Minimum Tree Width

Vladimir Kastratović 31/2019, Viktor Gizdavić 146/2017

27. septembar 2024.

Sadržaj

1 Uvod

Neka je dat neusmeren graf $G = (V, E)$. Dekompozicija stabla je par $(\{X_i : i \in I\}, T)$ gde je $T = (I, F)$ drvo i $\{X_i\}$ je skup podskupova od V, tako da:

1. $\bigcup_{i \in I} X_i = V$,
2. za svako $(v, w) \in E$, postoji $i \in I$ tako da $u, v \in X_i$,
3. za svako $v \in V$, skup $\{i \in I : v \in X_i\}$ predstavlja povezano poddrvo od T.

Širina stabla dekompozicije je jednaka $Tw = \max_{i \in I} |X_i| - 1$. Minimalna širina stabla je najmanje takvo Tw.

Ovaj problem pripada grupi NP-teških problema za koje se ne zna da postoji algoritam koji izračunava rešenje u polinomijalno mnogo koraka. Mnogi algoritmi koji rešavaju NP-teške probleme iz oblasti računarske inteligencije, su eksonencijalne složenosti samo pri izračunavanju širine stabla. Jedan od sličnih problema ovom iz naslova je: "Junction-tree elimination [Lauritzen and Spiegelhalter, 1988]". Ovaj algoritam se odvija

u dva koraka: (1) konstrukcija dobre dekompozicije stabla i (2) rešavanje problema nad ovom dekompozicijom, gde je drugi korak uglavnom eksponencijalan po širini stabla dekompozicije iz koraka (1). Radi postizanja efikasnijeg rešenja, može se koristiti "branch and bound"tehnika. Neke od poznatih heuristike koje se koriste u ovoj metodi su "min-width", "max-cardinality search, min-fill". Međutim, iako njihova upotreba obećava efikasnije rešenje, u najgorem slučaju, algoritam može biti eksponencijalne složenosti. Zbog toga, u ovom radu je izložena novija heuristika bazirana na "min-width", koja će biti opisana u narednim sekcijama. Algoritam koji upotrebljava ovu heuristiku u svom izračunavanju, može se završiti u $O(\log|V|)$ složenosti.

2 Definicije i leme

Pre nego što predstavimo rešenjee problema, potrebno je upoznati se sledecim definicijama i lemama.

Definicija 2.1. Kord ciklusa C je grana koja se ne nalazi u ciklusu, ali njene tačke se nalaze u ciklusu.

Definicija 2.2. Za ciklus C kažemo da je bez korda, ako ne sadrži nijedan kord i dužine je najmanje 4.

Definicija 2.3. Za graf G kažemo da je trijangulisan ukoliko u njemu ne postoji ciklus bez korda.

Definicija 2.4. Za čvor v grafa G, kažemo da je simplicijalan, ukoliko njegovi susedi kreiraju klik(potpuno povezan graf). Čvor v je skoro simplicijalan ukoliko svi susedi osim jednog formiraju klik.

Definicija 2.5. Poredak $\pi = [v_1, v_2, ..., v_n]$, nazivamo poredak savršene eliminacije ukoliko za svako $1 \leq i \leq n$, v_i je simplicijalan u podgrafu $G^{'} = (V^{'}, E^{'})$ gde je $V^{'} = \{v_i, ...v_n\}$.

Definicija 2.6. Klik veličine k + 1 je k-stablo veličine k + 1. K-stablo veličine n + 1 može se konstruisati od k-stabla veličine n tako što se doda novi čvor i poveže se sa bilo kojim klikom veličine k. Podgraf k-stabla je parcijalno k-stablo.

Lema 2.7. K-stablo je širine k. Parcijalno k-stablo je širine najviše k.

Lema 2.8. Graf je trijangulisan akko postoji poredak savršene eliminacije. Dodatno, ukoliko je trijangulisan, bilo koji čvor može biti na početku permutacije.

Lema 2.9. Za bilo koji poredak savršene eliminacije kordalnog grafa T, važi da je širina od T jednaka $\max\left(|N(v_i)| \,\middle|\, v \in V,\, N(v) \cap \{v_i, \ldots, v_n\}\right)$

Lema 2.10. Neka je G graf i M njegov minor. Tada važi da je $\mathrm{tw}(G) \geq \mathrm{tw}(M)$.

3 Algoritmi za rešavanje problema

3.1 Gruba sila

Algoritam grube sile podrazumeva prolazak kroz sve moguće permutacije čvorova i nalaženje minimuma od dobijenih širina stabala.

Širina stabla se računa obradom jedne permutacije, i dekomponovanjem grafa na drvo sačinjeno od torbi čvorova. Torbe su zapravo čvorovi koji objedinjuju više originalnih čvorova. Originalni čvorovi se redom izbacuju iz trenutne permutacije i pritom dodaju u torbe čuvajući simplicijalnost grafa. Čuvanjem simplicijalnosti, otvaraju se nove prilike kao kandidati za torbe. Ukoliko u nekom trenutku torba predstavlja podksup neke od već postojećih torbi, ne dodaje se u novonastalo drvo. U suprotnom, nova torba se vezuje za sve torbe sa kojima ima presek. Permutacija se smatra obrađenom kada su svi čvorovi obradjeni, a širina stabla je za jedan manje od broja čvorova u najvećoj torbi.

3.2 Optimizovani algoritam

Za izračunavanje minimalne širine stabla koristimo "branch and bound"algoritam. Za heuristiku, "min-width"i "minor-min-width".

3.2.1 min-width algoritam

Ulaz: Neusmeren, povezan graf G.
Izlaz: Donja granica širine grafa G.

1. $lb = 0$
2. Ponavljaj:
   (a) Nađi čvor v sa najmanjim stepenom.
   (b) Napravi novi graf G' bez čvora v.
   (c) $lb = MAX(lb, degree_G(v))$
   (d) Postavi G na G'
3. dokle god ima čvorova u G
4. return lb

3.2.2 minor-min-width algoritam

Ulaz: Neusmeren, povezan graf G.
Izlaz: Donja granica širine grafa G.

1. $lb = 0$
2. Ponavljaj:
   (a) Spoji granu između čvora v najmanjeg stepena i čvora $u \in N(v)$ takvog da je stepen ćvora u najmanji u $N(v)$ da bi napravili novi graf G'
   (b) $lb = MAX(lb, degree_G(v))$
   (c) Postavi G na G'
3. dokle god ima čvorova u G
4. return lb

Minor-min-width heuristika je bazirana na teoremi iz teorije grafova, koja tvrdi da širina grafa nikad nije manja od širine njegovor minora.

3.2.3 BnB algoritam

Ulaz: Neusmeren, povezan graf G.
Izlaz: Minimalna širina stabla.

1. Inicijalizacija: Stanje s je par grafa $G^s = G$ i parcijalnog poretka $x^s = \phi$.
   $g(s) = 0, h(s) = mmw(G), f(s) = h(s)$, $ub = \infty$.
2. If($f(s) < ub$) **BB**(s)
3. return ub

pod-procedura BB(s)

1. IF $|V_G s| < 2$ THEN $ub = MIN(ub, f(s))$
2. ELSE FOR za svaki čvor v u G^s:
   (a) Napravi stanje $s' = (G^{s'}, x^{s'})$ gde je
   $G^{s'} = elim(G^s, v)$ i $x^{s'} = (x^s, v)$.
   (b) $g(s') = MAX(g(s), degree_{G^s}(v))$
   (c) $h(s')$ = **minor-min-width**$(G^{s'})$
   (d) $f(s') = MAX(g(s'), h(s'))$
   (e) If $f(s') < ub$ onda $BB(s')$

Operacija $elim(elim(G^s, v)$ podrazumeva eliminisanje čvora v čineći ga simplicijalnim. Rezultat ove operacije je:

$$G_0 = (V \setminus \{v\}, E \cup E_0), \quad \text{gde } E_0 = \{(v_1, v_2) \mid v_1, v_2 \in N(v)\}$$

BnB algoritam je optimizacija grube sile koristeći heuristike.

Pre samog pokretanja algoritma, vrednost gornje granice se postavlja na beskonačno.

Vrednost donje granice se računa uz pomoć heuristike u toku rada algoritma nad trenutnom permutacijom.

Ukoliko u toku obrade neke permutacije, vrednost donje granice postane veća od gornje, obustavlja se dalja pretraga. U suprotnom, vrednost gornje granice se ažurira na maksimum donje granice i koristi se u narednim permutacijama.

Svako stanje s je par: Graf G^s i parcijalni poredak x^s. Graf G^s u stanju s se dobija izbacivanjem čvorova u poretku x^s iz originalnog grafa G. Naredna stanja se dobijaju iz stanja s izbacivanjem proizvoljnog čvora $v \in V_{G^s}$. Vrednost g stanja s je širina poretka x^s od početnog čvora, vrednost h je donja granica širine stabla G^s.

4 Eksperimentalni rezultati

U narednoj tabeli su prikazani rezultati izvršavanja svakog od tri algoritma. Grafovi zadati brojem čvorova i grana su nasumično generisani. Za svaku zadatu veličinu grafa je izgenerisano po pet grafova i nad njim primenjeni algoritmi. Prosečne vrednosti su zatim unete u tabelu.

Za algoritam grube sile, već za n >=10, vreme izvršavanja algoritma traje makar 5 minuta, pa su ti rezultati odsečeni.

n	**e**	**brute force**	**min-width**	**minor-min-width**	**avg tw**
5	8	0.00155411	0.000109767	0.000291568	3
8	15	1.29493	0.000596336	0.00301754	3.33333
10	30	-	0.00208704	0.0141712	6
12	40	-	0.00535412	0.0325081	7
15	50	-	0.0465442	0.107265	7
17	50	-	10.2123	10.4519	7.33333
20	100	-	5.49768	36.4117	4
20	150	-	0.0100785	0.172352	5

Tabela 1: Prosečno vreme izvršavanja algoritma i dobijeni rezultati

5 Zaključak

BnB algoritam izložen u sekciji 2.2.2 se može dodatno unaprediti na sledeće načine:

1. Prilikom inicijalizacije, umesto postavljanja gornje granice na beskonačnost, može se izračunati min-fill ili nekom drugom heuristikom. Ovo doprinosu bržem zasecanju.
2. Poboljšavanje vrednosti f za svako stanje s
3. Redukcija faktora grananja u svakom stanju
4. Korišćenje pravila propagacije i zasecanja
5. Matricu povezanosti zameniti katalogom čiji su ključevi čvorovi grafa, a preslikane vrednosti skup suseda čvorova.

Literatura

[1] Vibhav Gogate and Rina Dechter, *A Complete Anytime Algorithm for Treewidth*, 11. Jul 2012.