UPOTREBA GENETSKOG ALGORITMA ZA REŠAVANJE PROBLEMA MINIMAL BANDWIDTH

Nikola Lazarevic 267/2017
mi17267@alas.matf.bg.ac.rs

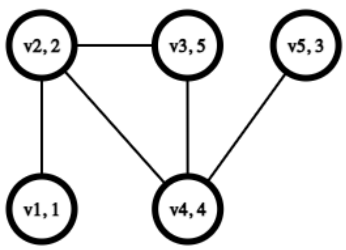
September 29, 2023

Sadržaj

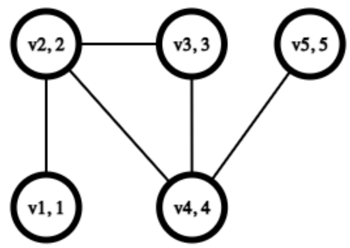
1 Formulacija problema

Neka je dat neusmeren graf $G(V, E)$, gde je V skup čvorova, a E skup grana grafa G. Potrebno je pronaći linearno uređenje čvorova skupa V to jest injektivnu funkciju f definisanu na sledeći način $f : V \rightarrow \{1, 2, ..., |V|\}$ tako da vrednost funkcije $max_{(u,v) \in E} |f(u) - f(v)|$ bude minimalna.

Primer. Neka je dat graf $G(V, E)$ i neka je dato uređenje $f : f(v_1) = 1, f(v_2) = 2, f(v_3) = 5, f(v_4) = 4, f(v_5) = 3$, pri čemu važi da je $|V| = 5$.



(a) Uređenje f grafa G



(b) Uređenje f' grafa G

Protok pojedinačnih čvorova grafa G, na osnovu uređenja f, računamo na sledeći način:

$$B_f(v_1) = \max\{|1 - 2|\} = 1$$
$$B_f(v_2) = \max\{|2 - 1|, |2 - 5|\} = 3$$
$$B_f(v_3) = \max\{|5 - 2|, |5 - 4|\} = 3$$
$$B_f(v_4) = \max\{|4 - 2|, |4 - 5|, |4 - 3|\} = 2$$
$$B_f(v_5) = \max\{|3 - 4|\} = 1$$

Protok grafa G na osnovu uređenja f, Slika (a), je:

$$B_f(G) = \max_{v \in V} B_f(v) = \max\{1, 3, 3, 2, 1\} = 3$$

Ukoliko izvršimo zamenu oznaka čvora v_3 i čvora v_4 dobijamo novo uređenje čvorova f'.

Ponovo, protok pojedinačnih čvorova grafa G, koristeći novo uređenje f', računamo kao:

$$
\begin{aligned}
B_f(v_1) &= \max\{|1-2|\} = 1 \\
B_f(v_2) &= \max\{|2-1|, |2-3|\} = 1 \\
B_f(v_3) &= \max\{|3-2|, |3-4|\} = 1 \\
B_f(v_4) &= \max\{|4-2|, |4-3|, |4-5|\} = 2 \\
B_f(v_5) &= \max\{|5-4|\} = 1
\end{aligned}
$$

Protok grafa G na osnovu uređenja f', Slika (b), je:

$$B_f(G) = \max_{v \in V} B_f(v) = \max\{1, 1, 1, 2, 1\} = 2$$

Na osnovu prethodnog primera možemo primetiti da se rotacijom oznaka čvorova a a u okviru uređenja f dobija uređenje f' grafa G i važi da je protok grafa G manji pri uređenju f', pri kom je protok grafa G niži od protoka u slučaju uređenja f.

Takođe, potrebno je dodati da se problem određivanja minimalnog protoka grafa može formulisati i u terminima matrice. Naime, svaki graf $G(V, E)$ moguće je predstaviti u vidu matrice susedstva. Matrica susedstva grafa G dimenzija je $|V|x|V|$, pri čemu je vrednost matrice na poziciji $(i, j) = 1$ ukoliko postoji između čvorova i i j u suprotnom je 0. Ono što je karakteristika ovih matrica je u velikom broju slučajeva reč o retkim matricama, tj. matricama sa velikim brojem 0.

2 Gruba Sila

Algoritam grube sile podrazumeva da se ispitaju sva moguća uređenja zadatog grafa i da se kao rešenje vrati ono uređenje koje daje minimalni protok grafa.

Uređenje grafa može se predstaviti u vidu permutacije bez ponavljanja brojeva iz intervala $[1..|V|]$ u dužini $|V|$, gde je V skup čvorova posmatranog grafa.

Implementacija ovog algoritma podrazumeva sekvencijalno generisanje permutacija i računanje protoka grafa za svaku permutaciju pojedinačno.

Broj mogućih uređenja zadatog grafa iznosi $|V|!$.

Prethodno navedeni algoritam eksponencijalne je složenosti pa se iz tog razloga on koristi samo u demonstrativne svrhe to jest za proveru ispravnosti drugih metoda na grafovima manjih dimenzija. Ukoliko se izvršavanje algoritma privede kraju, pronalazak optimalnog rešenja je zagarantovan.

3 Lokalna pretraga

Algoritam lokalne pretrage, kao osnovni pristup za rešavanje problema optimizacije, polazi od početnog rešenja, obično nasumično izabranog. U svakoj iteraciji ovog algoritma, vršimo manje promene ili modifikacije na trenutnom rešenju. Ako se nakon ovih promena dobije rešenje koje je bolje u smislu ciljnog kriterijuma, tada to poboljšano rešenje prihvatamo i prelazimo na narednu iteraciju. U suprotnom, ukoliko izmene ne rezultiraju boljim rešenjem, zadržavamo trenutno rešenje i prelazimo na sledeću iteraciju.

Ovaj proces se ponavlja dok se ne dostigne određeni uslov zaustavljanja, kao što može biti maksimalni broj iteracija ili nedostizanje boljeg rešenja u zadatom vremenskom okviru.

Neka je dat graf $G(V, E)$. Uređenje čvorova zadatog grafa G, kao i u slučaju grube sile, biće predstavljeno permutacijom bez ponavljanja brojeva iz intervala $[1..|V|]$ u dužini $|V|$.

Neka je v proizvoljan čvor grafa G i neka je $N(v)$ skup svih suseda čvora v u grafu G. Tada možemo definisati minimalnu i maksimalnu oznaku čvorova susednih čvoru v na sledeći način:

$$min(v) = min\{f(u) : u \in N(v)\}$$
$$max(v) = max\{f(u) : u \in N(v)\}$$

Sa $N(v)$ označavamo skup svih suseda čvora v u grafu G.

Najbolje obeležje za čvor v u okviru trenutnog uređenja f predstavlja:

$$mid(v) = \frac{min(v) + max(v)}{2}$$

Odatlje definišemo skup svih čvorova $S(v)$ koji su pogodni za zamenu sa čvorom v na sledeći način:

$$S(v) = \{u : |mid(v) - f(u)| < |mid(v) < f(v)|\}$$

Neka je $B_f(G)$ protok grafa G za uređenje f, tada svaki čvor v skupa V za koji važi da je $B_f(G) = B_f(v)$ predtsavlja kritičan čvor.

U slučaju problema pronalaska minimalnog protoka grafa lokalnu pretragu možemo unaprediti na sledeći način. U svakoj itearciji vršimo identifiovanje svih kritičnih čvorova trenutnog uređenja. Pronađenje kritične čvorove obilazimo sekvencijalno i za svaki od njih određujemo listu pogodnih zamena (dobijenu listu potrebno je sortirati rastuće kako prvo bile ispitane bolje opcije). Nakon zamene dobija se novo uređenje na osnovu kog se računa protok grafa.

Smatramo da je rešenje bolje ukoliko je nakon zamene protok grafa ili broj kritičnih čvorova manji.

Prethodno opisani proces se ponavlja dok trenutno rešenje ne dostigne maksimalno moguće unapređenja ili dok algoritam ne dosegne unapred određeni broj iteracija.

Algoritam lokalne pretrage idejno je vrlo jednostavan i lak za implementiranje. Ovakva metoda se često koristi za lokalnu optimizaciju, gde se fokusira na pretragu okoline trenutnog rešenja, ali ima ograničenja u pronalaženju globalno najboljeg rešenja, s obzirom na svoj lokalni karakter pretrage..

4 Genetski algoritam

Genetski algoritam pripada grupi evolutivnih algoritama, koji simuliraju proces evolucije inspirisan prirodnom selekcijom. U ovom procesu, svaka jedinka se takmiči sa ostalim jedinkama u cilju preživljavanja, pri čemu bolje prilagođene jedinke imaju veće šanse za opstanak i reprodukciju. U slučaju genetskog algoritma, jedinke u populaciji predstavljaju potencijalna rešenja određenog problema. Tokom vremena, populacija evoluira ka sve boljim rešenjima.

Osnovne komponente genetskog algoritma su: reprezentacija, populacija, selekcija, ukrštanje i mutacija.

Reprezentacija. U evolutivnim algoritmima, rešenja su reprezentovana kao hromozomi, pri čemu svaki hromozom sadrži gene koji predstavljaju karakteristike tog rešenja. Odabir odgovarajućeg načina kodiranja je ključno za rešavanje problema

Rešenje ćemo predstaviti u vidu niza dimenzije $|V|$. Svakom element niza predstavlja jedan čvor grafa, a njegova vrednost je oznaka datog čvora. Takođe sve vrednosti u okviru niza su različite i pripadaju intervalu $[1..|V|]$.

Inicijalna populacija. Inicijalnu populaciju generišemo nasumično, nakon čega primenjujemo lokalnu pretragu na određen procenat populacije.

Fitnes funkcija. Funkcija fitnes koristi se za utvrđivanje kvaliteta jedinke, odnosno koliko je jedinka dobro prilagođena.

Kao fitnes funkciju koristićemo protok grafa. Jedinka sa manjim protokom predstavlja prilagođeniju jedinku.

Selekcija. Selekcija podrazumeva odabir jedinki koje će biti uključene u proces ukrštanja i imati potomstvo.

Selekciju vršimo po modelu turnira. Ona podrazumeva da se na slučajan način izabere k jedinki, a potom se od tih k jedinki bira ona sa najboljim fitnesom.

Takođe koristićemo generacijski model sa elitizmom. On podrazumeva da po završetku iteracije genetskog algoritma staru populaciju u potpunosti menjamo novom. Takođe na početku svake iteracije u novu populaciju iz prethodne prenosi se određen broj najboljih jedinki.

Ukrštanje. Proces ukrštanja podrazumeva spajanje dva roditelja u cilju generisanje jednog ili dva deteta.

Ukrštanje vršimo tako što na slučajan način biramo poziciju k u okviru hromozoma. Svi elementi prvog roditelja do pozicije k kopiraju se u prvo dete do pozicije k dok se svi elementi drugog roditelja do pozicije k kopiraju u drugo dete. do pozicije k. Potom se vrši popunjavanje preostalih pozicija prvog deteta oznakama koje nedostaju ali u redostledu u kom se te oznake nalaze u drugom roditelji, slično važi i za drugo dete.

Mutacija. Mutacija ima svoju eksplorativnu funkciju i predstavlja slučajnu promenu vrednosti gena. Svaka jedinka se podvrgava mutaciji s niskom verovatnoćom. Mutacija donosi mogućnost unošenja potpuno novih informacija u populaciju, što se ne može postići isključivo putem ukrštanja. Mutacija takođe doprinosi raznovrsnosti populacije, pomažući algoritmu da istražuje šire područje potencijalnih rešenja.

Mutaciju vršimo tako što na slučajan način biramo broj rotacija, k, koje ćemo izvršiti, a potom, za svako k na slučajan način biramo po dva gena u okviru hromozoma koja će biti rotirana.

Kriterijum zastavljanja. Algoritam se zaustavlja nakon što je dostignut prethodno postavljeni broj generacija.

5 Rezultati

Genetski algoritam primenjen je na grafove prikazane u narednoj tabeli u konfiguraciji koja podrazumeva veličinu populacije od30 jedinki, 20 iteracija sa verovatnoćom mutacije 0.002. Lokalna pretraga koja se koristi za optimizaciju jedinki populacije izvršava se u najviše 15 iteracija.

Test results						
Graph name	Optimal	Node numbers	LS	LS time	GA	GA time
bcspwr01	5	39	10	0.0461	6	6.4525
will57	7	57	19	0.0994	9	6.5747
dwt59	6	59	13	0.1392	5	8.3633
dwt66	3	66	12	0.1397	5	8.9059
dwt72	6	72	14	0.1094	10	5.6851
ash85	9	85	24	0.6702	17	10.2593
nos4	10	100	36	1.1588	11	13.1337
west0132	33	132	52	3.49	48	19.6195
will199	66	185	74	6.8948	49	23.6918
gre___185	21	185	69	12.0524	57	21.3249

6 Zaključak

Za potrebe rešavanja problema pronalaženja uređenja čvorova zadatog grafa sa minimalnim protokom razvijani su algoritam grube pretrage, alogirtam lokalne pretrage i genetski algoritam.

Genetski algoritam koji koristi optimizaciju u vidu lokalne pretrage koja se primenjuje na jedinke populacije, pokazuje dobre rezultate na grafovima male i srednje veličine. Sa druge strane običan genetski algoritam svoje slabosti ispoljava već ma grafovima srednje veličine.

Prethodno opisan genetski algoritam dodatno se može unaprediti tako što će se u inicijalnu populaciju dodati jedinke koje su dobijene primenom pretrage u širinu iz različitih čvorova zadatog grafa.