

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 656

Bojanje grafova prilagodljivim metaheurističkim postupcima

Dino Šantl

Zagreb, svibanj 2014.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

SADRŽAJ

1. Uvod	1
1.1. Sažetci korištenih znanstvenih članaka	1
2. Formalni opis problema	2
2.1. Tehnički opis problema	2
2.1.1. Ulazni i izlazni podaci	3
2.2. Matematički opis problema	3
2.2.1. Definicije za klasičan problem bojanja grafova	4
2.2.2. Bojanje težinskih grafova	8
2.3. Analiza problema bojanja težinskih grafova	10
2.3.1. Prostor pretrage stanja	10
2.3.2. Pozadina problema iz perspektive računske teorije složenosti .	14
2.3.3. Posljedica " <i>No free lunch</i> " teorema	15
3. Algoritmi	17
4. Infrastruktura	18
5. Testiranje	19
6. Rezultati	20
7. Zaključak	21
Literatura	22

1. Uvod

Uvod rada. Nakon uvoda dolaze poglavlja u kojima se obrađuje tema.

1.1. Sažetci korištenih znanstvenih članaka

2. Formalni opis problema

U ovom poglavlju opisuje se problem bojanja grafova i njegova primjena u telekomunikacijskom problemu koji se rješava. Najprije se predstavlja problem u telekomunikacijama uz pretpostavke i ograničenja. Zatim se isti problem opisuje pomoću matematičkog modela tj. problema bojanja grafova. Prvo se objašnjava općeniti problem bojanja grafova, a zatim se matematički model prilagodi konkretnom problemu.

2.1. Tehnički opis problema

Za shvaćanje problema dovoljno je pretpostaviti da postoji korisnikova oprema (što je najčešće mobilni uređaj) i bazna stanica. Bazne stanice geografski su statične, za razliku od mobilnih uređaja. Ono što se u ovom radu naziva baznom stanicom je radio pristupna mreža (engl. *Radio Access Network*) (*RAN*). *RAN* dio je telekomunikacijskog sustava i nalazi se između opreme korisnika i jezgre mreže. To je sloj u mreži koji je zaslužan za prenošenje komunikacije između mobilnog uređaja do antene i od antene do drugog mobilnog uređaja.

Za što kvalitetniji rad mreže potrebno je optimirati parametre *RAN*-a. Jedan od parametra je *scrambling* kôd. *Scrambling* kôd međuostalim služi kako bi mobilni uređaj mogao razlikovati bazne stanice. Zbog toga je potrebno svakoj baznoj stanici pridružiti različit kôd. Problem je u tome što je dostupno samo 512 različitih kodova. To znači da neke bazne stanice moraju imati isti kôd (ako se u sustavu nalazi više od 512 baznih stanica). Ako neke dvije bazne stanice imaju isti kôd i mobilni uređaj se nalazi u doseg signala od obje stanice, tada ne zna s kojom baznom stanicom komunicira te pada kvaliteta usluge.

Potrebno je dodjeliti *scrambling* kôdove tako da dvije bazne stanice koje mogu biti istovremeno vezane za jedan mobilni uređaj nemaju isti kôd ili ako je to nemoguće smanjiti, dodjeliti kôdove tako da je negativan utjecaj na kvalitetu usluge što manji. Postoje još neka tehnička ograničenja koja je potrebno uvažiti, a biti će navedena u nastavku.

Iako postoji 512 različitih kôdova, svaka bazna stanica ima ograničen skup kôdova koje može koristiti. Dakle, za svaku baznu stanicu zadan je skup kôdova. Za neke bazne stanice postoji svojstvo nepromjenjivost, što znači da kôd koji je trenutno zadan za tu baznu stanicu mora ostati takav. Takve se vrste baznih stanica nazivaju **nepromjenjivima**. Početno stanje mreže definirano je kôdovima koji su pridjeljeni nekoj baznoj stanici. Osim toga za svaku baznu stanicu poznata je njezina vrsta. Vrsta može biti označena sa slovima: *A*, *B* ili *C*. Bazne stanice različitih vrsta ne utječu jedna na drugu.

2.1.1. Ulazni i izlazni podaci

Kao ulaz algoritam dobije inicijalno stanje mreže. Odnosno za svaku baznu stanicu poznat je trenutni kôd koji koristi i sva pravila koja moraju biti zadana. Izlaz algoritma je popis oznaka baznih stanica i za svaku od oznaka pripadajući kôd.

Ulazni podaci

1. Popis i definicija domene boja (skupova boja)
2. Popis čvorova (baznih stanica)
 - (a) Oznaka čvora
 - (b) Vrsta čvora (grupa) - *A*, *B* ili *C*
 - (c) Oznaka domene za boju čvora (koji skup boja koristi)
 - (d) Početna boja za čvor
 - (e) Oznaka da li je čvor nepromjenjiv

Izlazni podaci

Kao izlaz algoritma koristi se par brojeva (i, c) gdje je i oznaka za čvor, a c je boja čvora.

2.2. Matematički opis problema

Potrebno je modelirati problem u kojem postoji *bazna stanica* i veze između istih, koje predstavljaju mjeru u kojoj jedna bazna stanica utječe na drugu. Jedan od mogućih modela je graf. Čvorovi će predstavljati bazne stanice, a jakost između dvije bazne

stanice biti će modelirano pomoću težine brida. U nastavku bazna stanica nazivat će se čvorom. Nad ovako postavljenim grafom problem je dodjeliti kôdove tako da ako je moguće nema bridova koji na svojim krajevima imaju čvor s istim kôdom. Problem je poznat pod nazivom *bojanje grafa*. Kako je uobičajeno pričati o bojama čvora (a ne o kôdovima) od sad pa nadalje *scrambling* kôd nazivat će se bojom čvora. U nastavku će prvo biti opisan klasičan problem bojanja grafova, gdje se promatraju beztežinski grafovi (oni čiji bridovi nemaju težine). Nakon toga problem se poopćuje na težinske grafove, čime se modelira predhodno opisan problem.

2.2.1. Definicije za klasičan problem bojanja grafova

Najprije se definiraju matematički pojmovi. Nakon toga formalno je opisan problem bojanja grafova. Uz to se nadovezuje teorija izračunljivosti.

Pojam grafa

Definicija 1. Jednostavni graf G sastoji se od nepraznog konačnog skupa $V(G)$, čije elemente nazivamo čvorovi grafa G i konačnog skupa $E(G)$ različitih dvočlanih podskupova $V(G)$ koje zovemo bridovi.

Definicija 2. Skup vrhova koji su susjedni vrhu v zovemo susjedstvo vrha v i označavamo s oznakom $H(v)$.

Definicija 3. Stupanj vrha v grafa G jednak je broju bridova koji su incidentni s v . Označavamo ga s $d(v)$.

Definicija 4. Neka je G jednostavan graf i ω funkcija $\omega : E \rightarrow \mathbb{R}$. Par (G, ω) naziva se težinski graf. Pri čemu funkcija ω svakom bridu iz G dodjeljuje jedan element iz skupa realnih brojeva. Neka je e neki brid grafa G , njegova težina označena je s $\omega(e)$.

Definicija bojanja grafova

Definicija 5. Definiramo funkciju $\phi : V(G) \rightarrow \mathbb{N}$, koja svakom čvoru u grafu pridružuje jedan prirodan broj koji $\phi(v)$, gdje je v čvor u grafu G . Broj $\phi(v)$ nazivamo boja čvora, a funkciju ϕ nazivamo *bojanje grafa*.

Definicija 6. Bojanje grafa s najviše k boja nazivamo *k-bojanje grafa*.

Definicija 7. Ako se graf može obojati s najviše k boja tada takvo bojanje nazivamo *legalno k-bojanje grafa*.

Definicija 8. Graf je **k-obojev** ako postoji legalno k-bojanje grafa.

Definicija 9. Ako je graf G **k-obojev**, ali nije **(k-1)-obojev** tada kažemo da je **k** kromatski broj grafa G , gdje se koristi oznaka $\chi(G) = k$

Definicija 10. Podskup skupa $V(G)$ nazivamo **nezavisni skup** ako u njemu ne postoje dva čvora koja su susjedna.

Definicija 11. Brid koji spaja dva čvora iste boje nazivamo **konfliktni brid**.

Definicija 12. Dva čvora koja spaja konfliktni brid nazivamo **konfliktnim čvorovima**.

Definicija 13. Particiju skupa $V(G)$ na k disjunktnih nepraznih podskupova V_1, \dots, V_k tako da vrijedi $V(G) = \bigcup_{j=1}^k V_j$ zovemo **k-dioba** grafa G . Ako su podskupovi V_1, \dots, V_k ujedno i nezavisni skupovi onda se to naziva **legalna k-dioba** grafa G .

Teorem 1. Graf G je **k-obojev** ako postoji **legalna k-dioba** grafa G .

Dokaz. Pretpostavimo da je graf G **k-obojev**. Definiramo skupove S_i tako da čvor grafa v pripada skupu S_i ako je obojan bojom i . Takvi skupovi su neprazni i disjunktni, te unija skupova S_i čini skup svih vrhova grafa G . Kako je G **k-obojev** tada ne postoji boja zbog koje bi neki brid bio konfliktni pa slijedi da je podjela na skupove S_i **legalna k-dioba** grafa G jer su skupovi S_i nezavisni.

Drugi smjer dokazuje se tako što se pretpostavi da postoji **legalna k-dioba** grafa G na skupove S_i . Kako je svaki skup indeksiran s indeksom i , tada svakom čvoru koji se nalazi u skupu S_i dodjelimo boju i . Kako su skupovi S_i prema pretpostavci disjunktni, neprazni i nezavisni tada ne postoje dva čvora koja bi imala istu boju. Q.E.D \square

Definicije računske teorije složenosti

Definicija 14. **Problem odluke** je problem koji uvijek ima odgovor **da** ili **ne**. Primjerice, problem da li je graf točno **k-obojev** je problem odluke. Problem traženja kromatskog broja grafa nije problem odluke, jer tražimo točan broj **k**.

Definicija 15. Problem odluke za koje postoje algoritmi koji daju odgovor, a čije vrijeme izvršavanje ovisi polinomno o veličini ulaznih podataka spadaju u **klasu P**.

Definicija 16. Problem odluke spada u **klasu NP problema** ako se točnost njegovog rješenja može ispitati u polinomnom vremenu.

Definicija 17. Problem odluke spada u **klasu NP-potpunih problema** ako spada u **klasu NP problema** i koji ima svojstvo da se svaki drugi problem iz **klase NP** može polinomno reducirati na njega.

Definicija 18. Kažemo da je problem **NP-težak** akko postoji **NP-potpun problem** koji se može polinomno reducirati na njega.

Problem odluke za koji postoji algoritam čije trajanje ovisi polinomno o veličini ulaznih podataka tada je to **P problem**. Ako za neki problem odluke ne možemo pronaći algoritam čije vrijeme izvođenja ovisi polinomno o veličini ulaznih podataka, ali točnost rješenja možemo provjeriti u polinomnom vremenu tada je to **NP problem**. **NP-potpun problem** je problem odluke čiji algoritam možemo iskoristiti da bi riješili sve **NP probleme** tako da koristimo polinoman broj poziva tog algoritma. Općeniti problemi (koji ne moraju biti problemi odluke) nazivaju se **NP-teški problemi** ako postoji problem u klasi **NP-potpunih problema** koji se može riješiti pomoću polinomnog broja poziva algoritma za promatrani **NP-teški problem**. Odnos **P** i **NP** klasa još je uvijek otvoren problem ($P = NP$ ili $P \subset NP$). Kada bi se moglo dokazati da je neki **NP-potpun problem** moguće riješiti u polinomnom vremenu, tada bi se svi **NP problemi** mogli riješiti u polinomnom vremenu, pa bi klase **P** i klase **NP** bile jednake. Ako se pak pokaže da za neki **NP problem** ne postoji algoritam čije izvršavanje ovisi polinomno o ulaznim podacima, tada bi klasa **P** bila pravi podskup od klase **NP**.

No free lunch teorem za optimizacijske algoritme

Definicija 19. **Optimizacija** je grana matematike koja proučava pronalaženje ekstrema funkcija.

Definicija 20. **Kombinatorna optimizacija** je grana optimizacije, gdje je domena funkcije skup s konačnim brojem elemenata.

Definicija 21. Konkretna funkcija koja se proučava naziva se **funkcija cilja**.

Definicija 22. Funkcije cilja koje se promatraju imaju diskretnu i konačnu domenu i kodomenu. Iako se kao elementi koodomene mogu pojaviti realni brojevi, zbog toga što su računala diskretni strojevi s konačno mnogo memorije, to je samo konačni podskup realnih brojeva.

Teorem 2. Funkcija cilja definira se kao: $f : \mathbb{X} \rightarrow \mathbb{Y}$, gdje su skupovi \mathbb{X} i \mathbb{Y} diskretni i konačni. Tada je broj svih mogućih funkcija jednak $|\mathbb{F}| = |\mathbb{Y}|^{|\mathbb{X}|}$.

Dokaz. Za svaki element domene možemo odabrati točno $|\mathbb{Y}|$ elemenata koodomene.

$$|\mathbb{F}| = \prod_{j=1}^{|\mathbb{X}|} |\mathbb{Y}| = |\mathbb{Y}|^{|\mathbb{X}|} \quad (2.1)$$

□

Definicija 23. Vremenski niz d_m je niz parova domene i kodomene.

$$d_m = \{(d_m^x(1), d_m^y(1)), (d_m^x(2), d_m^y(2)), \dots, (d_m^x(m), d_m^y(m))\} \quad (2.2)$$

Iz nekog niza d_m može se izvući samo niz elemenata domene i to se označava s \mathbf{d}_m^x ili samo niz elemenata kodomene što se označava s \mathbf{d}_m^y .

Definicija 24. Prostor svih vremenskih nizova veličine m : $\mathbb{D}_m = (\mathbb{X} \times \mathbb{Y})^m$, a prostor svih vremenskih nizova maksimalne veličine m je: $\mathbb{D} = \bigcup_{m \geq 0} \mathbb{D}_m$.

Definicija 25. Optimizacijski algoritam a definira se kao: $a : d \in \mathbb{D} \rightarrow \{x | x \notin d_m^x\}$

Optimizacijski algoritam preslikava neki vremenski niz (podatke iz predhodnih koraka izvođenja) u novu vrijednost domene. S time da ta vrijednost domene ne smije biti već viđena. Ovo ograničenje koristi se u dokazu teorema, ali u praksi se često mogu javiti slučajevi gdje se više puta računa vrijednost funkcije za isti element domene.

Definicija 26. Heuristikom se naziva optimizacijski algoritam koji ne mora kao rezultat dati globalni ekstram **funkcije cilja**, ali daje dovoljno dobre rezultate u svrhu bržeg kraćeg vremenskog izvođenja.

Definicija 27. Metaheuristikom naziva se familija optimizacijskih algoritama, čije konkretno izvođenje ovisi o njegovim parametrima.

Definicija 28. Skup svih dozvoljenih rješenja u optimizaciji nazivamo **prostor pretrage**.

Teorem 3. "No free lunch" teorem za optimizaciju - Neka su a_1 i a_2 dva različita heuristička algoritma koji traže ekstrem funkcije. Funkcije su predstavljene crnom kutijom. Tada vrijedi ova jednakost:

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2) \quad (2.3)$$

Dokaz teorema može se pronaći u [literatura].

Kako se ne zna ništa više o funkcijama (npr. simbolički zapis) već samo ulazni i izlazni parovi ne može se koristiti nikakvo unutarnje znanje koje bi nekom algoritmu dalo prednost. Ako je poznat broj koraka m tada za bilo koji algoritam, sume vjerojatnosti su međusobno jednake, pri čemu se gleda vjerojatnost da se pojavio neki niz izlaznih vrijednosti funkcije d_m^y . Iz niza izlaznih vrijednosti d_m^y može se lako izvući minimalna ili maksimalna vrijednost koja je tada konačan izlaz algoritma. Dakle ne postoji algoritam koji bi bio dominantniji za sve funkcije cilja f . Zbog toga je potrebno

svaki optimizacijski problem promatrati veoma usko i koristiti dodatne informacije o samoj funkciji f .

Predhodne definicije i teoremi služe kako bi se uskladili nazivi za matematičke pojmove koji će se koristiti u radu. Sve definicije uzete su iz [literatura], gdje se međuostalim mogu naći i druge definicije i teoremi vezani za teoriju grafova, računsku teoriju složenosti i "No free lunch" teorem.

2.2.2. Bojanje težinskih grafova

Za navedeni telekomunikacijski problem potrebno je odabrati prikladan matematički model. Već je spomenuto da se bazne stanice modeliraju čvorovima u grafu. Postavlja se pitanje što predstavlja brid u tome grafu. Očigledno je da jedna bazna stanica utječe na drugu baznu stanicu, tj. mobilni uređaj istovremeno vidi više baznih stanica. To znači da bi brid mogao predstavljati relaciju da li neka bazna stanica vidi drugu baznu stanicu. Ali problem se može razviti i korak dalje. Nije svejedno na kojoj su udaljenosti bazne stanice, te koja je njihova snaga. To znači da dvije bliže (ili jače) bazne stanice više utječu jedna na drugo nego što to čine dvije udaljenije. Idajda je svakom bridu dodijeliti težinu koja predstavlja neku mjeru koliko jedna bazna stanica utječe na drugu. To znači da brid između dva čvora daje mjeru međusobnog utjecaja čvorova. Ako je utjecaj premali (tehnički nevidljiv), tada brid između ta dva čvora ne postoji.

Na tako zadanom težinskom grafu potrebno je napraviti bojanje grafa tako da se poštuju tehnički uvjeti. Sad se pojavio problem jer bojanje težinskog grafa nije dobro definiran problem. Zato se klasični problem bojanja grafova poopćuje:

Definicija 29. *Bojanje težinskog grafa je optimizacijski problem u kojem se minimizira funkcija cilja:*

$$f(\phi) = \sum_{i=1}^{E(G)} \omega(e_i) \cdot R(e_i) \quad (2.4)$$

, gdje je $\omega(e_i)$ težina brida, a R funkcija koja ima vrijednost 1 ako je e_i konfliktan brid ili 0 inače. Funkcija f ovisi o bojanju grafa, tj. funkcija cilja jednaka je sumi težina konfliktnih bridova.

Zbog tehničkih ograničenja potrebno je uvesti još malo matematičkih detalja koji pokrivaju te slučajeve.

Definicija 30. *Funkcija ϕ definira se kao funkcija koja svakom ne nepromjenjivom čvoru v u grafu G pridružuje prirodan broj iz skupa dopuštenih boja za taj čvor.*

Definicija 31. *Ako su dva čvora **nepromjenjiva** spojena bridom i imaju iste boje, tada smatramo da taj brid nije **konfliktan**.*

Razlog ovakvoj definiciji je to što bridovi koji spajaju dva nepromjenjiva čvora različitih boja ne mogu se nikada poboljšati, pa shodno tome neovisni su o funkciji bojanja grafa ϕ .

Definicija 32. *Ako su dva čvora koja spadaju u suprotne vrste spojena bridom, tada se takav brid ne smatramo **konfliktnim**.*

Tehnički takva dva čvora ne utječu jedan na drugog pa tada nema potrebe to ugrađivati u funkciju cilja.

U slučaju da je čvor obojan bojom koja nije u njegovoj domeni tada je potrebno dodati u funkciju cilja kaznu za takav slučaj. Iako po definiciji funkcije bojanja grafa to nije moguće, zbog inicijalnog stanja boja koje algoritam primi kao ulaz može se dogoditi da boja čvora nije u njegovoj domeni. Stoga se definira proširena funkcija cilja kao:

Definicija 33.

$$f(\phi) = \sum_{i=1}^{E(G)} \omega(e_i) \cdot R(e_i) + \sum_{i=1}^{V(G)} C(v_i) \quad (2.5)$$

Prva suma jednaka je već definiranoj funkciji cilja. Drugi član je suma funkcije C po čvorovima, gdje funkcija C ima vrijednost ε ako je čvor v_i krivo obojan ili 0 ako je obojan bojom iz svoje domene.

Za epsilon se odabire neki veliki pozitivan broj. U konkretnoj implementaciji koja se koristi u radu $\varepsilon = 10000000$.

Definicija 34. *Smatra se da je bojanje **valjano** ako postotak promjene boja inicijalnog stanja ne prelazi prag od α posto.*

U konkretnom problemu $\alpha = 66\%$.

Sažetak problema kojeg modeliramo pomoću težinskog grafa glasi:

- Svaki čvor može poprimiti jednu boju iz skupa dopuštenih boja za taj čvor
- Svaki čvor spada u jednu grupu tj. vrstu (A, B ili C)
- Neki čvorovi su definirani kao nepromjenjivi, njima se boja nikad ne smije mijenjati
- Dodatan uvjet je da postotak promjenjenih čvorova naprema inicijalnom stanju ne smije biti veća od 66%
- Optimizacija se provodi nad funkcijom cilja definiranom formulom (2.5).

2.3. Analiza problema bojanja težinskih grafova

U ovom odjelku biti će analiziran zadani matematički problem optimizacije. U svakom koraku analize uspoređuju se klasičan problem bojanja grafova i bojanje težinskih grafova. Kreće se od analize prostora pretrage. Zatim se dokazuje u kojoj se klasi problemi nalaze. Na kraju se pokazuje da "No free lunch" teorem vrijedi i za bojanje grafova i kakve posljedice donosi.

2.3.1. Prostor pretrage stanja

Kod klasičnog problema bojanja grafova analiziraju se dva problema. Prvi je odrediti da li se graf može pobožati s najviše k boja, a drugi problem je odrediti najmanji takav k . Pretpostavimo za početak da provjeravamo da li je graf **k-oboživ**. Pitanje je na koliko načina se graf može pobožati ako koristimo najviše k boja. Svaki čvor može poprimiti k boja. To znači da je ukupan broj bojanja N jednak:

$$N = k^{V(G)} \quad (2.6)$$

Valja primjetiti da u ovom brojanju veličine prostora stanja postoji više jednako vrijednih bojanja. U bojanju nije bitna točna boja, već je bitno da različiti skupovi čvorova imaju različite boje. Točno pitanje je na koliko načina možemo čvorove podijeliti u k skupova (teorem 1).

$$N = \sum_{i=1}^k S(V(G), i) \quad (2.7)$$

Oznaka $S(m, n)$ je za *Stirlingov* broj druge vrste, gdje je m broj različitih elemenata (čvorovi) koje smještamo u n istovrsnih skupova boja, tako da svaka boja ima barem jedan čvor. Kako je dopušteno da kutije mogu biti prazne, tada se problem pretvori u disjunktne probleme gdje se koristi samo i kutija.

Ako se zahtjeva da svaka od k boja mora biti iskorištena, ekvivalentno da ni jedan od k skupova ne smije biti prazan, tada je ukupan broj stanja nešto manji:

$$N = S(V(G), k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^{V(G)} \quad (2.8)$$

Ako se čvrsto zahtjeva da svaka boja mora biti barem na jednome čvoru, tada je to *Stirlingov* broj druge vrste, gdje je i navedena formula za izračun $S(m, n)$.

Neka se promatra problem traženja kromatskog broja grafa $\chi(G)$, tj. minimalni broj boja k , a da se graf može legalno obožiti. Maksimalni broj za koji je to potrebno

provjeriti je broj čvorova u grafu, zato jer u tom slučaju svaki čvor povezan je sa svakim drugim i svaki čvor mora imati svoju boju. Kromatski broj grafa je dakle ograničen odozgo s brojem čvorova. Za svaki broj k do $V(G)$ treba provjeriti da li je graf **k-obojev**. Ukupan broj stanja je:

$$N = B(V(G)) = \sum_{i=0}^{V(G)} S(V(G), i) \quad (2.9)$$

Broj $B(n)$ naziva se *Bellov broj* i on predstavlja broj načina na koje se skup od n članova može podijeliti u neprazne podskupove. Za bolji uvid koliko brzo *Bellov niz* raste dane su nejednakosti (za $n \geq 8$):

$$2^n \leq B(n) \leq n! \quad (2.10)$$

Bellov broj brže raste od eksponencijalne funkcije, ali sporije od faktoriijela.

Dokaz. Prvo se pokazuje nejednakost: $2^n \leq B(n), n \geq 5$. Dokaz se provodi indukcijom. Koriste se razvoj binoma:

$$2^n = (1 + 1)^n = \sum_{k=0}^n \binom{n}{k}$$

i svojstvo Bellovog broja:

$$B(n + 1) = \sum_{k=0}^n \binom{n}{k} B(k)$$

$$2^n \leq B(n) \quad (2.11)$$

$$\text{Baza indukcije: } n = 5, 2^5 \leq B(5) \rightarrow 32 \leq 52 \quad (2.12)$$

$$\text{Pretpostavka: } 2^n \leq B(n) \quad (2.13)$$

$$2^{(n+1)} \leq B(n+1) \quad (2.14)$$

$$2 \cdot 2^n \leq B(n+1) \quad (2.15)$$

$$\sum_{k=0}^n 2 \cdot \binom{n}{k} \leq \sum_{k=0}^n \binom{n}{k} B(k) \quad (2.16)$$

$$\sum_{k=0}^3 2 \cdot \binom{n}{k} + \sum_{k=4}^n 2 \cdot \binom{n}{k} \leq \sum_{k=0}^3 \binom{n}{k} B(k) + \sum_{k=4}^n \binom{n}{k} B(k) \quad (2.17)$$

$$\text{Prvo se dokazuje: } \sum_{k=0}^3 2 \cdot \binom{n}{k} \leq \sum_{k=0}^3 \binom{n}{k} B(k) \quad (2.18)$$

$$2 \binom{n}{0} + 2 \binom{n}{1} + 2 \binom{n}{2} + 2 \binom{n}{3} \leq \quad (2.19)$$

$$\binom{n}{0} B(0) + \binom{n}{1} B(1) + \binom{n}{2} B(2) + \binom{n}{3} B(3) \quad (2.20)$$

$$2 + 2n + \frac{n(n-1)}{2} 2 + \frac{1}{3} n(n-1)(n-2) \leq \quad (2.21)$$

$$1 + n + \frac{n(n-1)}{2} 2 + \frac{n(n-1)(n-2)}{6} 5 \quad (2.22)$$

$$1 + n \leq n(n-1)(n-2) \left(\frac{5}{6} - \frac{1}{3} \right) \quad (2.23)$$

$$1 + n \leq \frac{n(n-1)(n-2)}{3} \quad (2.24)$$

$$3 + 3n \leq n^3 - 3n^2 + 3n \quad (2.25)$$

$$3n^2 + 3 \leq n^3 \quad (2.26)$$

$$3 \leq n^3 - 3n^2 \quad (2.27)$$

$$3 \leq n^2(n-3), \text{ istina za } n \geq 5 \quad (2.28)$$

$$\text{Druge dvije sume: } \sum_{k=4}^n 2 \cdot \binom{n}{k} \leq \sum_{k=4}^n \binom{n}{k} B(k) \quad (2.29)$$

$$\text{Član po član sume: } \binom{n}{k} 2 \leq \binom{n}{k} B(k) \quad (2.30)$$

$$2 \leq B(k), \text{ za } k \geq 4 \quad (2.31)$$

□

Dokaz. Potrebno je dokazati $B(n) \leq n!$. Dokaz se provodi iz svojstava za Bellov broj

(Berend, D.; Tassa, T. (2010)):

$$B(n) \leq \left[\frac{0.792n}{\ln(1+n)} \right]^n$$

i svojstvo faktoriijela koja slijedi iz analize Stirlingove aproksimacije:

$$n! \geq \left(\frac{n}{e} \right)^n$$

$$B(n) \leq \left[\frac{0.792n}{\ln(1+n)} \right]^n \leq \left(\frac{n}{e} \right)^n \leq n! \quad (2.32)$$

$$\text{Dovoljno je pokazati: } \left[\frac{0.792n}{\ln(1+n)} \right]^n \leq \left(\frac{n}{e} \right)^n \quad (2.33)$$

$$\left[\frac{0.792n}{\ln(1+n)} \right] \leq \left(\frac{n}{e} \right) \quad (2.34)$$

$$0.792n \cdot e \leq n \cdot \ln(n+1) \quad (2.35)$$

$$0.792 \cdot e \leq \ln(n+1) \quad (2.36)$$

$$e^{0.792 \cdot e} \leq n+1 \quad (2.37)$$

$$8.61 - 1 \leq n \quad (2.38)$$

$$7.61 \leq n \quad (2.39)$$

□

Zaključak predhodne analize je da za relativno veliki graf postoji velik broj stanja koje je nemoguće pretražiti pomoću *brute force* algoritama. Analiza prostora stanja za zadani problem malo je drugačija. Kako svaki čvor ima konačan broj boja koje može poprimiti ukupan broj stanja je:

$$N = \prod_{i=1}^{V(G)} K_i \quad (2.40)$$

, gdje je oznaka K_i ukupan broj boja koje može poprimiti čvor s indeksom i . Kako je moguće dodijeliti maksimalno samo 512 boja, možemo ograničiti K_i :

$$K_i \leq M = 512 \quad (2.41)$$

$$N = \prod_{i=1}^{V(G)} K_i \leq \prod_{i=1}^{V(G)} M = M^{V(G)} \quad (2.42)$$

Broj stanja eksponencijalno ovisi o broju čvorova, što je isto kao i za klasičan problem bojanja grafa jako velik broj stanja. I u ovom slučaju važno je primjetiti da

su neka stanja ekvivalentna. Broj jedinstvenih stanja je broj particija skupa čvorova u skupove boja, ali uz ograničenje da barem jedan čvor ima neku od zadanih M boja, tada vrijedi ova nejednakost za broj različitih stanja:

$$N \leq \sum_{i=1}^M S(V(G), i) \quad (2.43)$$

Broj stanja je manji jer se poštuje ograničenje da čvorovi mogu poprimiti neku od boja u podskupu od ukupnog konstantnog broja boja M . Kada je $k \geq M$ tada broj stanja optimizacijskog problema je manji od broja stanja u problemu gdje se traži odgovor na pitanje da li je graf **k-obojujiv**. Ovo razmatranje biti će zanimljivo i u sljedećem poglavlju gdje se problem svrstava u klase računske teorije složenosti.

2.3.2. Pozadina problema iz perspektive računske teorije složenosti

Problem odluke da li je graf **k-obojujiv** spada u klasu **NP-potpunih** problema. Traženje kromatskog broja grafa $\chi(G)$ spada u klasu **NP-teških** problema. Dokazi se mogu pronaći u [1]. Potrebno je pokazati da postavljen optimizacijski problem bojanja grafa spada u klasu **NP-teških** problema. Problem sigurno ne može spadati u druge spomenute (P, NP, NP-potpun) klase zato jer to nije problem odluke. Postupak dokazivanja da problem spada u klasu **NP-teških** problema ima nekoliko koraka. Prvi korak je pronalaženje nekog **NP-potpunog** ili **NP-teškog** problema koji će se koristiti u dokazu. Zatim se pokaže da je taj *stari* problem moguće polinomno reducirati na problem za koji se dokazuje da je **NP-težak**.

Dokaz. Odaberemo **NP-potpun problem** koji ispituje da li je graf G **k-obojujiv**. Da bismo riješili taj problem koristimo zadani optimizacijski problem. Ulaz u optimizacijski algoritam međuostalim su graf G i ograničenja boja za svaki od čvorova. Svakom čvoru daje se skup $\{1, 2, \dots, k\}$ s bojama. Svim bridovima daje se težina 1. Sada se pokreće crna kutija s optimizacijskim problemom koja će riješiti taj problem. Ako je rezultat optimizacije tj. funkcije cilja $f = 0$, tada je graf **k-obojujiv**, inače nije. U konstantnom vremenu možemo reducirati problem, iz čega sljedi da je optimizacijski problem **NP-težak**. \square

Zanimljivo je primjetiti da **NP-potpun** problem k-obojujivosti može imati više stanja od **NP-teškog** problema optimizacije. Što znači da je moguće da vrijeme izvršavanja optimizacije bude manje od vremena provjere k-obojujivosti. Kako nije poznat odnos NP i P klase, za ovaj optimizacijski problem nameće se korištenje heuristika. U ovom

tehničkom problemu nije potrebno do kraja minimizirati funkciju cilja, nego dobiti dovoljno malu vrijednost f što će uzrokovati bolju kvalitetu mobilne mreže.

2.3.3. Posljedica "No free lunch" teorema

U originalnom članku, gdje se dokazuje "No free lunch" teorem [literatura], pretpostavlja se da sve funkcije cilja imaju iste domene i kodomene. Za problem bojanja grafova to nije tako. Funkcija cilja definirana je kao: $f : V(G) \rightarrow \mathbb{R}$, ali kako računalo ima konačnu memoriju tada je funkcija zapravo definirana kao: $f : V(G) \rightarrow \mathbb{Y}$, gdje je \mathbb{Y} konačni podskup realnih brojeva koji se mogu prikazati na računalu. Problem za "No free lunch" teorem je to da veličina domene funkcije cilja ovisi o broju čvorova grafa. Broj grafova je beskonačan, jer možemo uzeti proizvoljan broj čvorova, što znači da postoji i beskonačan broj mogućih funkcija cilja. Teorem radi samo ako je broj funkcija cilja ograničen (teorem 3). Kako je na nekom računalu nemoguće prikazati proizvoljno veliki graf, broj čvorova može se ograničiti s nekom granicom M . Sada je moguće prebrojati sve funkcije cilja f .

Dokaz. Pitanje je koliko funkcija postoji, ako je broj članova domene ograničen s M . Broj funkcija koje imaju točno k elemenata u domeni ima:

$$|\mathbb{F}_k| = |\mathbb{Y}|^k \quad (2.44)$$

Ukupan broj funkcija je suma po svim k do granice M (koristi se suma za geometrijski niz):

$$|\mathbb{F}| = \sum_{k=1}^M |\mathbb{F}_k| = \sum_{k=1}^M |\mathbb{Y}|^k = \frac{|\mathbb{Y}|^{M+1} - |\mathbb{Y}|}{|\mathbb{Y}| - 1} \quad (2.45)$$

□

Ako se ograniči broj čvorova, dobije se konačan broj funkcija te je to lako ugraditi u teorem 3. Teorem govori o tome da nemamo nikakvo znanje o funkciji. U ovom problemu funkcija cilja direktno ovisi o grafu tj. strukturi i to je važna informacija koja treba biti iskorištena da bi se dobili bolji algoritmi. Glavna posljedica ovog teorema je razvoj pohlepnih algoritama koji koriste znanje o strukturi (grafu) i primjena tih algoritama u metaheurističkim postupcima. Važno je uočiti da je svaki graf posjeduje vlastitu funkciju cilja. To znači da korištenjem algoritama na sljepo koji rade na nekim grafovima ne mogu jamčiti efikasnost na drugim.

U članku [2] opisuje se kako postoji slučajevi u kojima hiper-heuristike mogu imati "Free lunch" svojstvo. Hiper-heuristike su algoritmi koji traže najbolji algoritam koji

može riješiti neki problem, tj. pronaći ekstrem funkcije cilja. Kako je gore opisano, svaki graf ima svoju funkciju cilja. Ako je skup funkcija cilja dovoljno malen, moguće je da hiper-heuristika uvijek pronađe neki algoritam koji je bolji od svih ostalih za konkretnu funkciju cilja. Ovo nije dokazano za bojanje grafova (tako ni za optimizacijski problem bojanja), ali daje motivaciju u razvoju algoritama. U zadnjem dijelu opisa algoritama u ovom radu koriste se metode strojnog učenja kako bi se izvuklo znanje za primjenu u nekim pohlepnim algoritmima, tj. kako bi se odabrao najbolji pohlepan algoritam, što je zapravo jednostavan oblik hiper heuristike. Osim toga u implementaciji je otvoren prostor gdje se lako mogu donostiti odluke i dinamički birati algoritmi koji u trenutnoj točki u prostoru stanja daju bolje rezultate od drugih.

3. Algoritmi

4. Infrastruktura

5. Testiranje

6. Rezultati

7. Zaključak

Zaključak.

LITERATURA

- [1] M. R. Garey, D. S. Johnson, i L. Stockmeyer. Some simplified np-complete problems. U *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, stranice 47–63, New York, NY, USA, 1974. ACM. doi: 10.1145/800119.803884. URL <http://doi.acm.org/10.1145/800119.803884>.

- [2] Riccardo Poli i Mario Graff. There is a free lunch for hyper-heuristics, genetic programming and computer scientists. U Leonardo Vanneschi, Steven Gustafson, Alberto Moraglio, Ivanoe De Falco, i Marc Ebner, urednici, *EuroGP*, svezak 5481 od *Lecture Notes in Computer Science*, stranice 195–207. Springer, 2009. ISBN 978-3-642-01180-1. URL <http://dblp.uni-trier.de/db/conf/eurogp/eurogp2009.html#PoliG09>.

Bojanje grafova prilagodljivim metaheurističkim postupcima

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.