

Generovanie oblique planárnych triangulácií

Obhajoba

Matúš Matok

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského

20. júna 2021

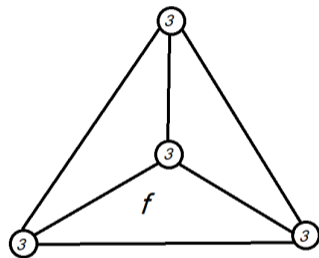
Školiteľ: RNDr. Ing. František Kardoš, PhD.

Teoretický základ

- V práci používame štandardnú terminológiu teórie grafov.
- Taktiež používame terminológiu, ktorá nie je bežne známa každému so základnými znalosťami v teórii grafov, ktorú spolu s ďalšími užitočnými pojmami objasníme a pripomenieme na nasledujúcej strane.

Teoretický základ

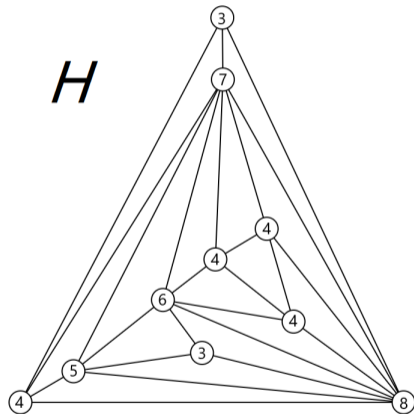
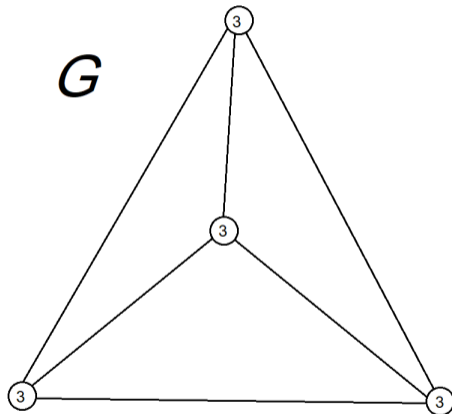
- *Planárny (rovinný) graf* je graf, pre ktorý existuje nakreslenie do roviny také, že sa žiadne dve hrany nepretnú.
- *Planárna triangulácia* je planárny graf, v ktorom každá stena susedí s práve 3 rôznymi vrcholmi.
- Typ steny f je n -tica (v našom prípade iba trojica) $\langle d_1, \dots, d_n \rangle$ stupňov vrcholov susediacich so stenou f .



Obr.: Planárna triangulácia v ktorej f je typu $\langle 3, 3, 3 \rangle$.

Teoretický základ

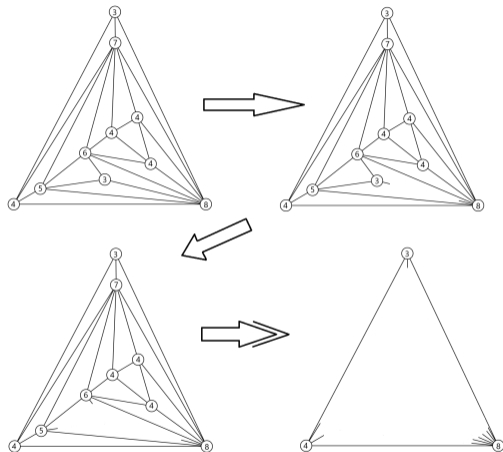
- *Oblique graf* je graf, v ktorom sa nenachádzajú dve steny rovnakého typu.



Obr.: *F* nie je oblique graf, zatiaľ čo *H* oblique je. Navyše, *H* je oblique planárna triangulácia.

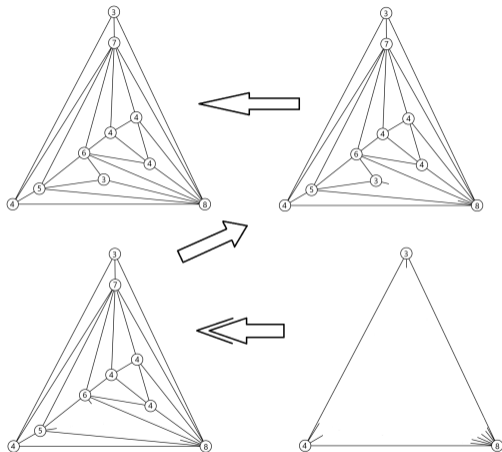
- Navrhnuť algoritmus schopný nájsť oblique planárne triangulácie pre rôzne parametre.
- Základný parameter je Δ , ktorý označuje maximálny stupeň vrchola v hľadanom grafe.
- Prvý z konštrukčných cieľov je nájsť všetky oblique planárne triangulácie pre čo najvyššie Δ .
- Druhý konštrukčný cieľ je nájsť nejakú oblique planárnu trianguláciu pre čo najvyššie Δ .

- V minulosti skúmané.
- Hansjoachim Walther, Margit Voigt, Jens Schreyer, František Kardoš a Jozef Miškuf.
- Práce vymedzili teoretické obmedzenia pre oblique planárne triangulácie.

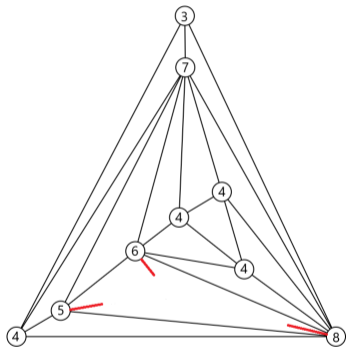


Obr.: Podgrafy oblique planárnej triangulácie

Algoritmus



Obr.: Generovanie oblique planárnej triangulácie za predpokladu vlastníctva nedeterministického Turingovho stroja



- *Polhrana* je hrana, ktorá je susedná s práve jedným vrcholom. (Na obrázku červenou farbou.)
- Každý vrchol v má *label* λ , pre ktorý platí $\lambda = d(v) + \#\text{polhrán incidentných s } v$. Cyklická postupnosť hrán a polhrán incidentných s v je fixná.
- *Otvorená stena* je stena, v ktorej sa nachádza aspoň jedna polhrana. Ostatné steny nazývame *zatvorené*.

Obr.: Situácia v podgrafe oblique planárnej triangulácie

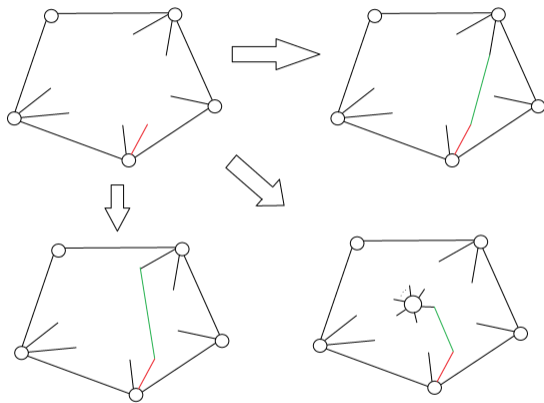
Algoritmus je navrhnutý ako DFS.

- Výber operácie a operandov,
- vykonanie operácie,
- kontrola, či je graf v "dokončiteľnom" stave,
- návrat do pôvodného stavu.

Typy operácií

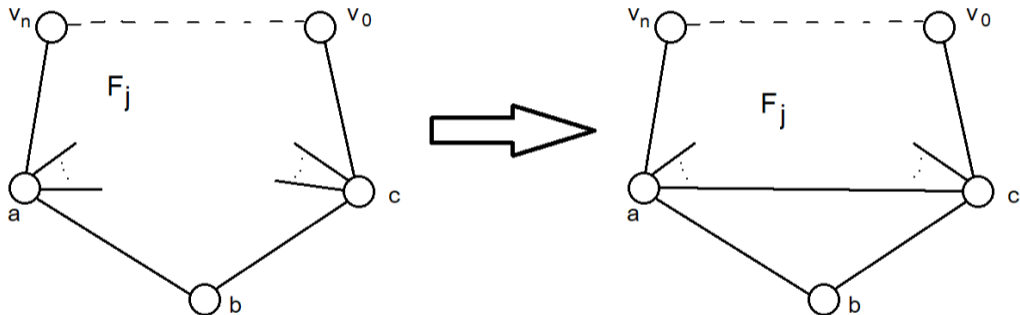
- Nedeterministické (vetviace) operácie.
 - Spájanie,
 - pridávanie.
- Deterministická (nútené) operácia.
 - Deterministické spájanie.

Nedeterministické operácie



Obr.: Možné vytvorenia hrany z červenej polhrany v otvorenej stene.

Deterministická operácia



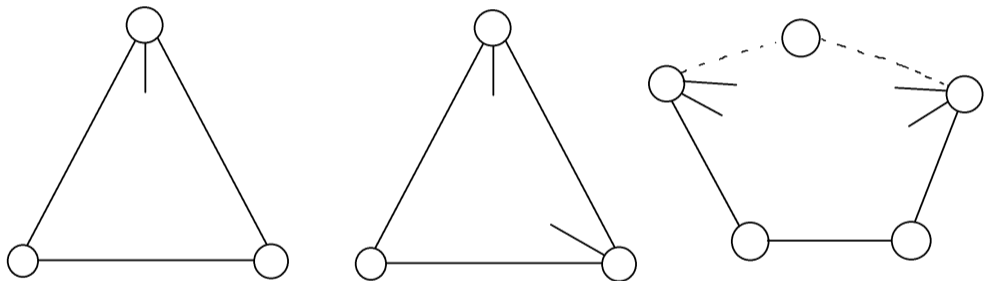
Obr.: Takzvaná deterministická situácia vyžadujúca deterministické spojenie.

Výber operácie

- V prípade, že graf obsahuje vrchol, ktorý nemá v niektorej stene, s ktorou susedí žiadne polhrany, vykonaj deterministické spojenie.
- V opačnom prípade, môžeme vybrať ľubovoľnú polhranu a spojiť ju s so všetkými vhodnými polhranami v danej stene.

Kontrola správnosti

Je potrebné skontrolovať, či sa graf nedostal do stavu, v ktorom vieme s istotou povedať, že ho nie je možné doplniť na oblique planárnu trianguláciu.

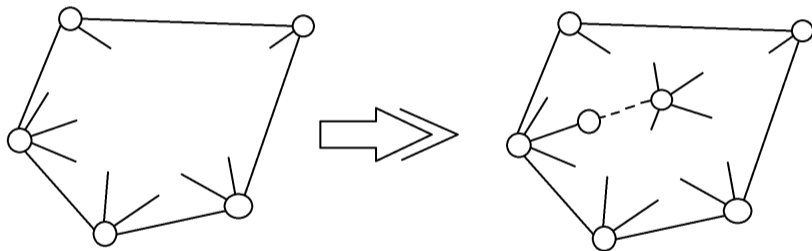


Obr.: Príklady stien, ktoré nie sú uzatvorené.

V prípade, že sme pridali novú zatvorenú stenu, je nutné skontrolovať, či sa taká stena v grafe už nenachádza.

Zložitosť

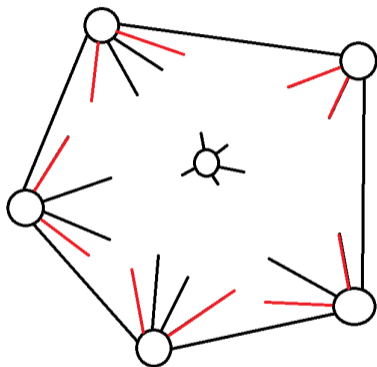
Takto popísaný algoritmus vykonáva DFS na nejakom (zatiaľ nekonečnom) prehľadávacom strome. Našou snahou bolo tento strom spraviť čo najmenší a konečný.



Obr.: Príklad potenciálne nekonečnej vetvy

Záruka konečnosti

- Naším cieľom bude pridať v každom kroku rekurzie aspoň jednu zatvorenú stenu.
- Deterministické spájanie zaručuje ako jediné prídanie novej uzatvorenej steny.
- Ostatné operácie musíme vykonávať tak, aby viedli k deterministickému spájaniu.



Obr.: Príklad polhrán zabezpečujúcich deterministické spojenie v ďalšom kroku

Orezanie stromu do hĺbky

- Toto sa už z časti deje, keďže pre nejaké Δ máme len konečný počet stien, ale vieme to spraviť oveľa efektívnejšie.
- Keďže naším cieľom je vygenerovať oblique planárnu trianguláciu, vieme vypočítať koľko vrcholov daného stupňa sa môže v takom grafe pre nejaké Δ nachádzať.

$$|V_k| = \begin{cases} \lfloor \frac{(\Delta-3)(\Delta-4)}{6} \rfloor, & \text{if } k = 3 \\ \lfloor \frac{(\Delta-3)(\Delta+2)}{2k} \rfloor, & \text{if } k > 3, \end{cases}$$

kde k je stupeň vrchola a $|V_k|$ je maximálny počet vrcholov daného stupňa.

- To nám dáva prvý spôsob ako orezať prehľadávací strom do hĺbky.

Orezanie stromu do hĺbky

- Ďalší spôsob je odvodený z Eulerovho vzorca.

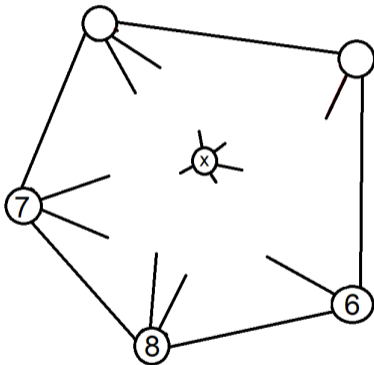
$$\sum_{v \in G} (6 - \deg(v)) = 12$$

- Úpravami dostávame nasledovnú rovnicu.

$$3n_3 + 2n_4 + n_5 = 12 + n_7 + 2n_8 + \dots + (\Delta - 6)n_\Delta.$$

Orezanie stromu do šírky

- Úzko súvisí s výberom polhrany.



- Graf už obsahuje steny $\langle 4, 7, 8 \rangle$, $\langle 5, 7, 8 \rangle$ a $\langle 3, 6, 8 \rangle$.
- Je menej dostupných stien typu $\langle x, 7, 8 \rangle$ ako $\langle x, 6, 8 \rangle$.

Obr.: Situácia pre pridávaním nového vrchola

Modifikovaný algoritmus

- Zavedenie nového parametra, ktorý obmedzí počty vrcholov vyšších stupňov (> 6). Týmto spôsobom obmedzíme počet možných stien, čím zrýchlime algoritmus.
- Cena, ktorú za to platíme, je že dostaneme veľmi špecifické grafy, to avšak nie je problém pre náš cieľ.
- Kľúčové je odhadnúť ako zvoliť obmedzenia na počty vrcholov pre dané Δ .

Modifikovaný algoritmus - vplyv parametrov

Δ	restrictions	time	Δ	restrictions	time
18	$2 \times 7, 2 \times 8, 1 \times 9$	180s	20	$3 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	1.8s
18	$3 \times 7, 2 \times 8, 1 \times 9$	4.5s	20	$3 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	20s
18	$3 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	3s	20	$4 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	68s
18	$4 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	12s	20	$4 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	5s
18	$4 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	3s	20	$3 \times 7, 2 \times 8, 1 \times 9$	> 200s
18	$3 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	1.8s	21	$3 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	3.5s
19	$3 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	9s	21	$4 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	19s
19	$3 \times 7, 2 \times 8, 1 \times 9, 1 \times 10$	< 1s	21	$3 \times 7, 2 \times 8, 2 \times 9, 1 \times 10$	> 200s

Výsledky pôvodného algoritmu

- Programu sa podarilo nájsť všetky oblique planárne triangulácie pre $\Delta = 8$. Beh trval približne 10 sekúnd. Týchto grafov je 11.
- Programu sa podarilo za 4 hodiny nájsť 1500 neizomorfných oblique planárnych triangulácií pre $\Delta = 9$, pričom nepoužíval operáciu nedeterministického spájania. Toto rozhodnutie bolo vykonané z časových dôvodov.
- Myslíme si, že algoritmus by bol v konečnom čase schopný nájsť všetky oblique planárne triangulácie pre $\Delta = 9$.
- Pre $\Delta > 9$ je vysoko nepravdepodobné, že by sme boli schopní nájsť všetky oblique planárne triangulácie.

Výsledky modifikovaného algoritmu

- Podarilo sa nám nájsť aspoň jednu oblique planárnu trianguláciu pre $\Delta \leq 26$.
- Pre $\Delta = 26$ hľadanie trvalo približne 12 hodín. Pre $\Delta = 27$ program po 2 dňoch nič nenašiel.