Univerza v Ljubljani
Fakulteta za matematiko in fiziko

Predstavitev 4-ciklov v snarkih

Eva Strašek in Vanja Kalaković

Mentorja: Janoš Vidali, Riste Škrekovski

Predmet: Finančna matematika

Ljubljana, 2023

1 Naloga:

Želimo preveriti, ali (in kdaj) uvedba 4-ciklov v snark ohranja kromatični indeks(tj. kromatični indeks ostane 4). Uvedbo 4-cikla lahko izvedemo vsaj na dva načina:

1. Vzemite dva robova ab in cd v G in ju dvakrat razdelite, tako da dobite pot au1u2b iz roba ab in pot cv1v2d iz roba cd. Nato povežite u1 z v1 in u2 z v2.

2. Naj bo ab rob v G in a1, a2 druga dva soseda a, in naj bosta b1, b2 druga dva soseda b. Zdaj odstranimo vrhova a, b in povežemo a1 z b1 in a2 z b2.

2 Uvod

Začeli bomo z definicjami in izreki, ki nam bodo v pomoč pri razumevanju naloge in pisanju programa.

Definicija 1.1: Stopnja vozlišča v v grafu G (oznaka je $deg_G(v)$) je enaka številu povezav grafa G, ki imajo vozlišče v za svoje krajišče, pri čemer štejemo zanke dvakrat.

Definicija 1.2: Če so vsa vozlišča grafa G enake stopnje k, pravimo, da je graf **k-regularen**; 3-regularnim grafom pravimo tudi **kubični grafi**.

Definicija 1.3: Graf je **povezan**, če za poljubni vozlišči obstaja pot med njima, sicer je graf **nepovezan**.

Definicija 1.4: **Komponenta** grafa G je maksimalen povezan podgraf grafa G.

Definicija 1.5: Vozlišče $v \in V(G)$ je **prerezano vozlišče**, če ima podgraf G - v več komponent kot graf G.

Definicija 1.6: Povezavi, ki ima za krajišči prerezani vozlišči, pravimo **prerezana povezava** ali **most**.

Definicija 1.7: **Kromatično število** grafa G (oznaka χ(G)) je najmanjše število barv, ki jih potrebujemo, da vozlišča grafa pobarvamo tako, da so vsa sosednja vozlišča pobarvana paroma različno.

Definicija 1.8: **Kromatični indeks** grafa G je število χ'(G), ki predstavlja najmanjše število barv, ki jih potrebujemo za barvanje povezav grafa G tako, da so sosednje povezave pobarvane paroma različno.

Vizingov izrek: Za enostaven graf G z maksimalno stopnjo Δ(G) velja Δ(G) $\leq$ χ'(G) $\leq$ Δ(G) + 1.

Definicija 1.9: Če so vsa vozlišča $v_0, v_1, \ldots, v_k$ različna, govroimo o poti, v primeru, ko pa so vsa vozlišča različna, razen $v_0 = v_k$, imamo opravka z **obhodom**. Dolžini najkrajšega obhoda v grafu G pravimo **notranji obseg**.

Definicija 2.1: Naj bo A množica prerezanih povezav moči 3 in G graf. Če G - A predstavlja dve komponenti, ki vsebujeta cikel, pravimo da je A ciklični prerez.

Definicija 2.2: Ciklična povezavna povezanost grafa G (zapis λ_c(G)) je velikost najmanjšega cikličnega prereza grafa G. Pravimo, da je G ciklično k-povezavno povezan, če je λ_c(G) $\geq$ k. Ali drugače, grafu G moramo odstraniti najmanj k-povezav, da nam ta razpade na dve komponenti, ki vsebujeta cikel.

Definicija 2.3: Snark je ciklično 4-povezavno povezan kubičen graf z notranjim obsegom vsaj 5 in kromatičnem indeksom 4.

Lema: Naj bo graf G' dobljen tako, da kubičnemu grafu G dodamo vozlišča a_1, a_2 in a_3, a_4 zaporedno na povezavi $e_1 = u_1v_1$ in $e_2 = u_2v_2$ ter povezave a_1a_4 in a_2a_3. Če je χ'(G') = 4, potem je χ'(G) = 4.

3 Opis dela

Najprej sva se lotili vpeljevanja 4-ciklov po metodi 1. Napisali sva funkcijo, ki kot argumente sprejme snark, število vozlišča, ki nastopa v grafu. Tu sva pazili, da si povezave zaporedno sledijo, torej če vzameva za a = 0 in b = 1, morava za c vzeti 2 in za d vzeti 3. Nato sva določili povezavo eno, ki je ab ter povezavo dve, ki je cd. Na vsako povezavo sva doodali dve oglišči, ki skupaj z originalnima vozliščema tvorita pot (npr. au1u2b). Ko so bila vozlišča dodana, sva dodali še povezave med u1 in v1 ter u2 in v2. Na koncu funkcije sva dali, da mora vrniti kromatični indeks novo nastalega grafa.

Nato sva napisali še funkcijo, kjer poljubno zberemo povezave na katere dodajamo vozlišča u1,u2,v1 in v2. Tu smo definirali **random edge 1** kot naključno izbrani rob in **random edge 2**, ki je prav tako naključno izbran, dodali pa smo še pogoj, da nesme bitit isti kot prvi. Od tu naprej ponovimo postopek kot zgoraj.

Prav tako naju je zanimalo, kaj se zgodi s kromatičnim indeksom, če 4-cikel uvedemo po metodi 2. Zato sva pri tretji funkciji najprej shranili seznam vozlišč, nato pa iz tega seznama naključno izbrali eno vozlišče in ga shranili pod spremenljivko a. S funkcijo .neighbors sva poiskale vse sosede vozlišča a in naključno izbrale enega ter ga shranili pod spremenljivko b. Pogledali sva še vse sosede vozlišča b, nato pa iz obeh seznam odstranili vozlišče a in b. Ker nas ne zanimi seznam vozlišč ampak ime oziroma oznaka posameznega vozlišča sva vsakega soseda a shranili pod svojo spremenljivko (a1 in a2) in enako za sosede b. Nato sva dodali povezavi med a1b1 in a2b2 ter odstranili povezavo ab. Na koncu sva dodali še, da mora funkcija vrniti kromatični indeks novo nastalega grafa.

Funkcije sva nazadnje preverili na 8 snarkih, kjer se lepo vidi kdaj se kromatični indeks ohrani in kdaj ne.

4 Končni sklep

Ko poženemo prvo funkcijo opazimo, da pri vsakem grafu dobimo kromatični indeks 4, ta pa je enak kot pri snarkih. Ta sklep nam da tudi lema, ki opisuje točno to metodo, ki jo mi uporabimi v funkciji 1. Torej v primeru, da na snark G dodamo vozlišča u1u2 in v1v2 na dve zaporedni povezavi (tj. ab in cd) bo novo nastali garf G' ohranil kromatični indeks.

Pri drugi funkciji se v nekaterih primerih kromatični indeks ne ohrani. To se zgodi, saj funkcija naključno zbere robove na katere dodaja vozlišča. Opazimo lahko, da v primerih ko za robova izberemo dve sosednji povezavi, kromatični indeks ne bo več 4 (tj. če izberemo recimo ab in bc). Da pridemo do te ugotovitve nam je v pomoč prva funkcija, saj vidimo da pri prvi funkciji se kromatični indeks ohranja. Torej naš sklep je, da v primeru naključnih izborov robov, novo nastali graf G' ne ohranja kromatičnega indeksa v primeru, ko sta izbrana sosednja robova.

Pri zadnji funkciji opazimo enako, kot pri prvi, da je kromatični indeks vedno enak 4. Torej v primeru, da novo nastali graf generiramo po metodi 2 bo ta vedno ohranil kromatični indeks.