Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Efektívne hľadanie minimálne dominujúcej množiny na reálnych sieťach

Magisterská práca

2014 **Viktor Tomkovič**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Efektívne hľadanie minimálne dominujúcej množiny na reálnych sieťach

Magisterská práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky FMFI
Vedúci práce: Mgr. Martin Čajági

Bratislava, 2014 **Viktor Tomkovič**

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Viktor Tomkovič
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský

Názov: Efektívne hľadanie minimálnej dominujúcej množiny na reálnych sieťach

Cieľ: Prvým z cieľov práce je implementovať aktuálne úplne efektívne algoritmy poskytujúce riešenia pre problém minimálnej dominujúcej množiny s rôznymi obmedzeniami na danú množinu. Ako je napríklad súvislosť, či minimálna vzdialenosť medzi vrcholmi v pokrytí. Tieto budú slúžiť ako benchmark pri menších grafoch. Druhým krokom je implementovať heuristické algoritmy, aproximatívne algoritmy a urobiť vzájomný pomer efektívnosť (čas-priestor) / veľkosť chyby plus porovnanie oproti úplným algoritmom z prvého cieľu.
Tretím cieľom je vyvinúť čo najefektívnejšie algoritmy pracujúce na sieťach v rádoch desiatok až stá tisícov vrcholov a miliónov hrán a otestovať ich na dátach reálnych existujúcich sietí.
Pod najneefektívnejším sa rozumie zlepšenie pre konkrétne typy sietí v čase alebo v chybe. Z nášho pohľadu je prioritnejší čas, pretože cieľom je rýchlo získavať požadované informácie aby sme ich vedeli spracovať a poskytnúť ako vstup pre ďalšie aplikácie.

Vedúci: Mgr. Martin Čajági
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. PhDr. Ján Rybár, PhD.

Dátum zadania: 04.12.2012

Dátum schválenia: 04.12.2012 prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie

Ďakujem.

Abstrakt

Táto práca skúma rôzne algoritmy na riešenie problému hľadania minimálnych dominantných množín (MDS). Konkrétne skúma ich využitie v sieťach malého sveta.
Kľúčové slová: minimálna dominujúca množina, MDS, algoritmy a dátové štruktúry, siete malého sveta.

Abstract

This work is.

Keywords: minimal dominating set, MDS, algorithms and date structures, small-world network.

Obsah

Úvod

Toto je úvod do mojej diplomovej práce. Bude doplnený neskôr.

Kapitola 1

Definície

V tejto kapitole sme zaviedli niektoré pojmy, s ktorými sa budeme stretávať počas nasledujúcich kapitol. Sú to prevažne pojmy z teórie grafov. Keďže väčšina článkov, ktorými sme sa zaoberali v ďalších častiach je anglického pôvodu, rozhodli sme sa pre značenie uprednostniť knihu Graph Theory (Diestel 2000) pred knihou Grafové algoritmy (Plesník 1983).

Základným pojmom je pre nás množina, čo je súbor navzájom rôznych objektov. Množinu prirodzených čísel vrátane nuly označujeme $\mathbb{N}$. Množinu celých čísel označujeme $\mathbb{Z}$. Množinu reálnych čísel $\mathbb{R}$. Pre reálne číslo x označujeme hornú celú časť $\lceil x \rceil$ a označuje najmenšie celé číslo väčšie alebo rovné ako x. Podobne dolnú celú časť označujeme $\lfloor x \rfloor$ a označuje najväčšie celé číslo menšie alebo rovné ako x. Základ logaritmov napísaných ako „log“ je 2 a základ logaritmov napísaných ako „ln“ je e. Množina $=\{A_1, \ldots, A_k\}$ navzájom disjunktných podmnožín množina A je *rozdelenie* ak $A = \bigcup_{i=1}^{k} A_i$ a všetky i platí, že $A_i = \emptyset$.

1.1 Grafy

Graf $G = (V, E)$ je usporiadaná dvojica množiny vrcholov a množiny hrán, ktorá má následujúce vlastnosti:

- $V \cap E = \emptyset$ (hrany a vrcholy sú rozlíšiteľné)
- $E \subseteq \{\{u, v\} : u \neq v; u, v \in V\}$ (hrana spája dva vrcholy)

Graf sa zvyčajne znázorňuje nakreslením bodov pre každý vrchol a čiar medzi dvoma bodmi tam, kde existuje hrana. Rozmiestenie bodov a čiar nemá význam.

O grafe s množinou vrcholov V hovoríme, že je grafom *na* V. *Množina vrcholov grafu* G je označovaná $V(G)$ a to aj v prípade, kedy graf G má za množinu vrcholov inú množinu ako V. Napríklad, pre graf $H = (W, F)$ označujeme množinou vrcholov $V(H)$ a platí $V(H) = W$. Podobne označujeme *množinu hrán grafu* $E(G)$ (v hore uvedenom príklade platí $E(H) = F$). Pre jednoduchosť hovoríme, že vrchol (hrana) patrí grafu a nie množine vrcholov (hrán) grafu a preto sa občas vyskytuje označenie $v \in G$ a nie $v \in V(G)$.

Rád grafu je počet vrcholov v grafe a označujeme ho ako $|G|$. *Prázdny graf* $(\emptyset, \emptyset)$ označujeme $\emptyset$. Grafy rádu 0 alebo 1 sa označujeme ako *triviálne*.

Vrchol v je incidentný s hranou e ak platí $v \in e$. Hranu $\{x, y\}$ jednoduchšie označujeme ako xy. *Hranami* $X - Y$ označujeme množinu hrán $E(X, Y) = \{\{x, y\} : x \in X, y \in Y\}$. Namiesto $E(\{x\}, Y)$ píšeme $E(x, Y)$ a podobne aj namiesto $E(X, \{y\})$ píšeme $E(X, y)$. s Zápisom $E(v)$ označujeme $E(v, V(G))$ a hovoríme o *hranách vrchola* v.

Dva vrcholy x, y grafu G sú *susedia*, ak existuje hrana xy v grafe G. Dve hrany $e \neq f$ sú susedné, ak majú spoločný jeden vrchol. Ak sú všetky vrcholy v grafe navzájom susedné, graf je *kompletný* (alebo *úplný*). Kompletný graf s n vrcholmi označujeme K^n.

Majme dva grafy $G = (V, E)$ a $G' = (V', E')$. Potom $G \cup G' := (V \cup V', E \cup E')$. Podobne $G \cap G' := (V \cap V', E \cap E')$. Ak platí $G \cap G' = \emptyset$ tak hovoríme, že grafy sú *disjunktné*. Ak platí $V \subseteq V'$ a $E \subseteq E'$, tak potom je graf G' *podgrafom* grafu G. Zapisujeme $G' \subseteq G$.

Ak $G' \subseteq G$ a G' obsahuje všetky hrany $xy \in E$ pre $x, y \in V'$, tak hovoríme, že graf G' je *indukovaný podgraf* grafu G. Taktiež hovoríme, že V' *indukuje* G' na G a zapisujeme $G' =: G[V']$. Zápis $G[\{v\}]$ skracujeme na $G[v]$.

Ak je U nejaká množina vrcholov, tak zápisom $G - U$ (operátory $-$ a $\setminus$ budeme občas zamieňať) označujeme $G[V(G) \setminus U]$. Inými slovami graf $G - U$ dosiahneme tak, že z grafu G vymažeme všetky vrcholy z množiny U a všetky incidentné hrany k nim. Pre $G - \{u\}$ používame aj zápis $G - u$. Pre graf $G = (V, E)$ a množinu hrán $F = \{xy : x, y \in V\}$ zapisujeme $G + F = (V, E \cup F)$ a $G - F = (V, E \setminus F)$.

Komponent grafu je taký maximálny podgraf, kde medzi každou dvojicou vrcholov existuje cesta.

1.2 Vlastnosti grafu

Uvažujme o grafe $G = (V, E)$. Množinu susedov vrchola v označujeme $N_G(v)$. Pokiaľ to bude z kontextu jasné, tak iba skrátene $N(v)$. Zápis rozšírme na množiny. Pre množinu $U \subseteq V$ zapisujeme $N(U)$ množinu susedov všetkých vrcholov $u \in U$. Množinu $N(U)$ nazývame *susedmi* U. *Susedov vrátane vrchola* označujeme susedov vrchola s vrcholom samotným. Pre vrchol v susedov vrátane vrchola zapisujeme ako $N[v] := N(v) \cup \{v\}$. Podobne môžeme rozšíriť zápis aj na množiny. *Pokrytím* množiny $U \subseteq V$ nazývame množinu $N[U] := N(U) \cup U$.

Číslo $d_G(v) = d(v) := |E_G(v)|$ sa nazýva *stupeň* vrchola. Je to počet susedov vrchola (neplatí pre digrafy, multigrafy a iné zložité grafy, ktorými sa tu však nezaoberáme). Vrchol stupňa 0 je *izolovaný*. Číslo $\delta(G) := min\{d(v), v \in G\}$ je *minimálny stupeň* grafu G. Podobne číslo $\Delta(G) := max\{d(v), v \in G\}$ je *maximálny stupeň* grafu G.

1.3 Cesta a cyklus

Cesta je graf $P = (V, E)$ v tvare:

$$V = \{x_0, x_1, x_2, x_3, \ldots, x_k\} \qquad E = \{x_0x_1, x_1x_2, x_2x_3, \ldots, x_{k-1}x_k\},$$

kde všetky vrcholy x_i sú navzájom rôzne. Vrcholy x_0 a x_k sa nazývajú *konce* cesty a zvyšné vrcholy sú *vnútorné* vrcholy. Cestu zjednodušene označujeme sledom vrcholov: $P = x_0x_1x_2 \cdots x_k$. Aj keď nevieme rozlíšiť medzi cestami $P_1 = x_0x_1x_2 \cdots x_k$ a $P_2 = x_kx_{k-1}x_{k-2} \cdots x_0$, často si zvolíme jednu možnosť a hovoríme o *ceste z* x_0 *do* x_k (v tomto prípade sme si vybrali cestu P_1). *Dĺžka cesty* je číslo k (cesta môže mať dĺžku 0).

Cyklus je cesta $P = (V, E)$ v tvare

$$V = \{x_0, x_1, x_2, x_3, \ldots, x_{k-1}\} \qquad E = \{x_0x_1, x_1x_2, x_2x_3, \ldots, x_{k-2}x_{k-1}, x_{k-1}x_0\}$$

O ceste má zmysel hovoriť iba ak $k \geq 3$. *Dĺžka cyklu* je číslo k. Je to počet vrcholov (a zároveň aj hrán) v grafe.

V nasledujúcej časti si ukážeme dátové štruktúry, s ktorými sme v práci pracovali.

1.4 Strom

Strom je súvislý graf, ktorý má n vrcholov a $n - 1$ hrán.

TODO: treba uviesť základnú vlastnosť a nejaké kecy o tom, čo a ako

1.5 Toky

Veľa vecí z reálneho sveta sa dá modelovať pomocou grafov alebo štruktúr podobných grafom. Ide napríklad o elektrickú rozvodnú sieť, cestnú sieť, vlakovú/dráhovú sieť, komunikačnú sieť. Pri týchto znázorneniach vystupujú vždy dvojice komunikácií a „križovatiek“ (elektrické vedenie s trafostanicami, cesty s mestami, dráhy so zástavkami, linky s prepojovacími stanicami). Každá komunikácia má svoju kapacitu. V týchto štruktúrach má zmysel sa pýtať otázky, ako napríklad koľko veľa prúdu, zásob, dát dokáže prúdiť medzi dvoma vrcholmi. V teórii grafov hovoríme o tokoch.

Konkrétne komunikáciu si môžeme predstaviť ako hranu $e = xy$, ktorá vyjadruje aj smer prúdenia. K usporiadanej dvojici (x, y) môžeme priradiť hodnotu k vyjadrujúcu kapacitu komunikácie. Znamená to, že k jednotiek môže prúdiť z vrcholu x do vrcholu y. Alebo usporiadanej dvojici (x, y) môžeme priradiť zápornú hodnotu $-k$ a to znamená, že k jednotiek prúdi opačným smerom. To znamená, že pre zobrazenie $f : V^2 \to \mathbb{Z}$ (množina V označuje vrcholy), bude platiť, že $f(x, y) = -f(y, x)$, keď x a y sú susedné vrcholy.

Keď už máme vrcholy a komunikácie, musíme mať aj *zdroj* vecí, ktoré po komunikáciach budú

prúdiť a taktiež miesta, z ktorých budú tieto veci odchádzať z modelu. Tie sa označujú ako *stoky*. Okrem týchto špeciálnych vrcholov platí, že

$$\sum_{y \in N(x)} f(x, y) = 0$$

Pokiaľ platia pre graf $G := (V, E)$ a zobrazenie $f : V^2 \to \mathbb{Z}$ vlastnosti $f(x, y) = -f(y, x)$ pre susedné vrcholy x a y a pre vrcholy mimo zdrojov a stôk, že $\sum_{y \in N(x)} f(x, y) = 0$, tak budeme hovoriť o *toku* na grafe G.

Graf sme si zadefinovali ako štruktúru, ktorá má hrany *neorientované*, to znamená, že nevieme rozlíšiť, kde hrana „začína" a kde „končí". Pri tokoch sme ale začali rozlišovať túto vlastnosť a tým sa hrany stali *orientovanými*. Grafy sa nazývajú neorientované, pokiaľ sú aj ich hrany neorientované a naopak, ak sú orientované hrany, tak hovoríme aj o orientovanom grafe. V ďalšom texte budeme orientovanosť uvádzať iba vtedy, keď nebude jasne vyplávať z kontextu.

1.6 Siete

Ďalšie veci, ktoré môžeme pri modelovaní sietí z reálneho sveta skúmať sú veci ohľadne štruktúry. Má zmysel sa pýtať na to, aký je priemer grafu, či vieme určiť hierarchiu vrcholov a jej, aké komunikácie treba prerušiť na to, aby sa sieť rozpadla, kam treba nasadiť obmedzený počet špiónov, aby sme získali čo najviac informácií a podobne.

Pri modeloch reálnych sietí má zmysel zaoberať sa nielen stupňami jednotlivých vrcholov ale aj distribúciou stupňov a priemerným stupňom vrchola. *Priemerný stupeň grafu* $G = (V, E)$ označujeme $\overline{d_G} = \bar{d}$ a vypočítame ako:

$$\overline{d_G} = \bar{d} = \frac{\sum_{v \in V} d_v}{|V|}$$

K zadefinovaniu distribúcie budeme potrebovať ešte jednu funkciu. Táto funkcia vracia hodnotu 1 v bode 0 a vo všetkých ostatných bodoch vracia hodnotu 0. Takže dobre slúži ako filter. Označme ju τ a zadefinujme:

$$\tau(n) = \begin{cases} 1 & \text{ak } n = 1 \\ 0 & \text{inak} \end{cases}$$

Distribúcia stupňov vrcholov grafu G je funkcia $p(d)$ reprezentujúca pravdepodobnosť toho, že vrchol má stupeň d. Platí, že:

$$p(d) = \frac{\sum_{v \in V} \tau(d - d_v)}{|V|}$$

TODO: vysvetli, čo to znamená..

Suma vo funkcii prechádza všetkými vrcholmi a vďaka filtračnej funkcii τ spočíta, koľko je vrcholov stupňa d.

Zaujímavou vlastnosťou grafu je *klasterizačný koeficient vrchola*. Je definovaný ako pomer počtu hrán

medzi susediacimi vrcholmi daného vrchola a všetkými možnými (aj potenciálnymi) hranami medzi susediacimi vrcholmi. Formálne, pre graf $G := (V, E)$ je *klasterizačný koeficient* vrchola v hodnota:

$$c_v = \frac{|E(N(v))|}{\binom{|N(v)|}{2}}$$

Priemerný klasterizačný koeficient $\bar{c}$ grafu G je definovaný ako:

$$\bar{c} = \frac{\sum_{v \in V}^{c_v}}{|V|}$$

Podobne ako distribúciu stupňov vrcholov zadefinujeme aj distribúciu klasterizačných koeficientov. *Distribúcia klasterizačných koeficientov grafu* G je funkcia $c(d)$, ktorá hovorí o tom, aký je priemer klasterizačných koeficientov pre všetky vrcholy stupňa d. Vypočítame ho ako:

$$c(d) = \frac{\sum_{v \in V} \tau(d - d_v) c_v}{\sum_{v \in V} \tau(d - d_v)}$$

V následujúcich kapitolách sme sa zamerali a prebrali si jednotlivé existujúce algoritmy, ktoré boli implementované.

	naive	greedy	greedyQ	ch7alg33	ch7alg34OT	ch7alg35OT	fnaive	fproper
ba10	0.085	0.001	0.002	0	0.003	0.004	0.008	0.009
ba18	1.284	0.001	0.002	0	0.013	0.008	0.035	0.035
ba20	4.44	0.001	0.002	0	0.011	0.008	0.049	0.047
ba100	-	0.015	0.004	0.004	0.07	0.045	0.473	0.506
ba200		0.026	0.006	0.009	0.135	0.073	30.515	30.412
ba1000		0.087	0.045	0.037	0.883	0.512		
ba2000		0.142	0.095	0.053	2.524	0.988		
ba10k		1.844	0.362	0.54	45.781	6.662		
ba20k		8.813	1.294	3.8		22.256		
ba100k		66762	68.669	135.197				
rnd10		0.001	0.002				0.003	0.003
rnd15		0.001	0.002				0.048	0.043
rnd20		0.001	0.002				0.107	0.112
rnd100		0.011	0.004				1293.811	1302.94
rnd200		0.032	0.008					
rnd1000		0.072	0.043					
rnd2000		0.114	0.096					
rnd10k		2.484	0.44					
rnd20k		11.281	1.671	4.947				
zly10		0.005	0.003	0.003	0.04	0.018	0.119	0.117
zly20		0.022	0.01	0.022	0.098	0.036	0.809	0.814
zly100		0.09	0.151	0.21	6.213	2.107		

	naive	greedy	greedyQ	ch7alg33	ch7alg34OT	ch7alg35OT	fnaive	fproper	macov
ba10	0.085	0.001	0.002	0	0.003	0.004	0.008	0.009	1
ba18	1.284	0.001	0.002	0	0.013	0.008	0.035	0.035	1
ba20	4.44	0.001	0.002	0	0.011	0.008	0.049	0.047	1
ba100	-	0.015	0.004	0.004	0.07	0.045	0.473	0.506	1
ba200		0.026	0.006	0.009	0.135	0.073	30.515	30.412	1
ba1000		0.087	0.045	0.037	0.883	0.512			2
ba2000		0.142	0.095	0.053	2.524	0.988			
ba10k		1.844	0.362	0.54	45.781	6.662			
ba20k		8.813	1.294	3.8		22.256			
ba100k		66762	68.669	135.197					1
rnd10		0.001	0.002				0.003	0.003	1
rnd15		0.001	0.002				0.048	0.043	2
rnd20		0.001	0.002				0.107	0.112	3
rnd100		0.011	0.004				1293.811	1302.94	3
rnd200		0.032	0.008						5
rnd1000		0.072	0.043						
rnd2000		0.114	0.096						5
rnd10k		2.484	0.44						5
rnd20k		11.281	1.671	4.947					5
zly10		0.005	0.003	0.003	0.04	0.018	0.119	0.117	5
zly20		0.022	0.01	0.022	0.098	0.036	0.809	0.814	5
zly100		0.09	0.151	0.21	6.213	2.107			5

Kapitola 2

Popis softvéru

V predchádzajúcich kapitolách sme si popísali niektoré dátové štruktúry a vybrané algoritmy. Hlavnou náplňou je ich testovanie.

2.1 Analýza existujúcich riešení

V marci roku 2015 o žiadnych podobných riešeniach nevieme.

2.2 Špecifikácia požiadaviek

TODO: Čo očakávame od softvéru a prečo je viac fajn ako konkurencia.

2.2.1 Požiadavky

TODO: Čo má softvér robiť z hladiska funkčnosti.

2.2.2 Prevádzkové požiadavky

TODO: Čo má softvér robiť z hladiska hardvéru.

2.3 Návrh

TODO: Stručný opis, čo chceme dosiahnuť.

2.3.1 Technické požiadavky

TODO: Self-descripting

2.3.2 Používateľské požiadavky

TODO: Self-descripting

2.3.3 Použité technológie a návrhové vzory atď.

TODO: Self-descripting

2.3.4 Rozhranie aplikácie

TODO: Vstup/výstup – aj pre jednotlivé časti softvéru

Kapitola 3

Implementácia softvéru

V tejto kapitole postupne uvedieme realizáciu návrhu popísaného v kapitole 1. Popíšeme ako sme implementovali jednotlivé algoritmy a ukážeme si štruktúru a ovládanie aplikácie. Na záver uvedieme ako dopadlo testovanie softvéru ().

Kapitola 4

Dosiahnuté výsledky

TODO: Tu budú všetky tie faily, cez ktoré som si prešiel.

Záver

V práci sme popísali niektoré algoritmy a porovnali ich implementácie. Fomin (Fomin, Grandoni a Kratsch 2005) napísal dobrú prácu... ...

Literatúra

Diestel, Reinhard (2000). *Graph theory. 2000*.

Fomin, Fedor V, Fabrizio Grandoni a Dieter Kratsch (2005). "Measure and conquer: domination–a case study". In: *Automata, Languages and Programming*. Springer, str. 191–203.

Plesník, Ján (1983). *Grafové algoritmy*.