Projekt 1: Greedy algoritam za TSP

Irhad Šarić
Prirodno-matematički fakultet u Sarajevu
Napredni algoritmi i strukture podataka
irhad.saric@hotmail.com

***Abstract* – U ovom radu će biti opisan jedan od greedy algoritama za Traveling Salesman Problem i rezultati će biti upoređeni sa druga dva greedy algoritma (algoritam najbližeg susjeda i algoritam najmanje grane).**

I. UVOD

Traveling Salesman problem predstavlja problem putujućeg trgovca koji mora obići n gradova polazeći od grada u kojem se trenutno nalazi na način da se vrati iz polazećeg grada nakon što se obiđu svi ostali gradovi i da pri tome pređe što je moguće manje kilometara. Ovaj problem spade u klasu NP teških problema što znači da se moraju koristiti neke heuristike za njegovo rješavanje. Jedan od greedy algoritama za rješavanje ovog problema je i algoritam koji će biti opisan u nastavku.

Na početku je potrebno odabrati grad iz kojeg se polazi (u nastavku 0). Nakon toga se kreira tura sljedećeg oblika:

$$0 - 1 - 0 - 2 - 0 - 3 - \cdots - n - 0$$

Ovo nije validna tura jer se jedan grad posjećuje više puta. Ideja algoritma je da se nađu dva grada i i j kod kojih je najveća ušteda puta tako da se ide direktno iz jednog grada u drugi bez da se vraća u početni grad.

II. ALGORITAM

Na početku programa učitava se tekstualni fajl imena države za koju se traži rješenje. Svaki red u fajlu predstavlja jedan grad date države. Fajl se sastoji iz tri kolone, u prvoj koloni je indeks grada, drugoj geografska širina, a u trećoj geografska dužina grada. Nakon toga se pravi graf čiji su čvorovi gradovi date države.

Nakon što je napravljen graf, pravi se prioritetni red, koji će sadržati grane između gradova (i, j) pri čemu je $i \neq j, i \neq 0, j \neq 0$. One će biti sortirane na način da je jedna manja od druge ako je prvoj vrijednost atributa ušteda manja ili jednaka od atributa ušteda druge grane. Atributi se računaju u hodu na sljedeći način:

$$\begin{aligned} ušteda(i,j) = udaljenost(pocetni, i) \\ + udaljenost(j, pocetni) \\ - udaljenost(i, j) \end{aligned}$$

Također se pravi vektor gradovi i vektor stepeni , koji u sebi sadrži indekse svih gradova osim pocetnog i stepene čvorova redom. Pod stepenom čvora se podrazumijeva broj njegovih susjeda.

Sve dok u vektoru gradovi imamo više od dva grada radi se sljedeće:

1. Skida se grana s reda
2. Provjerava se da li su stepeni krajeva grane manji od dva. U slučaju da jesu grana se dodaje u rutu i povećavaju se stepeni gradova za jedan.
    a. Provjerava se da li je nakon dodavanja grane stvoren ciklus. Ako jeste grana se izbacuje iz ture i stepeni gradova se smanjuju za jedan.
3. Provjeravaju se stepeni gradova posljednje grane. U slučaju da je stepen grada i jednak dva, onda se taj grad izbacuje iz vektora gradovi. Isto se radi i za grad j pri čemu su gradovi i i j početak i kraj posljednje grane.

Kada u vektoru gradovi ostanu samo dva grada to znači da jedino ta dva grada imaju stepen jedan, pa će se oni povezati sa početnim čvorom i te grane će se dodati u rutu.

III. VREMENSKA KOMPLEKSNOST

Vremenska kompleksnost umetanja svih grana između gradova je $O(n^2 logn)$ zbog toga što za n gradova je moguće napraviti $n * (n-1)$ grana, i svaku tu granu treba pohraniti u prioritetni red a za to je potrebno $O(logn)$ vremena.

Pošto se može obraditi maksimalno n^2 grana, i za svaku granu je potrebno ispitati da li postoji ciklus u do tada nađenoj ruti, za što je potrebno $O(k)$ vremena, pri čemu je k dužina trenutne ture. Iz ovoga se vidi da je ukupna vremenska kompleksnost algoritma jednaka $O(n^2 * k)$.

IV. POREĐENJE RJEŠENJA

U Tabeli 1. će biti prikazane dužine tura dobijenih pomoću prethodno opisanog greedy algoritma, algoritma najbližeg susjedstva i algoritma najmanje grane.

Država	Najbliži susjed	Najmanja grana	Greedy	Optimalna tura
Djibouti (38)	9745	7019	6664	6656
BiH (192)	30591	30030	26476	-
Luksemburg (980)	14370	13464	12342	11340
Sahara(29)	36388	39691	28953	27603
Urugvaj (734)	99247	91558	87154	79114
Oman (1979)	120908	111191	98982	86891

Tabela 1. Dužine tura dobijene primjenom algoritama

U Tabeli 2. će biti prikazano vrijeme potrebno da se algoritmi izvrše.

Država	Najbliži susjed	Najmanja grana	Greedy
Djibouti (38)	0s	0.001s	0.001s
BiH (192)	0s	0.007s	0.036s
Luksemburg (980)	0.022s	0.178s	2.675s
Sahara(29)	0s	0.001s	0.001s
Urugvaj (734)	0.012	0.098	1.246s
Oman (1979)	0.085s	0.768s	19.179s

Tabela 2. Vrijeme izvršavanja algoritama

V. ZAKLJUČAK

Iz Tabele 1. i Tabele 2. se može primijetiti da greedy algoritam daje dosta bolje rezultate što se tiče dužine ture, ali za države s većim brojem gradova je potrebno dosta više vremena za bolji rezultat. Bitno je naglasiti da se je za početni grad uzeo prvi grad iz svake države, te da se promjenom grada može dobiti bolja ili gora ruta.

VI. REFERENCE

[1] http://www.math.uwaterloo.ca/tsp/world/countries.html?fbclid=IwAR1-haUiJbEp_NvtVnf93A-So72LrMNfxdz2iNXXWWwq7x4s2ppdNu74lyE#LU

[2] https://www2.seas.gwu.edu/~simhaweb/champalg/tsp/tsp.html?fbclid=IwAR3wa2PE2A10EdINpGIMKwd6BLQ0rDXQKqv091n2PzOMgPOSkIWn1DmyN-A