Algoritmy pro specifické třídy grafů

2. ledna 2013

Obsah

1 Grafy omezené stromové šířky

Stromový rozklad G je strom T kde $X_i \in V_T \Rightarrow X_i \subseteq V(G)$. Dále, pro $\forall(v_1, v_2) \in E(G) \exists X_i; v_1 \in X_i \wedge z_2 \in X_i$ a když zafixuji $v \in V(G)$ potom $\forall X_i; v \in X_i$ je souvislý podstrom.

Místo posledního je možné brát že je-li X_k na cestě mezi X_i a $X_j \Rightarrow X_i \cap X_j \subseteq X_k$.

Šířka rozkladu T je $\max X_i - 1$.

Stromová šířka je minimální šířka rozkladu přes všechny rozklady (jeden graf může mít více rozkladů, rozklad kde jediný X_1 obsahuje všechny vrcholy určitě existuje).

Stromová šířka grafu je nejvýše 1 $\Leftrightarrow G$ je les.

Stromová šířka grafu je nejvýše 2 $\Leftrightarrow G$ je podgrafem sériově-paralelního grafu (což je totéž, jako že nemá K_4.

Pokud H je podgraf G, potom $tw(H) \leq tw(G)$.

Pokud G obsahuje K_n, potom stromový rozklad je alespoň $n - 1$.

$\forall(X_i, X_j) \in E(T)$ platí, že $X_i \cap X_j$ je řez v G, nebo každý uzel za X_i je podmnožinou X_j nebo naopak.

K programování použijeme dynamické programování, v jednom X_i je max omezená interference mezi částmi. Použijeme zakořeněný stromový rozklad, z podstromů bereme výsledky a spojujeme (posílá se více řešení nahoru, bere se jak to jde nejlepší).

1.1 Maximální nezávislá množina

Hledáme $\alpha(G)$ nějakého grafu G (tedy, maximální podmnožina vrcholů, kde nevede ani jedna hrana).

Kromě G dostaneme i zakořeněné T. Výstupem má být velikost této množiny.

Pro každý vrchol máme tabulku dvojic (S, a), kde S je nezávislá na X_i a lze ji rozšířit na nezávislou množinu o velikosti a na celém podstromu T.

Je-li X_i list, tak hrubou silou vytahám nezávislé podmnožiny.

Má-li X_i zpracované děti, potom pro každou nezávislou S na X_i potřebuji spočítat a. Pro každé dítě se podívám, jaká nejlepší jeho nezávislá množina je pořád ještě nezávislá s tou mojí S. Tedy, a bude $|S|+$ součet přes všechny děti takové, jaké jsou nejlepší nezávislé množiny těch „zbytků“.

Jednoduše vezmu všechny ty a, odečtu jim společné části.

V kořenu vezmu nejlepší a.

Časová složitost je $O(|V(T)| \cdot 2^{tw(G)})$. Pro danou fixovanou šířku to je $O(N)$ (protože optimální má nejvýše $n-1$ vrcholů v T).

1.2 Pěkné stromové rozklady

Pěkný stromový rozklad je takový, že jeho uzly jsou 4 druhů.

- Listy. Ty obsahují pouze jeden vrchol.
- Množivé uzly X_i. Mají jedno dítě X_j, platí, že $X_i = X_k + u$ (přidaný jeden vrchol).
- Inverze tohoto.
- Spojovací. Mají dvě děti X_j, X_k, obě děti mají stejné množiny.

Někdy se požaduje, aby kořen byl stupně 1.

Tvrzení 1 *Pro každý rozklad T existuje pěkný rozklad T' stejné šířky.*

Důkaz:
Zakořeníme v libovolném listu. Poté se moc-větvící nahradí binárními stromečky. Poté se natáhne to, co se moc liší, přeměňuje se symetrická diference. Nakonec se vytáhnou listy.

☻

Zajímalo by nás, jak velký bude tento rozklad vzhledem k n, pokud je stromová šířka fixovaná.

Problém bude jen u spojovacích vrcholů. Na obou stranách jsou různé podstromy (jinak můžu jednu zahodit). Tedy, těch může být maximálně lineárně.

1.3 Barevnost

Máme dán graf a nějaké k. Ptáme se, jestli je obarvitelný k barvami. Chromatické číslo je nejvíce stromová šířka $+1$.

Předpokládejme, že tedy k je nejvýše stromová šířka.

Pro každé X_i uložíme do tabulky všechna možná obarvení s nejvýše k barvami, s tím, že se berou jen ekvivalence.

Když X_i je list, pak je tam jen jedna množina. Když máme množivý, tak nemůže být strčen do stejné množiny, jako ty, se kterými je spojen. Ubírání je v pohodě. Je potřeba mergovat ty, které se „tváří“ stejné. Spojovací jen vybereme ty, které si dokáží odpovídat. Je-li to v kořeni neprázdné, tak má obarvení.

1.4 Hamiltonka

V tabulce si pamatujeme, jak by to mělo jít spojit nahoře, pamatují se nějaké dvojice.

1.5 Monadická logika 2. řádu

Věta 1 *Grafová vlastnost vyjádřitelná monadické logice 2. řádu lze ověřit v lineárním čase na grafech omezené stromové šířky.*

MSOL formule má proměnné pro vrcholy, hrany, množiny vrcholů a množiny hran. Máme rovnost, inkluze, spojky a kvantifikátory.

Patří sem např. k-barevnost pro dané k, hamiltonovskost (2-faktor + souvislost přes to, že nejde najít rozdělení).

Nechť tedy máme formuli φ a jeho pěkný rozklad T. Napřed určíme možné hodnoty proměnných φ v X_i. Poté je potřeba zařídit, aby to šlo reprezentovat konečně mnoha třídami. Nakonec je potřeba dokázat, že z dat dětí lze udělat data pro rodiče.

1.5.1 Reprezentace proměnných

Pokud je x_v vrcholová proměnná, potom si pamatujeme buď že už byla a nebo že bude (hádání jako u hamiltonky). U hran buď lokální, nebo „někde jinde“. Proměnná pro množiny vrcholů budou mít ukládané jen průniky s lokálním uzlem.

1.5.2 Generování hodnot volných proměnných

U listů jsou všechny hodnoty nahoře nebo lokálně, hranová proměnná jedině jinde, množiny buď prázdná, nebo jednoprvková u vrcholů.

U zapomínací je to jednoduché (jen se z nich stávají „spodní“ vrcholy nebo hrany). Toto zobrazení nemusí být prosté.

Když X_i bude generovací uzel, tak zachovává vše kromě „horních“. U některých horních si můžu vymyslet hodnotu a vzít si ji k sobě.

U joinu jen vybírám ty, které se shodují na externím, nebo je právě jeden interní.

1.5.3 Reprezentování formulí

Predikát $P(x_1, \ldots, x_k)$ je ***regulární***, pokud existuje konečná C a $C' \subseteq C$ a zobrazení h ze všech možných ohodnocení proměnných do C následujících vlastností:

- Když zobrazení h zobrazí dvě k-tice na stejný element, potom ty predikáty vycházejí také stejně.
- Výsledek v C' právě když je P splněný.
- h závisí pouze na svých synech.

Zbývá dokázat, že ty naše MSOL predikáty jsou regulární.

2 Kliková šířka

Nechť $I = \{1, \ldots, k\}$ je množina značek. ***Sestavovací strom*** je zakořeněný strom z uzlů typů:

- Vstupní uzel $i(u)$ pro $i \in I$ říká, že zavádím vrchol u se značkou i. Toto musí být list stromu.
- Sjednocovací $\oplus$, má dva syny a výsledný graf je disjunktní sjednocení jeho dvou synů.
- Spojovací uzel $\eta_{i,j}$ s jedním dítětem, říká, že má spojit vrcholy s danými značkami (všechny).
- Přeznačovac $\rho_{i \to j}$, přejmenuje všechny i na j.

Kořen odpovídá nějakému označkování grafu.

Kliková šířka je nejmenší počet značek nutných k vybudování grafu (značí se $cw(G)$.

Pozorování 1 *G je indukovaný podgraf G', potom $cw(G) \leq cw(G')$*

Důkaz:
Jen něco nevyrobím.

☻

Graf odpovídající uzlu je ovlivněn jen podstromem a cestou do kořene.

Pozorování 2 *Stačí předpokládat, že se každá hraná max. jednou.*

Důkaz:
Stačí vzít poslední, všechny, které bych spojil někde níže, se chovají stejně.

☻

2.1 Obarvení grafu

Řekneme, že $W \subseteq V(G)$ je označkována množinou značek, pokud je ta množina tvořena právě značkami vrcholů z W.

Na vstupu máme sestavovací strom, chceme zjistit, jaká je barevnost.

Ukládáme si obarvení grafů, ukládáme průniky s označkováním. Každé podmnožině značek přiřadím počet tříd značek, které používají právě tyto značky.

Pozorování 3 *Velikost tabulky je* ch^{2^k}.

Pro list je to jedno obarvení (jedna barevná třída), vyskytne se jen značka i.

Protože počítáme i s cestou až do kořene, tak tam hrany již jsou. Tedy, přidávání ran už se vzalo do úvahy dříve.

Když mění značky pomocí ρ, tak obarvení zůstávají stejné. Ty dvě se slévají, ta co zmizí bude prázdná.

Disjunktní sjednocení je problematické. To se dělá přes soustavu lineárních rovnic (co s čím spojuji). Je potřeba zahodit ty, které mají hranu ve stejné barevné třídě.

3 $L-1-1$ barvení

To je NP-úplné už pro sériově-paralelní grafy. Budeme zkoumat na grafech s omezeným vrcholovým pokrytím. Ptám se, jestli existuje obarvení celými čísly od 0 do λ takové, že sousedi nejsou stejní a ani ob dva, tak také. Je totéž, jako barvení v G^2. Problém je, že λ je součástí vstupu.

Chceme grafy, které mají vrcholové pokrytí velikosti maximálně k. Je vidět, že stromová šířka je nejvýše toto. Dokonce by se z uzlů nemusel dělat strom, ale jen cesta.

Máme graf G s vrcholovým pokrytím nejvýše k. Mimo W může být hodně vrcholů, ale jsou si ekvivalentní, pokud mají stejné sousedy ve W.

Je-li c nějaké $L-1-1$ obarvení G, potom každá barevná třída c protíná každou množinu $I_j := \{u \in V(G) - W; (u_i, w_i) \in E(G) \Leftrightarrow i \in J\}$ nejvýše v 1 vrcholu. Lze tyto barevné třídy rozdělit podle toho, do jakých I_j zasáhnou a který vrchol ve W obsahují.

Pokud J a J' mají neprázdný průnik, potom X_k (počet barev, které zasahují právě do nich) je 0.

Algoritmus 1:

Napřed určíme množiny I_j pro všechna $J \subseteq \{1, \ldots, k\}$. Zkusíme obarvit W nejvýše k barvami. Uděláme částečné rozšíření do sjednocení I.

Udělá se nějaký lineární celočíselný program, ten se použije.

☺

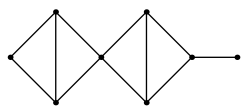
4 Průnikové grafy kruhů

Je to uzavřené na indukované podgrafy.

Rozpoznání grafu jednotkových disků je NP-těžké. Tedy, když nemám reprezentaci, tak je těžké ji vymyslet. Dá se na to přivést NAE-SAT, pomocí rámů (skoroobdélníčky), jsou tam praporky, nesmějí být dva praporky „proti sobě“ – překážely by si. Praporky jsou skoro všude, jen tam, kde je literál v klauzuli, tak to vynechám. Gace jsou nahoře, negace dole, tím rámem jde otáčet nahoru a dolu – tím určím ohodnocení.

4.1 Barevnost

3-obarvitelnost rovinných grafů je NP-těžká i pro stupeň max 4. Rovinné se stupněm max. 4 lze nakreslit do čtvercové mřížky. Udělám si řetízek trojůhelníčků:



Potom můžu dělat hranu, co drží barvu a nakonec dát tu opravdovou hranu. Tím to převedeme na ty rovinné.

Můžeme ale aproximovat – na okolí vrcholu se stejnou barvou může být max. 5 exemplářů. Hladový algoritmus je tedy 5-aproximace, nepotřebuji ani reprezentaci.

Kdybych chtěl 3-aproximaci, tak můžu jít zprava doleva. Zbývají už maximálně 3 sousedi, nalevo mám max. 3 obsazené barvy. To chce ale reprezentaci. Můžu ale brát místo toho můžu brát někoho, kdo má nezávislost nejvýše 3.

V případě různých kružnic vyberee nejenší kroužek, stejný trik jako bez reprezentace.

4.2 Nezávislá množina

Je to těžké.

4.3 Klikovost

Pro každý vrchol určíme nejvzdálenějšího souseda. d bude vzdálenost. Pokusíme se najít kliku takovou, že nějaké vrcholy jsou nevzdálenější v té klice.

Podle nejdelší hrany to rozpůlím na horní a dolní polovinu.

$G - w$ je doplněk bipartitního grafu, hrany mohou vést pouze z horní do dolní poloviny.

4.4 Nezávislá množina

Pro ni existuje Úplné Aproximační Schéma.

Z toho ϵ spočítáme nějaké k, to bude říkat, na jak velké kusy se to musí rozsekat a najít lokální optimum, aby to vyšlo. Představím si, že to máme v mřížce o velikosti políčka 1. Potom beru kusy $k \times k$. Vyzkouším vždycky všechny možné „posunutí“. Pokaždé umažu vše, co se dotýká okrajů kusů (aby zbyly jen vnitřky).

Nezávislé množiny tohoto jde zjistit v $n^{O(k^2)}$.

Potom vezmu takový výsledek, který je nejlepší ze všech posunutí.

Nyní je potřeba dokázat, jak přesně to aproximuje.

Může to vyházet z optimální max to, co je na čárách. Dokážeme, že tam je taková, která nevyháže moc. Každá možnost vodorovná vyškrtá jiné možnosti, každá svislá jiné. To jde podle Holubníka rozpočítat.

Pokud jsou různě velké kruhy, tak pokud mají omezeně velké rozdíly (třeba max. 3*), potom to seběhne stejně.

Jinak to rozdělíme do vrstev, po vrstvách použijeme totéž a výsledky zkombinujeme. V každé vrstvě se průměr kruhů liší maximálně dvojnásobně. Potřebujeme sesynchronizovat mřížky.

Dá se to postavit odspoda nahoru, doplňovat „mezery“ velkých oblastmi malých, které jsou už předpočítané.

4.4.1 Pokus o FPT

Máme dáno k, ptáme se, jestli tam je nezávislá velikosti k.

Když máme malou plochu, tak tam určitě není. Na druhou stranu, když tam najdu hladově (vyberu jeden, zabiju všechny, kterých se dotýká), tak tam určitě je (to je v případě, když je dostatečná plocha).

Zajímavé věci se tedy dějí jen v určitém rozmezí ploch. V takovém případě se použije metoda rozděl a panuj. Prostor štípnu na poloviny (zhruba, podle plochy) a to tak, aby řezaný úsek byl malý, spočítá se řešení na řezu a roztáhnu na obě strany.

Na řezu řešíme hrubou silou, v těch zbytcích (pro každou z nich) se zarekurzí na řez spojené se zbytkem. Počet pokusů na nezávislou v řezu je $2^{O(\sqrt{k}\cdot\log n)}$.

Abych získal malý řez, tak vezmu malou mřížku a vyházím všechno, co nemá žádný kontakt s kruhem. Tohle bude rovinný graf (ty čtverečky), najdu v tom planární separátor.

Budeme ale chtít tokázat, že FPT to nejde, budeme mít problém, jestli je tam klika velikosti k, což je těžký problém. To se udělá speciální mřížka a bude to nějak fungovat.

5 k na cestě

Máme k vrcholů, chceme najít indukovanou cestu, která prochází všema.

Pokud by byla neindukovaná, tak mám $k!$ minorů, ty můžeme najít.

Pokud by to bylo spáru-prosté (neobsahuje indukovaný $K_{1,3}$), tak je to hledání cesty.

Kvazi-line grafy jsou takové, že okolí každého vrcholu lze pokrýt dvěma klikami (tedy, je to něco jako tlustá cesta).

Line grafy, o jsou průnikáče intervalových, které nemají inkluze (že by jeden interval obsahoval celý jiný).

Nalezení indukované cesty na line-grafech je polynomiální. Vyzkouším všechny permutace a potom ke všemu uhádnu předchůdce/následníky (těch důležitých vrcholů). Poté odeberu všechny vrcholy, co neleží na žádné indukované cestě.

Nebude to potom obsahovat žádnou lichou antidíru velikosti alespoň 7 (doplněk indukovaného cyklu). Když tam mám vrchol s okolím 5-cyklem, tak ten vrchol umažu.

Homogenní dvojice klik je A, B taková, že zbytek jsou 4 množiny, taková, která není spojená s ani jednou, taková, která s oběma a dvě, které právě s jednou (když s klikou, tak se všemi v klice).

Když mám takovouhle dvojici, tak obě kliky můžu zkontrahovat.

Quazi-line grafy bez homogenních dvojic klik jsou buď circle-arc-graph a nebo slepenec intervalových pásků.

To potom jde procházet více „napřímo“ (když vlezu do pásku, musím projít, do kliky nesmím znovu). Tak to můžu zmenšit, to už bude line-graph.

6 Rozpoznávání grafů oblouků

Když si představíme, jakou máme reprezentaci – není důležitý přesný konec, jen pořadí, v jakém začínají a končí.

Jednak můžeme předpokládat, že graf na vstupu nemá univerzální vrchol (spojený se všemi – objel bych skoro celé kolo). Dále, liší se co se týče množin sousedů. Dále je to souvislé.

V jakém vztahu jsou dva oblouky v reprezentaci?

- Disjunktní. Hrana patří do doplňku.
- Potkávají se jedním koncem. Zařadím hranu do G_1.
- Potkávají se oběma konci „zvenčí“. Zařadím do G_2.
- Jeden oblouk obsahuje druhý. Zařadím do D_c.

Zapíšeme to maticí, dáme tam buď $1, 2, n, c, t$, když to je v G_1, G_2, doplňku, v D_c nebo v opačném D_c.

Budeme mít operace: Překlopení – necháme konce, oblouk se natáhne druhou stranou. Toto jde udělat i jen na matici.

6.1 Myšlenka algoritmu

Napřed vytvoříme nějakou matici. Poté budeme matici upravovat, bude garantované, že pokud T_4 má reprezentaci, potom část kružnice je nepokrytá, tedy jsou intervalové. Ta překlopení lze udělat zase zpátky.

Při tvorbě matice, D_c jde konstruovat snadno, doplněk také. Problém je rozeznat 1 a 2. Pokud objímají celé, tak musí sousedit dohromady se vším a co nesousedí s jedním musí být uvnitř druhého.

Napřed vyrobíme T_1, ta má vrchol stupně $O(m/n)$ takový, že všichni sousedé obsahují právě jeden konec oblouku tohoto vrcholu. Nejdříve najdeme vrchol v minimálního stupně. Pokud v nemá žádnou hranu v G_2, pak OK. Když mám v relaci uvnitř, tak překlopím a zlepšilo se to. Pokud ale má dvojkové relace, potom vezmu souseda v té relaci, který není pod obloukem něčeho jiného. Toho vezmu, překlopím a dám jako výsledek.

Rozdělíme si na sousedy a nesousedy tohoto vrcholu. Rozdělím do komponent, v každé překlopím jednu partitu – zbavím se dvojek a n.

Pak budu koukat na skupiny, co jsou do sebe vzájemně vnořené. Hledám alespoň jeden konec z jednoho oblouku zasahuje do D_1.