Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Hľadanie perfektných párení

Bakalárska práca

2020

Filip Novák

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Hľadanie perfektných párení

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	RNDr. Ján Mazák, PhD.

Bratislava, 2020
Filip Novák

54559536

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Filip Novák
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Hľadanie perfektných párení
Finding perfect matchings

Anotácia: Náplňou práce je implementovať niekoľko rôznych algoritmov, ktoré nájdu všetky perfektné párenia v grafe, a porovnať ich rýchlosť. Na základe experimentálnych poznatkov získaných z výpočtov pre kubické grafy chceme skombinovať tieto algoritmy tak, aby výsledok bol čo najefektívnejši pre zaujímavé konkrétne triedy snarkov.

Cieľ: Cieľom je získať prehľad v existujúcich algoritmoch pre hľadanie perfektných párení v grafoch. Vybrané algoritmy treba implementovať, experimentálne overiť ich výkonnosť na kubických grafoch a prípadne skombinovať do optimalizovaného algoritmu.

Vedúci: RNDr. Ján Mazák, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Dátum zadania: 25.09.2018

Dátum schválenia: 30.10.2019 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

študent vedúci práce

Poďakovanie: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Úvod **1**

1 Základné pojmy a problémy **3**

1.1 Definície 3

1.2 Problémy 4

2 Prehľad rôznych algoritmov **7**

2.1 Tutteho matica 7

2.2 Rabin-Vaziraniho algoritmus 8

2.3 Algoritmus3 9

2.4 Algoritmus4 9

3 Testovanie **11**

3.1 Porovnávanie algoritmov 11

3.2 Výsledok testovania 11

4 Implementácia algoritmu **13**

4.1 Návrh 13

4.2 Funkcie algoritmu 13

5 Zhodnotenie výsledného algoritmu **15**

5.1 Použitie na snarkoch 15

6 Lorem Ipsum **17**

Záver **19**

Zoznam obrázkov

Zoznam tabuliek

Úvod

Kapitola 1

Základné pojmy a problémy

V tejto kapitole sa oboznámime so základnými pojmami, s ktorými budeme pracovať a spomenieme si rôzne problémy v tejto oblasti.

1.1 Definície

Predtým ako sa začneme zapodievať tým, čo je perfektné párenie v grafe, si musíme zadefinovať párenie v grafe [3]. Veľmi úzko súvisí s naším hlavným pojmom.

Definícia 1.1 *Párenie M v grafe X je množina takých hrán, že žiadne dve nemajú spoločný vrchol. Veľkosť párenia je počet hrán v nej.*

Grafom $X = (V, E)$ sa myslí neorientovaný graf, kde V je množina vrcholov a E je množina hrán. Vrchol je pokrytý párením, ak je koncovým bodom jednej z hrán v párení M.

Poznáme dva typy párení, maximálne a perfektné. To čo nás ale bude zaujímať je perfektné párenie [3]. Teraz si ho zadefinujeme a neskôr uvdíme, že aj maximálne párenie je s ním nejak späté, hrá v ňom rolu.

Definícia 1.2 *Perfektné párenie M alebo 1-faktor je párenie, ktoré pokrýva každý vrchol grafu X.*

To znamená, že každý vrchol grafu X je incidentný presne s jednou hranou párenia. Grafom $X = (V, E)$ sa znova mýsli neorientovaný graf s množinou vrcholov V a množinou hrán E. Je veľmi dôležité aby graf X mal párny počet vrcholov, lebo grafy s nepárnym počtom nemôžu mať perfektné párenie. Ďalej vieme, že v perfektnom párení sú všetky vrcholy grafu pokryté, čo nám hovorí aj definícia.

Teraz sa dostávame k jednej veci, ktorú sme spomenuli vyššie a tou je maximálne párenie. Jedna z ďaľších vlastností každého perfektného párenia je, že je zároveň vždy aj maximálne. To znamená, že obsahuje najväčší možný počet hrán.

Zostáva nám si objasniť poslednú vec. Keďže táto práca je o hľadaní perfektných párení, tak je dôležité si ujasniť na akých grafoch alebo typoch grafov budeme používať rôzne algoritmy, ktoré budú hľadať tie párenia. Rozhodli sme sa, že to bude na grafoch nazývaných *snarky*, pretože majú zaujímavé vlastnosti a celkovo nie sú až tak dobre preskúmané [9].

Definícia 1.3 *V matematickej oblasti teórie grafov je snark jednoduchý, bez mostov, súvislý kubický graf s chromatickým indexom rovným 4.*

Inak povedané, keby odstránime nejakú hranu graf by to nerozdelilo, každý vrchol má troch susedov a potrebujeme minimálne štyri farby na zafarbenie hrán grafu tak, aby všetky susediace hrany mali rôzne farby.

1.2 Problémy

V tejto časti si povieme o niektorých problémoch páreni vo všeobecnosti. To znamená, že sa pozrieme aj na iné problémy, ale budú súvisieť trocha aj s našou problematikou.

Párenie v bipartitných grafoch

Všeobecný problém perfektného párenia je možné vyriešiť v polynomiálnom čase, ale zas niektoré súvisiace problémy sú NP-úplné. Dá sa dokázať že obmedzená forma problému perfektného párenia pre bipartitné grafy je NP-úplná, kde to obmedzenie sa týka rozdelenia vrcholov grafu. Pre stupne vrcholov tri alebo menej to bude stále NP-úplné. Dokonca existuje polynomiálny algoritmus pre stupne vrcholov obmedzené na dva alebo menej [7].

Problém zisťovania počtu párení

Zistiť počet párení v grafe je #P-úplné a to aj pre bipartitné grafy. Dokonca je #P-úplné aj nájdenie počtu perfektných párení v bipartitných grafoch [8]. Na objasnenie toho prečo je # pred triedou P-úplných problémov, tak je to preto, lebo to označuje triedu zaoberajúcu sa počítaním počtu akceptačných riešení problémov. Potom Kasteleynova veta hovorí, že počet perfektných párení v planárnom grafe možno vypočítať pomocou algoritmu FKT v polynomiálnom čase. Taká malá poznámka, počet párení v grafe sa nazýva *Hosoyov index.*

Problém maximálneho párenia

Tento problém má rôzne algoritmy na rôzne typy grafov. Pri neohodnotených grafoch je tento problém riešený Hopcroft-Karpovým algoritmom v čase $O(E\sqrt{V})$ [1]. Existuje pravdepodobnostný (randomizovaný) algoritmus od Muchy a Sankowského, založený na algoritme rýchleho násobenia matíc, ktorý poskytuje zložitosť $O(V^{2.376})$ [6].

Ako sme už raz spomínali vyššie, že maximálne a perfektné párenie spolu nejako súvisia, tak nám to potvrdzuje aj jedno také porovnávacie tvrdenie. Toto tvrdenie hovorí, že problém hľadania maximálneho párenia nie je ťažší, ako problém hľadania perfektného párenia [6]. Presnejšie povedané, problém určenia maximálneho párenia možno znížiť v randomizovanom čase $O(n^{\omega})$ na problém nájdenia perfektného párenia, kde ω je exponentom najlepšie známeho algoritmu násobenia matíc. Poznáme algoritmy, ktoré si o trochu zlepšili časovú zložitosť, ale nevýhodou je, že fungujú len na určitých typoch grafov alebo fungujú len pre obmedzený počet vstupov.

Taká malá poznámka na koniec. Najviac práce a problematík spojená s píreniami sa uskutočnila už na spomínaných bipartitných grafoch. To sú také grafy, ktorých množinu vrcholov je možné rozdeliť na dve disjunktné množiny tak, že žiadne dva vrcholy z rovnakej množiny nie sú spojené hranou.

Kapitola 2

Prehľad rôznych algoritmov

Táto kapitola sa zaoberá prehľadom rôznych existujúcich algoritmov na hľadanie perfektných párení. Niektoré z nich sa budeme snažiť implementovať.

Riešenie týchto problémov v polynomiálnom čase zostalo dlho nepolapiteľným cieľom, až kým Edmonds neprišiel s prvým algoritmom [2]. Potom bolo nájdených niekoľko ďalších algoritmov ako napríklad od Micaliho a Vaziraniho. Vazirani je jeden z najdôležitejších ľudí v tejto problematike párení. Budeme ho ešte spomínať pri Rabin-Vazirani algoritme, ktorý je jeden zo základných algoritmov.

Rôzne algoritmy, ktoré tu budeme spomínať, tak to budú väčšinou teoretické algoritmy, pri ktorých si ukážeme aj ich pseudokódy. Najskôr začneme tým, že si povieme niečo o Tutteho matici.

2.1 Tutteho matica

Ešte predtým ako začneme hľadať nejaké perfektné párenie v grafe, mali by sme si položiť otázku, že či náš graf vôbec obsahuje nejaké perfektné párenie. S odpoveďou na túto otázku prišiel Tutte. K odpovedi nám dopomôže Tutteho matica, ktorú si vytvoríme pomocou vstupného grafu, v ktorom chceme nájsť perfektné párenie.

Definícia 2.1 *(Tutteho matica). Nech $G = (V, E)$ je neorientovaný jednoduchý graf s $|V| = n$. Potom Tutteova matica z G je $n \times n$ matica T so záznamami:*

$$T[i,j] = \begin{cases} 0, & \textit{if (i, j)} \notin E \\ x_{ij}, & \textit{if (i, j)} \in E \textit{ and } i < j \\ -x_{ij}, & \textit{if (i, j)} \in E \textit{ and } i > j \end{cases}$$

kde x_{ij} sú formálne premenné (nie sú konkretizované konkrétnou hodnotou).

Ako vidíme z definície, týmto spôsobom si vytvoríme našu maticu T, kde jediný problém môže nastať v tom, že nevieme čo znamenajú formálne premenné. Sú to také premenné, ktoré sa používajú v podprograme na označenie jedného z údajov poskytnutých ako vstup do podprogramu alebo inak povedané, sú to premenné použité v definícii funkcie alebo metódy.

Nasledujúce Tutteho tvrdenie, ktoré vyslovíme je jadrom niektorých algoritmov pre perfektné párenie, o ktorých budeme ešte v tejto kapitole počuť. Kde ukázal veľmi elegantné spojenie medzi existenciou perfektného párenia a determinantom príslušnej Tutteho matice. Dôkaz si tu ukazovať nebudeme.

Jeho trvdenie znie, že ak máme neorientovaný jednoduchý graf $G = (V, E)$ s párnym $|V|$ a nech T je jeho príslušná Tutteho matica, tak potom $\det(T) \not\equiv 0$ práve vtedy, keď G má perfektné párenie [4]. Inak povedané, keď platí táto bijekcia:

$$det(T) \not\equiv 0 \iff G \text{ obsahuje perfektné párenie.}$$

Musíme si uvedomiť, že matica T nie je maticou čísel, ale dalo by sa povedať, že je to skôr formálna matica kvôli formálnym premenným, ktoré obsahuje. Determinantom Tutteho matice tiež nie je číslo, ale polynóm premenných v matici. $\mathrm{Det}(T) \equiv 0$ sa myslí, že polynóm je identický nulovému polynómu, kde jeho koeficienty sú nula.

V tejto časti sme sa oboznámili s jednoduchým algoritmom, ktorý nám povie, či náš graf má nejaké perfektné párenie. Stačí vypočítať maticu T, vypočítať jej determinant a následne otestovať, či $\det(T)$ je nulový polynóm alebo nie je. Lenže T je formálna matica, takže výpočet $\det(T)$ môže trvať exponencionálny čas. Nemusíme nutne počítať polynóm zodpovedajúci $\det(T)$, my len potrebujeme otestovať, či ten polynóm je identický nule. Existuje na to efektívny spôsob, ale len za predpokladu, že použijeme pravdepodobnoť (randomizáciu).

V ďalšej časti si ukážeme Rabin-Vaziraniho algoritmus, ktorý využíva, už náš spomínaný algoritmus s Tutteho maticou.

2.2 Rabin-Vaziraniho algoritmus

Konečne sme sa dostali k prvému algoritmu, ktorý nám nájde perfektné párenie. Rabin a Vazirani vyvinuli algoritmus, ktorý využíva rekurzívne Lovászov [5] algoritmus detekcie perfektného párenie. Predpokladajme, že sme zistili, že náš graf G má nejaké perfektné párenie a dokonca sme aj identifikovali hranu e, ktorá patrí do perfektného parenia. Jedno z kľúčových pozorovaní je, že podgraf G' z G vytvorený odstránením hrany e a všetkých susediacich hrán s e, má tiež perfektné párenie. Z toho dostávame, že akékoľvek perfektné párenie z G' v kombinácií s e vytvorí perfektné párenie pre G. Ak by sme mali nejakú metódu, ktorá by nám povedala, či je hrana v perfektnom párení, tak potom by sme mali rekurzívny algoritmus, ktorý by vytvoril perfektné párenie. Sú to Rabin a Vazirani, ktorí prišli s touto metódou, kde práve inverzia Tutteho matice obsahuje informácie o týchto hranách.

Rabin a Vazirani vyslovili tvrdenie, ktorého obsahom bolo, že nech máme neorientovaný jednoduchý graf G majúci perfektné párenie a T by bola toho grafu príslušná matica, tak potom $(T^{-1})_{i,j} \neq 0$ práve vtedy, keď $G - \{i, j\}$ má perfektné párenie. Nakoniec toto tvrdenie aj dokázali [4]. Jediná vec, ktorá nemusí byť jasná je $(T^{-1})_{i,j}$ a ako ju dostaneme. Pre maticu T máme $(T^{-1})_{i,j} = \dfrac{(adj\ T)_{i,j}}{det\ T}$, kde $(T^{-1})_{i,j}$ sa nazýva adjungovaná matica a jej hodnotou je determinant matice T po odstránení i-teho riadku a j-teho stĺpca.

Teraz si ukážeme pseudokód algoritmu popísaného vyššie, ktorý rekurzívne odstraňuje hrany v perfektnom párení [4]. Princíp algoritmu je v celku jednoduchý.

Algorithm 1 Rabin-Vaziraniho algoritmus

$M \leftarrow \emptyset$
while G nie je prázdne **do**
    Vypočítajte T_G a vytvorte inštanciu každej premennej s náhodnou hodnotou z $\{1, ..., n^2\}$
    Vypočítajte T_G^{-1}
    Nájdite i,j také, že $(v_i, v_j) \in G$ a $(T_G^{-1})_{i,j} \neq 0$
    $M \leftarrow M \cup \{(v_i, v_j)\}$
    $G \leftarrow G - \{v_i, v_j\}$
end while
return M

Musíme pripomenúť, že toto je pravdepodobnostný (randomizovaný) algoritmus, rovnako tak ako Lovászov algoritmus, od ktorého si pár vecí požičali. Rabin-Vaziraniho algoritmus dokáže nájsť nejaké perfektné párenie s konštantnou pravdepodobnosťou. Časová zložitoť algoritmu je $O(n^{\omega+1})$ na nájdenie perfektného párenie, pretože inverzia matice sa vypočíta v čase $O(n^{\omega})$ a vo všetkých ďalších operáciach, v každej z $O(n)$ iterácií vždy dominuje inverzia matice.

Dokonca existuje aj zrýchlený Rabin-Vaziraniho algoritmus od Muchy a Sankowského. Tento algoritmus dokáže byť pozmenený tak, že inverzia matice T trvá $O(n^2)$ času a perfektné párenie nájde v čase $O(n^3)$.

2.3 Algoritmus3

2.4 Algoritmus4

Kapitola 3

Testovanie

V tejto kapitole si rozoberieme výsledky na rôznych algoritmoch, dokopy z ktorých získame celkový výsledok testovania .

3.1 Porovnávanie algoritmov

3.2 Výsledok testovania

Kapitola 4

Implementácia algoritmu

V tejto časti popíšeme implementáciu čo najefetívnejšieho algoritmu pre hľadanie perfektných párení.

4.1 Návrh

4.2 Funkcie algoritmu

Kapitola 5

Zhodnotenie výsledného algoritmu

Táto kapitola sa zameriava na hodnotenie nášho výsledneho algoritmu. Budeme tu porovnávať výsledky na rôznych grafoch typu snark.

5.1 Použitie na snarkoch

Kapitola 6

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mollis nisi eget arcu dictum posuere. Praesent ullamcorper malesuada magna ut viverra. Aenean bibendum turpis a sagittis rhoncus. Morbi tristique, dolor a mollis malesuada, eros nibh laoreet massa, placerat tempor purus magna et enim. Fusce tempus, nibh sed vehicula semper, nibh justo semper quam, posuere varius ante arcu ac nunc. Cras tincidunt lacus pretium tellus porta aliquet. Suspendisse faucibus porta dolor ac lobortis. Donec molestie erat nec molestie commodo. Phasellus cursus tempus convallis. Cras nec placerat dui, in congue quam.

Suspendisse eu consectetur ante. Proin dapibus efficitur convallis. Sed viverra, libero vitae tincidunt malesuada, ante felis tempus ipsum, a rhoncus turpis lacus ut arcu. Phasellus tristique non lectus in vehicula. Sed id nibh metus. Duis et magna ac neque mollis volutpat ac non leo. Nulla imperdiet vulputate nisi, eget mattis leo bibendum non.

Maecenas maximus rutrum enim quis cursus. Curabitur dolor erat, bibendum nec facilisis a, congue ac turpis. Nullam ex urna, iaculis ut dui at, auctor dictum lacus. Pellentesque at pellentesque mi. Aliquam pretium vestibulum felis ut facilisis. In hac habitasse platea dictumst. Nam felis mi, convallis at tempus id, faucibus sed odio. Suspendisse sit amet arcu fermentum, lobortis massa ultrices, auctor metus. Nulla eu metus ante. Suspendisse potenti. Sed pellentesque augue et ultricies lobortis. Nunc id lorem sit amet nisl lobortis semper eget ut massa. Nam tristique gravida est, sed pretium ipsum convallis dictum.

Nam urna eros, porttitor et vehicula a, sodales sed est. Vestibulum non porttitor justo, ut pellentesque nisl. Donec a sem nulla. Maecenas mi lacus, consectetur nec lacus quis, mollis convallis nunc. Vestibulum auctor tellus et gravida scelerisque. Sed porttitor consectetur aliquam. Pellentesque tempor rutrum elit id consequat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque volutpat, erat quis scelerisque molestie, ex lectus facilisis ante, ut ultricies sem elit id diam. Fusce sit amet dui nec ex eleifend vehicula vitae eu nisl. Integer lorem elit, tempor et sollicitudin et, blandit vel ante.

Morbi facilisis massa quis dolor pharetra, fringilla volutpat ligula ullamcorper. Praesent blandit pellentesque neque, condimentum porta felis suscipit volutpat. In sit amet nulla maximus, viverra nibh eu, lacinia odio. Sed a odio at purus egestas cursus. Nulla facilisi. Pellentesque non leo mollis ligula consequat volutpat quis in augue. Vivamus luctus diam a felis fringilla, id egestas nibh venenatis. Ut ligula libero, vehicula vel pulvinar et, convallis eget tortor. Donec tincidunt est a nisi rhoncus placerat.

Záver

Literatúra

[1] Norbert Blum. A simplified realization of the hopcroft-karp approach to maximum matching in nonbipartite graphs. Technical report, Citeseer, 1999.

[2] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[3] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[4] Ioana Ivan, Madars Virza, and Henry Yuen. Algebraic algorithms for matching. 2011.

[5] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, volume 79, pages 565–574, 1979.

[6] Marcin Mucha and Piotr Sankowski. Maximum matchings via gaussian elimination. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–255. IEEE, 2004.

[7] David A Plaisted and Samuel Zaks. An np-complete matching problem. *Discrete Applied Mathematics*, 2(1):65–72, 1980.

[8] Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[9] Wikipedia. Snark (graph theory) — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Snark%20(graph%20theory)&oldid=905606780`, 2020. [Online; accessed 23-January-2020].