Ročníkový projekt
Report

Študent: Jakub Vaňko
Vedúci projektu: doc. RNDr. Robert Lukoťka, PhD.

Názov projektu: Hľadanie hamiltonovských a skoro hamiltonovských kružníc v kubických grafoch

Pôvodná špecifikácia projektu (zimný semester): Základným cieľom projektu je prehľadať všetky kubické planárne beztrojuholníkové grafy do 26 vrcholov a vyhľadať v nich dvojice (G, e) s vlastnosťou - žiadna hamiltonovská kružnica v G neobsahuje e (všetky planárne grafy do 26 vrcholov majú hamiltonovskú kružnicu). Na tomto výrazne zredukovanom zozname následne vykonať niekoľko podobných výpočtov (napr. nájsť takmer hamiltonovskú kružnicu vynechávajúcu daný vrchol).

Zmenená špecifikácia projektu (letný semester): Základným cieľom projektu je prehľadať všetky 3-súvislé a 4-súvislé bipartitné grafy do určitého počtu vrcholov, ktorý bude závisieť od dostupnej výpočtovej sily. Pri vyšších počtoch vrcholov grafov by sa prehľadávanie mohlo uskutočňovať niekoľko dní, či týždňov. My sa uspokojíme s časom prehľadávania radovo v hodinách. V každom takomto bipartitnom grafe budeme hľadať hranu, ktorá sa nenachádza v žiadnom hamiltonovskom cykle daného grafu.

Zimný semester

V zimnom semestri som sa oboznámil s novou terminológiou z oblasti teórie grafov a zároveň si pripomenul už nadobudnuté vedomosti. Okrem klasických definícií hamiltonovskej cesty a jej postačujúcich podmienok, či nutnej podmienky, som si naštudoval aj nové pojmy ako duálny, signovaný, či bipartitný graf.

Zoznámil som sa s rôznymi tvrdeniami týkajúcich sa hamiltonovských kružníc ako napr. s Tutteho tvrdením, že každý 4-súvislý planárny graf obsahuje hamiltonovskú kružnicu. Taktiež som sa stretol so zaujímavými typmi grafov ako je napr. Petersenov graf, ktorý je hypohamiltonovský, teda po odstránení jedného zo svojich vrcholov sa stane hamiltonovským.

Zoznámil som sa so spôsobmi ako hľadať hamiltonovskú kružnicu v grafe. Jedná sa o problém, ktorý je NP-úplný. Riešiť sa dá pomocou SAT solveru, alebo aj pravdepodobnostného algoritmu (https://arxiv.org/pdf/1008.0541.pdf). Pre našu potrebu, keďže budeme pracovať s grafmi rozumných veľkostí, nám postačí využívať na hľadanie hamiltonovskej cesty bežný backtracking.

Následne som nakódil jednoduchý algoritmus, ktorý využíva práve backtracking a na grafoch rozumnej veľkosti (zatiaľ je tam test na 5 vrcholoch) nám vypíše všetky hamiltonovské cesty vyskytujúce sa v grafe.

Bližšie som sa zoznámil s Barnett-ovym dohadom (Barnette’s conjecture), ktorý hovorí, že každý bipartitný jednoduchý polyhedron je hamiltonovský. Naštudoval som si aj prácu z roku 1985, ktorá tento odhad potvrdzuje pre takéto grafy veľkosti menšej ako 86 vrcholov (https://www.sciencedirect.com/science/article/pii/0095895685900723).

Taktiež som čas štúdia strávil aj nad problémom farbenia signovaných grafov (https://arxiv.org/abs/1906.05638).

Na záver som ešte nakódil jednoduchý generátor (zvyčajne) kubických grafov – to sú také, kde každý vrchol grafu je stupňa 3, čo znamená, že do každého vrcholu vchádzajú práve 3 hrany. Jedná sa o generátor, kde na zadanom počte vrcholov (pokiaľ chceme kubické grafy, tak musíme zadať párny počet vrcholov) náhodne generujem hrany a následne overujem, či spĺňajú podmienky kubického grafu. Medzi vygenerovanými grafmi sa nám však zriedka objavia aj také, ktoré nie sú kubické. Generátor využijem v ďalšej časti projektu na vygenerovanie veľkej množiny (zväčša) kubických grafov, nebude nám však prekážať, ak sa v danej množine sem-tam vyskytne aj graf, ktorý kubický nie je.

Daný generátor, ako aj kód na hľadanie hamiltonovských ciest v grafoch, nepochybne využijeme v ďalšej časti priebehu projektu po upresnení cieľov s pánom školiteľom na letný semester.

Letný semester

V letnom semestri sme si za cieľ stanovili prehľadať 3-súvislé a 4-súvislé bipartitné grafy a v týchto grafoch nájsť hrany, ktoré sa v ich hamiltonovských cykloch nevyskytnú ani raz. S pánom školiteľom sme sa rozhodli overiť hypotézu, že v takýchto grafoch sa takáto hrana vyskytovať nebude. Počet vrcholov, do ktorých budeme túto hypotézu chcieť overiť, sme na začiatku semestra ešte neboli schopný stanoviť, keďže to sa odvíja od rýchlosti generovania daných grafov. Chceli sme to overiť pre grafy, ktoré vieme vygenerovať a tprehľadať backtrackingom v rozumnom čase (rádovo niekoľko hodín). Napokon sme sa dostali na číslo 38 vrcholov pre 3-súvislé (približne 12 hodín behu programu) a 4-súvislé grafy (približne 7 hodín behu programu).

Objasnenie pojmov:

Súvislým grafom nazveme taký graf, ak medzi ľubovoľnými dvoma vrcholmi daného grafu existuje cesta.

***k*-súvislý** grafom nazveme taký graf, ktorý má viac vrcholov ako *k* a zároveň daný graf ostáva súvislý aj po odstránení menej ako menej ako *k* vrcholov.

Za **bipartitný graf** označíme taký, ktorého množina vrcholov V môže byť rozdelená do dvoch disjunktných množín V_1 a V_2, tak, že každá hrana bude obsahovať jeden vrchol z V_1 a druhý vrchol z V_2.

Duálny graf je graf G*, ktorého vrcholy sú stenami nejakého planárneho grafu G.

Prvou úlohou bolo teda nájsť schopný a efektívny generátor grafov. Náš triviálny generátor zo zimného semestra by bol veľmi neefektívny a navyše sme potrebovali generovať špecifické typy grafov, čo náš primitívny generátor nezvládal, keďže generoval iba kubické grafy (aj to nie so 100% istotou).

Výber padol na generátor grafov **plantri** (https://users.cecs.anu.edu.au/~bdm/plantri/), ktorý nám ponúka aj možnosť generovať 3-súvislé a 4-súvislé kubické bipartitné grafy, ktoré sme mali v pláne prehľadať.

Plantri generuje práve jeden graf z každej triedy izomorfizmu a obrovské množstvo grafov dokáže vygenerovať veľmi rýchlo a efektívne. Pre viac informácii si odporúčam pozrieť návod: (https://users.cecs.anu.edu.au/~bdm/plantri/plantri-guide.txt).

Po vybraní vhodného generátora grafov som sa následne musel s programom dôkladne zoznámiť, vyskúšať si generovať rozličné typy grafov. Spraviť testy rýchlosti generovania s výpisom, bez výpisu, so zápisom do textového súboru, bez zápisu do textového súboru. Rozlíšiť koľko trvá samotný backtracking, koľko trvá samotné generovanie grafov. Vyskúšať beh programu s formátom zápisu grafov v matici susednosti, či s vnútornou Plantri reprezentáciou grafov.

Testy nám poslúžili na to, aby sme si spravili prehľad o tom, do koľko zhruba vrcholov sme schopní prehľadávať grafy tak, aby to zbehlo v relatívne prijateľnom čase.

Ďalšou úlohou bolo do generátora zapracovať náš backtracking v podobe plugin-u, aby po vygenerovaní grafu generátorom sme na danom grafe okamžite spustili náš algoritmus na hľadanie hamiltonovských cyklov. Technické veci implementovania takéhoto plugin-u sú popísané v guide (https://users.cecs.anu.edu.au/~bdm/plantri/plantri-guide.txt), prípadne v samotnom mojom kóde.

Algoritmus obsahujúci backtracking zo zimného semestra som pozmenil. Backtracking som špeciálne prispôsobil na prácu s kubickými grafmi (keďže s inými typmi grafov sme nemali v pláne pracovať), čím sa mi ho podarilo zefektívniť. Pri riedkych grafoch nemalo zmysel uvažovať prácu s maticou susednosti, napriek tomu som spravil zopár testov na rýchlosť behu programu a hypotéza o tom, že čas bude pri matici výrazne vyšší sa mi potvrdila.

Plantri vo vnútri svojho programu pracuje s následovným formátom ukladania grafov: Jedná sa o pole, kde na 0-tom indexe sa nachádza informácia o počte vrcholov a následne v poli sa nachádzajú susedia vrcholov. Susedia sú oddelení 0.

Ukážka formátu:
5 3 4 0 3 5 0 1 4 5 2 0 1 5 3 0 2 3 4 0

Vzhľadom na to, že pracujeme s grafmi relatívne nie veľkých veľkostí, backtracking sa javil ako ideálny spôsob hľadania hamiltonovských cyklov. Keďže tento ročníkový projekt je súčasťou bakalárskej práce, tak v bakalárskej práci sa bližšie pozerám aj na iné spôsoby hľadania hamiltonovských cyklov ako napr. SAT solver.

Po implementácii backtrackingu sme následne chceli overiť správnosť fungovania nášho algoritmu, preto na potvrdenie správneho behu môjho programu som vypracoval jednoduché testy v súbore test.c. Máme k dispozícii graf kocky, ktorý je kubický, o tomto grafe vieme, že obsahuje 12 orientovaných hamiltonovských cyklov, ktoré sme si ručne uložili do poľa. Následne som spustil môj algoritmus a overoval, či nájde presne týchto 12 hamiltonovských cyklov.

Na otestovaní mám aj známy Petersenov graf, ktorý síce obsahuje hamiltonovskú cestu, ale neobsahuje žiaden hamiltonovský cyklus. Petersenov graf je najmenší kubický graf bez mosta, ktorý neobsahuje žiadnu hamiltonovský cyklus. Môj algoritmus v tomto prípade teda nemá nájsť žiadny hamiltonovský cyklus.

Testy sú potrebné pre overenie správnosti fungovania môjho algoritmu, hlavne počas vývoja softvéru, kedy dochádza k zmenám kódu, je potrebné pravidelne testovať softvér.

Záver

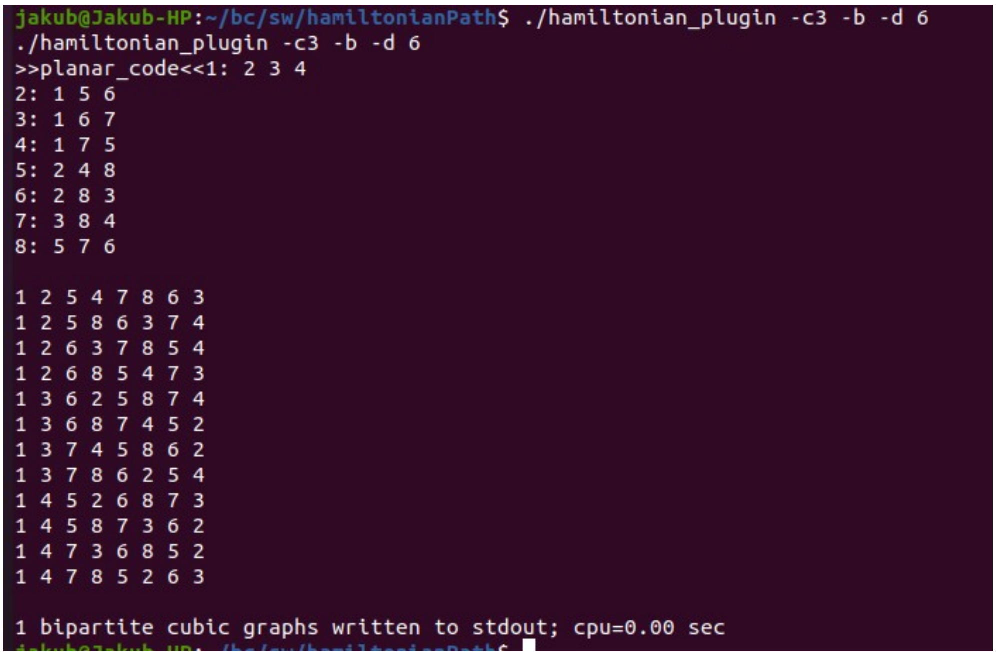
Následne som spustil náš algoritmus na daných grafoch. U 3-súvislých a aj u 4-súvislých bipartitných kubických grafoch sme sa dostali na počet vrcholov 38. Algoritmus bežal na 3-súvislých grafoch približne 13 hodín, u 4-súvislých to bolo približne 7 hodín. Čas sa určite predĺžil aj s dôvodu, že dané grafy sme si zapisovali do textového súboru (časť grafov na ukážku zapísaných v textovom súbore je dostupná na mojej stránke).
Pre kompiláciu môjho plugin-u s plantri, si prosím stiahnite súbory z môjho repozitára (na GitHube však nájdete už aj skompilovaný program). Robil som jemné úpravy v programe plantri.c, preto je potrebné mať túto moju verziu.
Pre skomplilovanie použite príkaz:

cc -o hamiltonian_plugin -O4 '-DPLUGIN="filter.c"' plantri.c cycles.c

Ukážka spustenia behu programu (generovanie 3-súvislých bipartitných grafov o 8 vrcholov) :

./hamiltonian_plugin -b -c3 -d 6

Pre viac informácii odporúčam nazrieť do plantri guide (https://users.cecs.anu.edu.au/~bdm/plantri/plantri-guide.txt), kde nájdete vysvetlenie jednotlivých “switchov” a ako docieliť generovanie potrebných grafov.


```
jakub@Jakub-HP:~/bc/sw/hamiltonianPath$ ./hamiltonian_plugin -c3 -b -d 6
./hamiltonian_plugin -c3 -b -d 6
>>planar_code<<1: 2 3 4
2: 1 5 6
3: 1 6 7
4: 1 7 5
5: 2 4 8
6: 2 8 3
7: 3 8 4
8: 5 7 6

1 2 5 4 7 8 6 3
1 2 5 8 6 3 7 4
1 2 6 3 7 8 5 4
1 2 6 8 5 4 7 3
1 3 6 2 5 8 7 4
1 3 6 8 7 4 5 2
1 3 7 4 5 8 6 2
1 3 7 8 6 2 5 4
1 4 5 2 6 8 7 3
1 4 5 8 7 3 6 2
1 4 7 3 6 8 5 2
1 4 7 8 5 2 6 3

1 bipartite cubic graphs written to stdout; cpu=0.00 sec
```
Ukážka behu programu

Pozor: môj algortimus je primárne upravený na prácu s kubickými grafmi, preto generovanie iných grafov v kombinácii s mojim algoritmom nedáva zmysel.

Pre nás sú dôležité prípony -c3 alebo -c4, -b, -d (spraví z grafu dual, to však je aplikované iba na output, ja si jednotlive vygenerované grafy mením na dual priamo v kóde, takže vieme sa zaobísť aj bez tejto prípony) a číslo n určujúce počet vrcholovneduálneho grafu. (počet vrcholov duálu, teda nášho bipartitného grafu je n*2 – 4).

Program vypíše na štandardný output graf v čitateľnej forme a všetky hamiltonovské cykly daného grafu. V samotnom kóde je možné si zapnúť aj výpis do textového súboru output.txt, ktorý bude kopírovať konzolový výpis.

Ak by sa nachádzala v grafe naša žiadaná hrana, tak ju vypíše tiež. Okrem nášho výpisu, program vypíše graf aj v plantri formáte (ktorý však nie je čitateľný, to sa dá zmeniť zmenou prípony, opäť sa odkazujem na guide).

Program bol nastavený tak, že ak nájde požadovanú hranu, tak do textového súboru vypíš danú hranu spoločne s grafom, v ktorom sa táto hrana vyskytla. Počas nášho behu programu však táto situácia nenastala, čo značí, že žiadna takáto hrana v prehľadávaných grafoch neexistovala.

Na mojej stránke sa nachádza aj textový súbor c4bipartite-38.txt, ktorý obsahuje malú časť vygenerovaných 4-súvislých bipartitných grafov ako ukážka. Vzhľadom na veľkosť súboru všetkých vygenerovaných grafov som sa rozhodol zverejniť malú časť z veľkého množstva grafov, ktoré sa nám podarilo vygenerovať.

Výsledok:

Hypotézu sme pre bipartitné grafy do 38 vrcholov potvrdili, žiadnu hranu takú, cez ktorú by neprechádzal žiaden hamiltonovský cyklus sme nenašli.