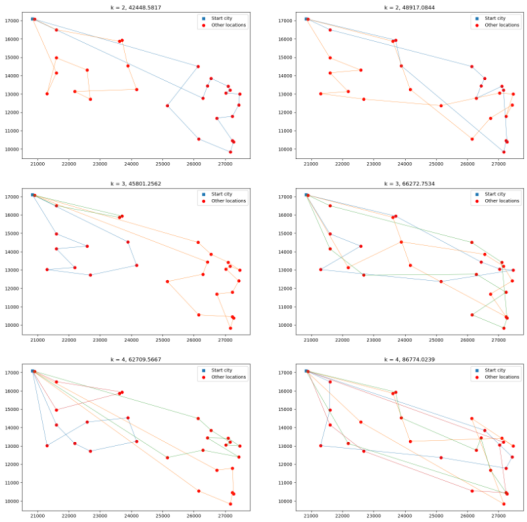
3.2 Att48: KMeans vs Random initialization



Slika 3: Att48: KMeans vs Random initialization

Slično kao i kod instance Berlin52, dužine putanja dobijenih klasterovanjem pomoću K-means algoritma su znatno manje nego kada nasumično inicijalizujemo klastere.

3.3 WI29: KMeans vs Random initialization



Slika 4: WI29: KMeans vs Random initialization

Ista stiuacija je i kod ove instance kao i kod prethodne dve - dužine putanja su kraće i putevi kojima se trgovci kreću su dosta prirodniji.

4 Zaključak

Primenom K-means algoritma za inicijalno klasterovanje lokacija dobijamo bolje rezultate u odnosu na nasumično klasterovanje lokacija čak i pri podrazumevanim parametrima i pri standardnoj implementaciji ACO algoritma. Potrebno je razmotriti unapređenje ACO radi daljeg poboljšanja rezultata, ali i ovako dobijeni rezultati su zadovoljavajući.