PMF BanjaLuka

PROJEKTNI ZADATAK

Predmetni professor:

Prof. dr. Dragan Matić

Student:

Srđan Paspalj

1. UVOD

U ovom radu se posmatra problem označavanja čvorova u velikim usmjerenim grafovima, težinskim grafovima koji imaju delimično poznate oznake.Grafovi se nasumično generišu, sa unaprijed određenim parametrima (gustina,težina...).U samom radu se koriste algoritmi za označavanje čvorova, optimizaciju označavanja na dva načina.Mjeri se vrijeme izvršavanja programa, svi rezultati i vremena se zapisuju.

2. STRUKTURE PODATAKA

U projektnom zadatku se koriste razne strukture podataka:

1. Lista grana – za svaku granu se čuva sa tri parametra: početni čvor,krajnji čvor,težina.

2. Llista susjeda – za svaki čvor se čuva lista susjednih čvorova,potrebno za Djikstra algoritam.

3. Matrica udaljenosti – čuva poznatu najkraću putanju od izvornog čvora do svih ostalih.

4. Min-Heap – koristi se za efikisnu implementaciju Djikstra algoritma,omogućava brz pristup čvoru sa najmanjom udaljenošću.

5. Lista oznake čvorova - čuva oznake čvorova.

3. ALGORITMI

1. Generisanje slučajnog grafa

Za kreiranje grafa primenjuje se metoda slučajnog generisanja grana.

Broj čvorova varira između 1000, 5000 i 10000.

Gustina grafa podešava se na vrednosti 0.3, 0.5 i 0.7, što predstavlja verovatnoću da između dva čvora postoji grana.

Težine grana nasumično se biraju iz intervala [-50, 50].

Petlje (grane koje spajaju čvor sam sa sobom) nisu dozvoljene.

Ovim postupkom se dobija težinski usmeren graf koji služi kao polazna tačka za dalje analize.

2. Provera povezanosti grafa

Nakon generisanja proverava se da li je graf jako povezan, tj. da li postoji put između bilo koja dva čvora.

Koristi se algoritam BFS (Breadth-First Search) ili alternativno DFS (Depth-First Search).

Ako graf nije jako povezan, dodaju se minimalne grane kako bi se osigurala povezanost.

3. Dodeljivanje inicijalnih oznaka

Čvorovi grafa se nasumično označavaju:

30% čvorova oznakom 0,

40% čvorova oznakom 1,

30% ostaje neoznačeno.

Za izbor označenih čvorova koristi se algoritam random shuffle, kojim se lista svih čvorova nasumično permutuje, a zatim se prvi deo niza dodeljuje oznakama.

4. Algoritam najkraćeg puta

Za pronalaženje udaljenosti između čvorova koristi se Dijkstrin algoritam.

Udaljenosti su neophodne da bi se za svaki neoznačeni čvor pronašli najbliži označeni čvorovi.

5. Dopunsko označavanje čvorova

Nakon inicijalnog označavanja, preostali čvorovi se dodeljuju jednoj od oznaka (0 ili 1) pomoću algoritma **širenja oznaka** (Label Propagation).

Postoje dva režima rada:

Bez propagacije – koriste se isključivo inicijalno označeni čvorovi. Novooznačeni čvorovi se ignorišu u daljem procesu.

Sa propagacijom – čim čvor dobije oznaku, koristi se u narednim koracima za označavanje drugih čvorova.

U oba pristupa, za svaki neoznačeni čvor posmatra se k najbližih označenih čvorova (k ∈ {5, 10, 15, 20, 30, 50, 100, 200}). Čvor dobija onu oznaku koja je među njima zastupljenija.

4. REZULTATI

Broj čvorova	Gustina	K	Mod	Vrijeme
1000	0.3	200	1	0.003000
1000	0.3	200	2	0.005000
1000	0.5	200	1	0.013000
1000	0.5	200	2	0.018000
1000	0.7	200	1	0.029000
1000	0.7	200	2	0.038000
5000	0.3	200	1	0.535000
5000	0.3	200	2	0.766000
5000	0.5	200	1	0.926000
5000	0.5	200	2	1.295000
5000	0.7	200	1	1.283000
5000	0.7	200	2	1.822000
10000	0.3	200	1	2.271000
10000	0.3	200	2	3.268000
10000	0.5	200	1	4.086000
10000	0.5	200	2	5.737000
10000	0.7	200	1	6.146000
10000	0.7	200	2	8.596000

Uticaj parametra K na vrijeme izvrašavanja:

Broj čvorova	Gustina	K	Mod	Vrijeme
10000	0.7	5	1	2.286000
10000	0.7	5	2	3.213000
10000	0.7	50	1	6.085000
10000	0.7	50	2	8.582000
10000	0.7	200	1	6.146000
10000	0.7	200	2	8.596000

5. ZAKLJUČAK

Mod 1 je brži u svim slučajevima,broj iteracija je minimalan sve se riješava u jednom prolazu kroz čvorove.Vrijeme se povećava linearno sa brojem čvorova i gustinom.Parametar K nema veliki uticaj na mod 1.Loša strana ovog načina je nepreciznost.Algoritam ne zalazi u dubinu i ne provjerava više puta susjede čvorova.Bolji način kada je bitnija brzina.

Mod 2 je sporiji u svim slučajevima,ali detaljniji i precizniji ima više iteracija što je dobro za složenije strukture.Koristi heap strukturu i više puta ubacuje ili izbacuje čvorove iz heap-a. Vrijeme se povećava značajno više u odnosu na mod 1.Parametar K ima veliki uticaj na mod 2 jer mora da provjeri više susjeda više puta. Bolji način kada je bitnija tačnost.

6. LITERATURA

https://www.geeksforgeeks.org/dsa/detect-negative-cycle-graph-bellman-ford/

https://www.geeksforgeeks.org/dsa/dijkstras-shortest-path-algorithm-greedy-algo-7/

https://www.youtube.com/watch?v=nvRkFi8rbOM

https://stackoverflow.com/questions/51799313/whats-the-difference-between-modified-dijkstra-with-single-source-single-desti

https://chatgpt.com/

Github link:

https://github.com/Srdjan101/OP2/tree/main/Projekt