Rešavanje problema maksimalne klike

Đorđe Stanojević, Dalma Beara

Februar 2019

Contents

1 Uvod

Neusmereni graf čine konačan skup čvorova i skup neuređenih parova tih čvorova koji se nazivaju ivice. Klika u nekom neusmerenom grafu G je drugi naziv za neki njegov kompletan podgraf, tj. podgraf u kom su svi čvorovi međusobno povrzani. Termin kompletni podgraf prvi put se pojavljuje u radu ,,Kombinatorni problem u geometriji" Erdoša i Sekereša 1935. godine, dok se termin klika pojavio oko 1949. godine u kontekstu modelovanja socijalnih klika (grupa ljudi koji svi poznaju jedni druge) kao kompletnih podgrafa. Prvi algoritam za nalaženje klike osmislili su Harari i Ros 1957. Složenost ovog problema razmatrana je dugo, od 1977, da bi se u 90-im došlo do toga da se on čak ni ne može precizno i efikasno aproksimirati. Tačnije, danas je poznato da se za svako ¿ 0 ne može napraviti bolji algoritam koji bolje aproksimira rešenje našeg problema nego u složenosti O(n^{1-}) (pod pretpostavkom da P nije jednako NP).

U kontekstu engleskih naziva maximal i maximum, razlikujemo dve definicije ovog problema. Termin ,,maximal clique" podrazumeva kliku čiji čvorovi nisu deo neke druge, veće klike, dok se ,,maximum clique" odnosi na najveću kliku u jednom grafu, i to je problem koji ćemo mi ovde rešavati. Postoji još nekoliko varijanti ovog problema, a one su:

- klika sa težinama – traži se kompletan podgraf koji ima najveći zbir težina (za ovo, naravno, graf mora biti težinski)
- k-klika – klika u kojoj učestvuje tačno k čvorova
- odlučivanje klike – odgovor na pitanje da li graf sadrži kliku veličine k
- izlistavanje svih maksimalnih (engl. maximal) klika u datom grafu

Problem koji rešavamo ima brojne primene, a najrasprostranjenije su u:

- bioinformatici – predviđanje strukture proteina, evoluciona drveta,
- kompjuterskoj hemiji – nalaženje hemikalija željene strukture, predviđanje preferirane orijentacije jednog molekula u drugom kad su vezani jedan za drugi
- automatskom generisanju test uzoraka – određivanje veličine test skupova
- sociologiji – socijalne mreže
- slučajnim procesima – nalaženje kliki u grafovima zavisnosti

Interesantno je da je ovaj problem komplementaran problemu maksimalnog nezavisnog skupa. Klika grafa G je nezavisan skup grafa komplementnog grafu G i obrnuto. Međutim, složenost ovih dvaju algoritama se razlikuje za neke tipove grafova. Na primer, maksimalna klika na ravanskom grafu se može naći u polinomijalnom vremenu (jer je poznato da su oni svi najviše 4-obojivi), dok je nalaženje maksimalnog nezavisnog skupa na takvim grafovima NP-težak problem. Maksimalna klika na savršenom grafu (to su oni čiji je hromatski broj jednak baš veličini njegove maksimalne klike) nalazi se u polinomijalnom vremenu.

Navedimo sada nekoliko poznatih algoritama koji rešavaju ovaj problem i njihove složenosti:

- Bron-Kerboš (1973.) – rešava problem najveće (engl. maximal) klike rekurzivni bektreking, složenost O($3^{n/3}$), izvedena iz toga da graf od n čvorova može imati najviše 3n-3 najvećih klika
- Tarjan i Trojanovski (1977.) – slično kao prethodni ali sa odsecanjem, složenost O($2^{n/3}$) = O(1.2599^n)
- Žian (1986.) – O($2^{0.304n}$) = O(1.2346^n)

- Robson (1986.) – $O(2^{0.276n}) = O(1.2108^n)$ kombinuje bektreking sa dinamičkim programiranjem
- Robson (2001.) – $O(2^{0.249n}) = O(1.1888^n)$, najbrži algoritam danas

Problem odlučivosti klike je NP-kompletan (može se svesti na problem zadovoljivosti logičkih formula) i spominje se u radu Stivena Kuka u kom je prvi put spomenuta NP-kompletnost. On se takođe može svesti na problem odlučivosti, pa je i NP-težak.

2 Opis rešenja zadatog problema

Cilj ovog projekta jeste prikaz nalaženja maksimalne klike u zadatom grafu korišćenjem genetskog algoritma. Genetski algoritam, kao primer evolutivnog istraživanja, predstavlja računarsku simulaciju u kojoj populacija apstraktno opisanih jedinki, koje su kandidati za rešenje problema, treba da se približava boljim rešenjima. Razvijen je u Americi 1970-ih godina, a ključnim autorom navodi se Dž.Holand. Osnovni elementi opštog genetskog algoritma:

- hromozom/genotip - reprezentacija jedinke
- funkcija prilagođenosti/fitnes funkcija - ocena kvaliteta jedinke
- selekcije - proces odabira jedinki koje će biti izabrane za stvaranje novih jedinki
- operator ukrštanja - proces dobijanja jedne ili više jedinki od dve prethodno izabrane
- operator mutacije - proces modifikovanja polazne jedinke koji simulira uticaj spoljašnjih faktora na jedinku
- kriterijum zaustavljanja

Opšti genetski algoritam

```
Ulaz: Graf zadat čvorovima u json formatu
Izlaz: Najkvalitetnija jedinka zadatog grafa i njena ocena

generiši početnu populaciju jedinki
izračunaj prilagođenost svake jedinke u populaciji
ponavljaj
    izaberi iz populacije skup jedinki za reprodukciju
    kreiraj nove jedinke primenom operatora ukrštanja i mutacije (i izračunaj njihovu prilag
    kreiraj novu generaciju
dok nije ispunjen kriterijum zaustavljanja
vrati najkvalitetniju jedinku u poslednjoj populaciji
```

Veći deo načina implementacije autori su radili po uzoru na master rad B.Huanga, pri čemu je sama implementacija rad samih autora. Elementi opštog genetskog algoritma kao i dodatni proces lokalne optimizacije implementirani su na sledeći način:

2.1 Reprezentacija jedinke

Kako je svako rešenje problema maksimalnih klika jedan podgraf G' zadatog grafa G, prirodan način za kodiranje hromozoma jeste binarni niz dužine n (gde je n broj čvorova) . Jedinica na poziciji i u ovom nizu označava pripadnost čvora i podgrafu G'. Napomena: Svaki hromozom ne mora nužno predstavljati kliku.

2.2 Kreiranje inicijalne populacije

Početna populacija nastaje primenom pohlepnog algoritma i to na sledeći način.
Odabrati nasumičan čvor v$_i idodatigapodskupuA, zatimnasumičnoodabratičvorv_j odsusedačvorav_i, označitičv$

2.3 Selekcija

U svakoj generaciji primenjuje se ruletska selekcija. Ruletska selekcija podrazumeva odabir jedinki slučajnim putem gde je verovatnoća izbora svake jedinke proporcionalna vrednosti njene fitnes funkcije.

2.4 Fitnes funkcija

Kako hromozom mora biti klika u trenutku izračunavanja fitnes funkcije, prirodno se nameće veličina klike (broj jedinica u hromozomu) kao ocena kvaliteta.

2.5 Operator ukrštanja

Primenjivana politika ukrštanja je višepoziciono ukrštanje. Broj tačaka preseka se menja od početnih deset (kada je to moguće) do krajnje dve uz smanjenje nakon svakih 20 iteracija.

2.6 Operator mutacije

Nakon što su dva novonastala hromozoma izabrana za mutaciju, na red stupa promena njihovog proizvoljnog gena u skladu sa unapred yadatom verovatnoćom.

2.7 Lokalna optimizacija

Ovaj korak sledi ukrštanju i mutaciji i ima značajnu ulogu u ubrzanju konvergencije ovog algoritma ka najboljem rešenju. Čine ga dve faze:

1. Izdvajanja klike iz hromozoma
2. Proširivanja klike

2.8 Kriterijum zaustavljanja

Algoritam se zaustavlja ukoliko je dostignut unapred zadat broj iteracija ili ukoliko nema napretka u oceni najbolje jedinke u s generacija, gde je s parametar stagnacije.

2.8.1 Izdvajanje klike iz hromozoma

Primenjuje pohlepni algoritam sa malom izmenom kod uklanjanja čvorova. Iz svakog potomka se izvlači niz čvorova sa najnižim i drugim najnižim stepenom, nakon čega se jedan od njih uklanja, a stepeni ostalih se ažuriraju u skladu sa tim i tako sve dok se ne dođe do hromozoma koji predstavlja kliku. Odstupanje od pohlepnog algoritma upravo predstavlja slučajnost izbora čvora a ne konstatan odabir onoga sa najmanjim stepenom.

2.8.2 Proširivanje klike

Ovaj algoritam nastoji da poveća veličinu klike dodavanjem novih čvorova u hromozom. Nakon dodavanja, proverava se da li je njihovim uključenjem klika prestala biti klika i ukoliko jeste taj dodati čvor se uklanja.

2.9 Drugi algoritam nalaženja

Kako bi bilo moguće porediti rezultate i vremena izračunavanja, implementiran je i algoritam Karagana i Pardalosa.

```
funkcija klika( U , veličina )
    ako je | U | = 0
        ako je veličina > max
            max = veličina
            Čuvanje novog rekorda
        vrati se

    dok je |U| != 0
        ako je veličina + |U| < max
            vrati se
        i = min{ j | v_j E U}
        U = U \ {v_i}
        klika( presek( U,N(v_i),veličina + 1)
    vrati se
```

3 Upoređivanje sa drugim rešenjima

Za potrebe testiranja implementiran je generator grafova koji proizvodi ulazne .json datoteke. Testirani grafovi imaju približno 50 čvorova. U sledećim tabelama nalaze se rezultati i vremena izvršavanja dobijena genetskim algoritmom i algoritmom Karagana i Pardalosa. Sva vremena data su u mikrosekundama.

Sledeći rezultati dobijeni su na sistemu sa sledećim specifikacijama: Intel Core i7-8750H, 2.20GHz $\times$ 12, 15.3GB RAM na operativnom sistemu Ubuntu 18.04

	Rešenje_g	Vreme_g	Rešenje_kp	Vreme_kp
Input16.json	3	253247	3	606
Input17.json	3	392379	3	186
Input18.json	3	392379	3	169
Input19.json	3	392379	3	168
Input20.json	3	392379	3	584
Input21.json	2	398557	3	206
Input22.json	3	418928	3	177
Input23.json	3	418928	3	446
Input24.json	3	436056	3	177

Sledeća tabela prikazuje rezultate dobijene na sistemu sa sledećim specifikacijama: Intel Core i3 , 2.4GHz x 4, 6.00GB RAM na operativnom sistemu Ubuntu 16.4.

	Rešenje_g	Vreme_g	Rešenje_kp	Vreme_kp
Input16.json	3	515607	3	482
Input17.json	2	407302	3	482
Input18.json	3	362203	3	395
Input19.json	3	277970	3	490
Input20.json	3	480969	3	402
Input21.json	3	245169	3	417
Input22.json	2	138161	3	463
Input23.json	2	74231	3	371
Input24.json	3	511127	3	424

4 Zaključak

U ovom radu bilo je razmatrano nalaženje problema najveće klike u zadatom grafu pomoću genetskog algoritma, i uprkos tome što su korišćeni parametri i vrednosti za koje je B. Huang empirijski utvrdio da su kvalitetni, algoritam Karagana i Pardalosa je davao bolja rešenja uz manje vreme izvršavanja. Ovakav epilog se mogao očekivati imajući u vidu da je taj algoritam, mada jednostavan, među najbržima kada su u pitanju retki grafovi. Svakako, složenost ovog genetskog algoritma moguće je popraviti, što je u budućnosti cilj.

References

[1] A fast algorithm for the maximum clique problem, Patric R. J. Östergård

[2] Finding Maximum Clique with a Genetic Algorithm, Bo Huang,The Pennsylvania State University,The Graduate School: Capital College, 2002

[3] Veštačka inteligencija, Predrag Janicic, Mladen Nikolic, 2018

[4] The Maximum Clique Problem, Panos M. Pardalos, Jue Xue, 1992.