Lokalna Gostost

Maruša Povhe, Neva Port, Matej Pirnat, Špela Podgoršek

Fakulteta za matematiko in fiziko

24. 5. 2012

Popolnoma goste grupe-klike

Definicija

Naj bo $G = (V, E)$ neusmerjen graf. Podmnožici $U \subseteq V$ pravimo **klika** natanko tedaj, ko je $G[U]$ poln graf.

Klika U je **maksimalna klika** v grafu $G = (V, E)$ natanko tedaj, ko v G ne obstaja klika U', za katero velja $U \subset U'$.

Klika je maksimalna klika v grafu G natanko tedaj, ko ima maksimalno moč med vsemi klikami v G.

Popolnoma goste grupe-klike

Strukturne lastnosti klik:

1. Klike so popolnoma goste, to pomeni, če je U klika velikosti k, potem je $\delta(G[U]) = \Delta(G[U]) = k - 1$. Višja stopnja ni mogoča.
2. Klike so popolnoma kompaktne, kar pomeni, da je $diam(G[U]) = 1$. Krajša razdalja med katerimakoli dvema točkama ni mogoča.
3. Klike so popolnoma povezane, torej, če je U klika velikosti k, potem je $(k - 1)$ - točkovno povezana in $(k - 1)$ - povezana. Višja povezanost ni mogoča.

Popolnoma goste grupe-klike

Izrek (Turan,1941)

Naj bo $G = (V, E)$ neusmerjen graf. Če je $m > \frac{n^2}{2} \cdot \frac{k-2}{k-1}$, potem obstaja klika velikosti k znotraj G.

V splošnem imamo lahko primer, da maksimalni kliki U_1, U_2 obstajata, če zadoščata pogojema: $U_1 \neq U_2$ in $U_1 \cap U_2$ neprazen.

Izrek (Moon and Moser, 1965)

Vsak neusmerjen graf G z n točkami ima največ $3^{\lceil \frac{n}{3} \rceil}$ maksimalnih klik.

Popolnoma goste grupe-klike

Družina vseh klik določenega grafa nam kaže neko strukturo:

1. Klike so zaprte za izključitev, to pomeni, če je U klika v G in $v \in U$, potem je tudi $U - \{v\}$ klika.
2. Klike so ugnezdene, to pomeni, da vsaka klika velikosti n vsebuje kliko velikosti $n-1$ (celo n klik velikosti $n-1$).

Popolnoma goste grupe-klike

Naj bo $G = (V, E)$ neusmerjen graf, U podmnožica točk in $N > 0$ neko naravno število.

1. U pravimo **N-klika** natanko tedaj, ko je za vsak $u, v \in U,\ d_G(u, v) \leq N$.
2. U pravimo **N-club** natanko tedaj, ko je $diam(G[U]) \leq N$.
3. U pravimo **N-clan** natanko tedaj, ko je U maksimalna klika in je $diam(G[U]) \leq N$.

Reševanje problemov

Imamo algoritme s časovno zahtevnostjo $O(n+m)$, ki nam pomagajo pri reševanju problemov:

1. Določijo, če je dana množica točk $U \subseteq V$ klika v G. Preprosto testiramo ali je vsak par točk iz U povezan v G. To je do $\binom{n}{2}$ parov, vendar tudi če imamo manj povezav, smo po testiranju m parov gotovi.
2. Določijo, če je dana klika $U \subseteq V$ maksimalna v G. Preprosto testiramo ali obstaja točka v $V - U$, ki je sosednja vsem točkam v U. Zopet smo v najslabšem primeru konec po m testiranjih.

Predpostavimo, da je množica točk grafa $G = (V, E)$ urejena. Pravimo, da je množica $U \subseteq V$ leksikografsko manjša kot množica $U' \subseteq V$ natanko tedaj, ko prva točka, ki ni v U in U', pripada U. Sledi:

3. Poiščejo leksikografsko najmanjšo maksimalno kliko, ki vsebuje neko kliko U'. Na začetku določimo $U := U'$ in ponavljamo za vse $v \in V - U$, v naraščajočem vrstnem redu ter testiramo za vsak v ali je $U \subseteq N(v)$. Če to velja, potem dodamo točko v v U. Ko končamo je U maksimalna klika, ki vsebuje U'. Časovna zahtevnost algoritma je $O(n + m)$.

Iskanje maksimalnih klik

Exhaustive search algoritem:

- Naštejemo vse možne kandidate množic $U \subseteq V$.
- Preverimo, če je U klika.
- Algoritem vrne največjo najdeno kliko.
- Ocena zgornje meje časovne zahtevnosti je v najslabšem primeru $O(n^2 \cdot 2^n)$.

Računsko je iskanje maksimalne klike težek problem.
Problem: KLIKA
Vhodni podatki: Graf G, parameter $k \in \mathbb{N}$
Vprašanje: Ali obstaja klika velikosti vsaj k znotraj G?

Iskanje maksimalnih klik

- $\omega(G)$ označuje velikost maksimalne klike grafa G.
- Če imamo algoritem, ki določa problem iskanja klike v času $T(n)$ z iskanjem na podlagi binarnega drevesa, potem lahko izračunamo $\omega(G)$ s časovno zahtevnostjo $O(T(n) \cdot \log n)$.
- Vsak $T(n)$ algoritem za računanje $\omega(G)$ nam da $T(n)$ algoritem za določanje problema iskanja klike.

Izrek

Problem iskanja klike je $\mathcal{N}P$-poln problem.

Posledica.

Brez privzetka $\mathcal{P} = \mathcal{NP}$, ne obstaja algoritem s polinomsko časovno zahtevnostjo za iskanje klike velikosti k v grafu, kjer vemo, da klika velikosti k obstaja.

Iskanje maksimalnih klik

Boljši algoritmi z eksponentno časovno zahtevnostjo:

- Poskusimo oblikovati hitre algoritme s super-polinomsko časovno zahtevnostjo.
- Exhaustive search algoritem nam da zgornjo mejo časovne zahtevnosti $O(n^2 \cdot 2^n)$, oziroma $O^*(2^n)$, kadar opustimo polinomske faktorje.
- Naš cilj je oblikovati algoritme s časovno zahtevnostjo $O^*(\beta^n)$, s čim manjšo možno β.

Izrek

Obstaja algoritem za iskanje maksimalne klike s časovno zahtevnostjo $O^*(1.3803^n)$.

Aproksimiranje velikosti maksimalnih klik

- Velikost največje klike v grafu G označimo z $\omega(G)$.
- Pravimo, da algoritem aproksimira $\omega(G)$ s pomočjo faktorja $f(n)$ natanko tedaj, ko algoritem pri danem vhodnem grafu G vrne kliko U iz G tako, da je $\omega(G) \leq f(n) \cdot |U|$.

Izrek

Obstaja algoritem s polinomsko časovno zahtevnostjo, ki vrne za graf G z n točkami kliko s faktorjem $O\left(\frac{n}{\log n^2}\right)$ za velikost $\omega(G)$.

Izrek

Brez privzetka $\mathcal{NP} = \mathcal{ZPP}$ ne obstaja algoritem s polinomsko časovno zahtevnostjo, ki za graf G z n točkami vrne kliko s faktorjem $n^{1-\varepsilon}$ za velikost $\omega(G)$ za vsak $\varepsilon > 0$.

Iskanje klik s fiksno velikostjo

- Velikost klike ni del vhodnih podatkov.
- Exhaustive search algoritem ima časovno zahtevnost $\Theta(n^k)$, kadar je velikost klike k fiksna.

Izrek

Obstaja algoritem za iskanje trikotnikov v grafu s časovno zahtevnostjo $O(n^{2.376})$.

Izrek

Za vsak $k \geq 3$ obstaja algoritem za iskanje klike velikosti k v grafu z n točkami s časovno zahtevnostjo $O(n^{\beta(k)})$, kjer je $\beta(k) = (\lfloor \frac{k}{3} \rfloor, \lceil \frac{(k-1)}{3} \rceil, \lceil \frac{k}{3} \rceil)$ in kjer ima množenje $n^r \times n^s$ matrike z $n^s \times n^t$ matriko časovno zahtevnost $O(n^{\alpha(r,s,t)})$.

Tabela nam kaže kaj pridobimo z uporabo matričnega množenja.

Velikost klike	**Exhaustive search**	**Matrično množenje**
3	$O(n^3)$	$O(n^{2.376})$
4	$O(n^4)$	$O(n^{3.376})$
5	$O(n^5)$	$O(n^{4.220})$
6	$O(n^6)$	$O(n^{4.751})$
7	$O(n^7)$	$O(n^{5.751})$
8	$O(n^8)$	$O(n^{6.595})$

Izrek

Za vsak $k \geq 3$ obstaja algoritem, ki šteje koliko je klik velikosti k katerim pripada vsaka točka iz grafa na n točkah, s časovno zahtevnostjo $O(n^{\beta(k)})$, kjer je $\beta(k) = (\lfloor \frac{k}{3} \rfloor, \lceil \frac{(k-1)}{3} \rceil, \lceil \frac{k}{3} \rceil)$

Štetje maksimalnih klik danega grafa G

Definicija: Skupni polinomski čas oz. Polinomska časovna zahtevnost algoritma

Algoritem vrne vseh C možnih konfiguracij v času, ki je omejen s polinomom v spremenljivki C, velikost vhodnih podatkov pa je enaka n.

Zahtevnost štetja maksimalnih klik v danem grafu G
Razred $\#\mathcal{P}$ je oznaka za razred vseh funkcij, ki štejejo rešitve $\mathcal{NP}$-problemov.
Problem štetja maksimalnih klik danega grafa G je $\mathcal{P}$-poln problem. Če obstaja algoritem za štetje maksimalnih klik grafa s polinomsko časovno zahtevnostjo, potem je ta problem iskanja maksimalnih klik P-poln problem in posledično velja $\mathcal{P} = \mathcal{NP}$.

Definicija: Polynomial delay - Polinomska zakasnitev

Algoritem, ki ima to lastnost generira konfiguracije eno za drugo v nekem vrstnem redu tako, da je časovni zamik od pričetka delovanja algoritma do prve konfiguracije, časovni zamik med katerimakoli dvema nadaljnjima konfiguracijama in časovni zamik med zadnjo vrnjeno konfiguracijo algoritma do časa, ko se algoritem ustavi, omejen s polinomom.

Štetje maksimalnih klik danega grafa G

Izrek

Za štetje vseh maksimalnih klik danega grafa G obstaja algoritem, ki ima lastnost polinomske zakasnitve $O(n^3)$ in zavzame zgolj $O(n+m)$ prostora.

Definicija: Leksikografska urejenost

Naj bo $(M, \preceq)$ linearna urejenost. Leksikografsko urejenost $\leq_{lex}$ na množici $M \times M$ definiramo takole

$$(m_1, n_1) \preceq_{lex} (m_2, n_2) \equiv m_1 \prec m_2 \vee (m_1 = m_2 \wedge n_1 \preceq n_2)\ .$$

Specified order - Določen vrstni red

Izrek

Naj bo G poljuben graf in U maksimalna klika tega grafa. Ugotavljanje ali v danem grafu obstaja maksimalna klika U' , ki je leksikografsko večja od maksimalne klike U, je $\mathcal{NP}$- poln problem.

Posledica

1. Brez privzetka $\mathcal{P} = \mathcal{NP}$, ne obstaja algoritem, ki bi za dani graf G in poljubno maksimalno kliko U tega grafa, generiral v leksikografskem vrstnem redu naslednjo maksimalno kliko v polinomskem času.
2. Brez privzetka $\mathcal{P} = \mathcal{NP}$, ne obstaja algoritem, ki bi za poljuben graf G generiral vse maksimalne klike v obratnem leksikografskem vrstnem redu s polinomsko zakasnitvijo.

Algorithm 1: Leksikografsko štetje maksimalnih klik

Input : Graf $G = (V, E)$
Output: Zaporedje maksimalnih klik grafa G v leksikografskem vrstnem redu.

Naj bo U_0 leksikografsko prva maksimalna klika;
Dodaj U_0 v prednostno vrsto Q;
while *Q ni prazna množica* **do**
    $U :=$ minimum(Q);
    output U;
    foreach *točko v_j grafa G, ki ni sosednja s točko $v_i \in U$ tako, da je $i < j$* **do**
        $U_j := U \cup \{v_1, \ldots, v_j\}$;
        if *$(U_j - \overline{N}(v_j)) \cup \{v_j\}$ je maksimalna klika grafa $G[\{v_1, \ldots, v_j\}]$* **then**
            Naj bo T leksikografsko najmanjša maksimalna klika, ki vsebuje $(U_j - \overline{N}(v_j)) \cup \{v_j\}$;
            Dodaj T v Q

Izrek

Algoritem 1 prešteje vse maksimalne klike danega grafa G z n točkami v leksikografskem vrstnem redu. Časovna zahtevnost algoritma je enaka $O(n^3)$.

Strukturno goste množice - Plexi

Definicija: Plexi

Naj bo $G = (V, E)$ poljuben neusmerjen graf in $N \in \{1, \dots, n-1\}$ naravno število. Podmnožici $U \subseteq V$ pravimo N-plex danega grafa G natanko tedaj, ko velja $\delta(G[U]) \geq |U| - N$.

- Klika je 1-plex. Vsak N-plex je hkrati tudi $(N+1)$-plex.
- Maksimalni N-plex danega grafa G
- Vsak podgraf N-plexa je tudi N-plex.

Strukturno goste množice - Plexi

Trditev

Naj bo $N \in \{1, \dots, n-1\}$naravno število in naj bo $G = (V, E)$ poljuben neusmerjen graf na n-točkah.

1. Če je V N-plex, kjer je $N < \frac{n+2}{2}$, potem je $diam(G) \leq 2$. V primeru, da velja tudi $n \geq 4$, potem je G 2-povezan graf.
2. Če je V N-plex, kjer je $N \geq \frac{n+2}{2}$ in je G povezan graf, potem je $diam(G) \leq 2N - n + 2$.

Dokaz 1. točke:

- Naj bo $N < \frac{n+2}{2}$.
- $u, v \in V$ tako, da velja $u \neq v$

- u in v sosednji točki $\Longrightarrow$ razdalja med njima je enaka 1
- u in v nista sosednji točki, razdalja med njima naj bo vsaj tri $\Longrightarrow N(u) \cap N(v) = \emptyset$, velja zveza

$$n-2 \geq \left|N(u) \cup N(v)\right| \geq 2\delta(G) \geq 2(n-N) > 2\left(n-\frac{n+2}{2}\right) = n-2$$

To je protislovje. $\Longrightarrow$ Razdalja med u in v je največ dva, zato velja $diam(G) \leq 2$.

- Sedaj moramo pokazati še, da če je $n \geq 4$, je graf G 2-povezan.
- Privzemimo nasprotno. Recimo, da v grafu G obstaja most. To je takšna povezava e grafa G, da če to povezavo odstranimo je graf $G - \{e\}$ sestavljen iz dveh povezanih komponent V_1 in V_2.
- Vsaka najkrajša pot od točke v množici V_1 do točke v množici V_2 mora vsebovati povezavo e.
- Ker velja $diam(G) \leq 2 \implies$ ena od komponent V_1, V_2 je singelton.
- Za stopnjo vsake točke iz V mora veljati:

$$n - N > n - \frac{n+2}{2} = \frac{n-2}{2} \geq 1 \ ,$$

kar je protislovje. Od tod sledi, da takšna povezava e v grafu G ne more obstajati.

Dokaz 2. točke:

- Recimo, da je $N \geq \frac{n+2}{2}$.
- Naj bo $\{v_0, v_1, \ldots, v_r\}$ najdaljša najkrajša pot grafa G.
- Ker ne obstaja krajša pot med točko v_0 in točko v_r velja, da točka v_i ni sosednja s točkami $v_0, \ldots, v_{i-2}, v_{i+2}, \ldots, v_r$ za $\forall i \in \{0, \ldots r\}$.
- Poleg tega ne more obstajati točka, ki bi bila sosednja z obema točkama v_0 in v_3. Zato velja naslednja zveza:

$$\{v_0\} \cup \{v_2, v_3, \ldots, v_r\} \cup \Big(N(v_3) - \{v_2, v_4\}\Big) \subseteq \overline{N}(v_0)\ .$$

- Opazimo, da imamo na levi strani disjunktno unijo množic. Sledi neenakost:

$$1 + (r-1) + d_G(v_3) - 2 \leq N\ .$$

Od tod sledi: $r + n - N - 2 \leq N$ in $diam(G) = r \leq 2N - n + 2$.

Zahtevnost iskanja N-plexa v danem grafu G

Problem: N-plex
Vhodni podatki: graf G, parameter $k \in \mathbb{N}$
Vprašanje: Ali obstaja N-plex velikosti najmanj k v grafu G?

Izrek

Iskanje N-plexa v danem grafu G je $\mathcal{NP}$-poln problem za vsa naravna števila $N > 0$.

Strukturno goste množice - Cores = Jedra

Definicija

Naj bo $G = (V, E)$ poljuben neusmerjen graf. Pravimo, da je podmnožica $U \subseteq V$ N-jedro natanko tedaj, ko velja $\delta(G[U]) \geq N$.

- N je red jedra
- Maksimalno N-jedro ali Glavno N-jedro
- Vsako $(N+1)$-jedro je tudi N-jedro in vsako N-jedro je $(n-N)$-plex.
- Če sta U in U' N-jedri, potem je tudi $U \cup U'$ N-jedro.

Strukturno goste množice - Cores = Jedra

Trditev

Naj bo $G = (V, E)$ poljuben neusmerjen graf in naj bo $N > 0$ poljubno naravno število. U in U' naj bosta maksimalni, povezani N-jedri v grafu G tako, da velja $U \neq U'$. Potem v grafu G ne obstaja povezava med U in U'.

Posledice zgornje Trditve:

- Edino maksimalno N-jedro danega grafa G je unija vseh njegovih maksimalnih povezanih N-jeder.
- Maksimalno 2-jedro povezanega grafa je povezano.
- Graf je gozd natanko tedaj, ko ne vsebuje nobenega 2-jedra.

Strukturno goste množice - Cores = Jedra

Trditev

Naj bo $G = (V, E)$ poljuben neusmerjen graf in naj bo $N > 0$ poljubno naravno število. Če rekurzivno odstranjujemo vse točke s stopnjo strogo manjšo od N in vse povezave, ki so povezane z njimi, potem preostala množica točk U tvori maksimalno N-jedro.

Definicija: Številka jedra

Številka jedra za točko $v \in V$ je definirana kot:

$$\xi_G(v) =_{\text{def}} \max\{N \mid \text{obstaja } N\text{-jedro } U \text{ v grafu } G \text{ tako, da velja } v \in U\}.$$

Algorithm 2: Izračun jedrnih števil

Input : Graf $G = (V, E)$
Output: Vektor ξ_G, ki vsebuje jedrna števila vseh točk grafa G.
Izračunamo stopnje vseh točk in jih shranimo v D;
Množico točk V sortiramo v naraščajočem vrstnem redu stopenj iz množice D;
foreach $v \in V$ *v sortiranem vrstnem redu* **do**
    $\xi_G(v)$:=D[v];
    foreach *točko u, ki je sosednja s točko v* **do**
        if $D[u] > D[v]$ **then**
            $D[u] := D[u] - 1$;
            Presortiramo množico V v naraščajočem vrstnem redu stopenj iz množice D.

Najbolj enostavne izvedbe algoritma imajo v najslabšem primeru časovno zahtevnost enako $O(mn\ log(n))$.

Izrek

Obstaja izvedba Algoritma 2, ki nam izračuna jedrna števila vseh točk v danem grafu G=(V,E), ki ima n točk in m povezav, v času $O(n+m)$.

Trditev

Za vsak $N > 0$ lahko maksimalno N-jedro za graf z n točkami in m povezavami izračunamo v času $O(n+m)$, kar je neodvisno od N.

Statistično goste grupe

Definicija

Naj bo $G = (V, E)$ nek neusmerjen graf z n točkami in m povezavami. Gostota grafa G, $\varrho(G)$ je definirana na sledeč način:

$$\varrho(G) =_{def} \frac{m}{\binom{n}{2}}$$

Gostota grafa predstavlja število opaženih, ulomljeno s številom vseh možnih povezav v neusmerjenem grafu.

Gosti podgrafi

Definicija

Naj bo $G = (V, E)$ neusmerjen graf in naj bo $0 \leq \eta \leq 1$ realno število. Podmnožica $U \subseteq V$ naj predstavlja η-gost podgraf, natanko tedaj, ko $\varrho(G[U]) \geq \eta$.

V η-gostem podgrafu sta poljubni dve točki povezani z verjetnostjo najmanj η. Kljub temu pa je možno, da imajo celo grafi s precej veliko gostoto izolirane točke.
Klika je 1-gost podgraf.

Gosti podgrafi

Trditev

Naj bo $0 \leq \eta \leq 1$ *realno število.* η*-gost graf* G *velikosti* k *vsebuje* η*-gost podgraf velikosti* $k-1$*.*

Sprehodi

Gostost lahko opazujemo tudi glede na število opažnih sprehodov poljubne dolžine v grafu.

Definicija

Naj bo $G = (V, E)$ nek neusmerjen graf z n točkami. Naj bo $l \in \mathbb{N}$ dolžina sprehoda. Za točko $v \in V$ definiramo njeno stopnjo reda l kot število sprehodov dolžine l, ki se začnejo v točki v, $d_G^l(v)$. Določimo $d_G^0(v) = 1$ za vse $v \in V$.

Jasno je, da je $d_G^1(v)$ stopnja točke $v \in G$.

Trditev

Naj bo $G = (V, E)$ neusmerjen graf. Za vse $l \in \mathbb{N}$ in vse $r \in \{0, ..., l\}$ je število sprehodov dolžine l $W_l(G) = \sum_{v \in V} d_G^r(v) \cdot d_G^{l-r}(v)$.

Sprehodi

Jasno je, da je največje število sprehodov dolžine l v grafu z n točkami $n(n-1)^l$. Sledi, da lahko definiramo gostoto reda l na naslednji način

$$\varrho_l(G) =_{def} \frac{W_l(G)}{n(n-1)^l}$$

Vemo: $\varrho_1(G) = \varrho(G)$

Trditev

Drži, da $\varrho_l(G) \leq \varrho_{l-1}(G)$ za vsak graf G in vsa naravna števila $l \geq 2$.

Gostota neskončnega reda

Definirajmo gostoto neskončnega reda grafa G na naslednji način:

$$\varrho_\infty(G) =_{def} \lim_{l \to \infty} \varrho_l(G)$$

Gostota neskončnega reda nas pripelje do diskretne funkcije gostote zaradi naslednjega $0 - 1$ izreka.

Izrek

Naj bo $G = (V, E)$ neusmerjen graf.

1. *Drži, da $\varrho_\infty(G)$ je 0 ali 1,*
2. *V je klika, natanko tedaj, ko $\varrho_\infty(G) = 1$.*

Izrek torej pravi, da je edina podgrupa, ki je η-gosta za nek $\eta > 0$ in vse rede, klika.

Preprosto lahko gostoto grafa z n točkami prevedemo v **povprečno stopnjo** točk grafa G na naslednji način:

$$\bar{d}(G) = \varrho(G)(n-1)$$

Ekstremni grafi: področje, ki se je razvilo na temelju Turan-ovega izreka.

Izrek

Dirac, 1963. Naj bo $G = (V, E)$ neusmerjen graf. Če $m > \frac{n^2}{2} \cdot \frac{k-2}{k-1}$, potem G vsebuje podgrafe velikosti $k + r$ s povprečno stopnjo najmanj $k + r - 1 - \frac{r}{k+r}$ za vse $r \in \{0, ..., k-2\}$ in $n \geq k + r$.

Najgostejši podgrafi

Izračun najgostejšega podgrafa glede na povprečne stopnje. Naj bo $\gamma^*(G)$ maksimalna povprečna stopnja nekega nepraznega podgrafa grafa G:

$$\gamma^*(G) =_{def} \max\{\bar{d}(G[U]) \mid U \subseteq V \text{ in } U \neq \{\}\}$$

Maksimalni podgraf z realizacijo $\gamma^*(G)$ je enolično določen.

- Problem: NAJGOSTEJŠI PODGRAF
- Vhodni podatki: Graf G
- Izhodni podatki: Množica točk grafa G, ki realizira $\gamma^*(G)$

Izrek

Obstaja algoritem za reševanje NAJGOSTEJŠEGA PODGRAFA na grafih z n točkami in m povezavami in je rešljiv v času $\mathcal{O}(mn(\log n)(\log \frac{n^2}{m}))$

Algoritem: Oznake

Imamo neusmerjen graf $G = (V, E)$ z n točkami in m povezavami. Dodamo še dve točki s in t in ju povežemo z vsemi točkami V s povezavami kapacitete m. Vse obstoječe povezave grafa G nadomestimo z dvema usmerjenima povezavama kapacitete 1. Dobimo $G' = (V', E')$.
$V' =_{def} \cup\{s, t\}$
$E' =_{def} \{(v, w)|\{v, w \in E\} \cup \{(s, v)|v \in V\} \cup \{(v, t)|v \in V\}$

$$u_\gamma(v, w) =_{def} \begin{cases} 1, & \text{if } \{v, w\} \in E \\ m, & \text{if } v = s \\ m + \gamma - d_G(v), & \text{if } w = t \\ 0, & \text{if } (v, w) \notin E' \end{cases}$$

$c(S, T) = \sum_{v \in S, w \in T} u_\gamma(v, w) = m|V| + |S_+|(\gamma - \bar{d}G|S_+|)$, kjer $S_+ = S - \{s\}$

Algorithm 3: Najgostejši podgraf z minimalnim prerezom in binarnim iskanjem

Input : Graf $G = (V, E)$
Output: Množica k točk grafa G

Določimo $l := 0, r := m$, in $U := \{\}$;
while $r - l \geq \frac{1}{n(n-1)}$ **do**
    $\gamma := \frac{l+r}{2}$
    Konstruitaj pretočno omrežje (V', E', u_γ)
    Poišči minimalni prerez S in T pretočnega omrežja
    if $S = \{s\}$ **then**
        $r := \gamma$
    else
        $l := \gamma$ in $U := S - \{s\}$
Vrni U

Maksimalna povprečna stopnja usmerjenega grafa

Za izboljšanje časovne zahtevnosti lahko uporabimo parametričen algoritem maksimalnega pretoka ($\mathcal{O}(nm \log \frac{n^2}{m})$). Problem NAJGOSTEJŠEGA PODGRAFA pa se da rešiti tudi z linearnim programiranjem. To nam nedvomno poslabša zgornjo mejo časovne zahtevnosti, vendar pa lahko slednjo metodo uporabimo tudi na usmerjenih grafih.

Za vsak usmerjen graf $G = (V, E)$ in ne-prazni množici $S, T \subseteq V$, naj $E(S, T)$ označuje množico povezav, ki gredo iz S v T.

Definiramo povprečno stopnjo para (S, T):

$$\bar{d}_G(S, T) =_{def} \frac{|E(S, T)|}{\sqrt{|S| \cdot |T|}}$$

Maksimalna povprečna stopnja usmerjenega grafa $G = (V, E)$ je definirana kot:

$$\gamma^*(G) =_{def} \max\{\bar{d}_G(S, T) \mid S, T \subseteq V \text{ in } S, T \neq \{\}\}$$

Problem je lahko rešljiv v polinomskem času. To prikazuje naslednja LP poenostavitev LP_γ, kjer γ zavzema vse možne vrednosti $|S|/|T|$:

$$\max \sum_{(u,v)\in E} x_{(u,v)}$$

Pri pogojih:

- $x_{(u,v)} \leq s_u$ za vse $(u, v) \in E$
- $x_{(u,v)} \leq t_u$ za vse $(u, v) \in E$
- $\sum_{u\in V} s_u \leq \sqrt{\gamma}$
- $\sum_{v\in V} t_u \leq \frac{1}{\sqrt{\gamma}}$
- $x_{(u,v)}, s_u, t_v \geq 0$ za vse $u, v \in V$ in $(u, v) \in E$

Če je $\mathcal{O}(n^2)$ mnogo razmerij za $|S|/|T|$ in posledično za γ, lahko izračunamo maksimalno povprečno stopnjo za G (in enako za podgrafe) v polinomskem času z binarnim iskanjem.

Najgostejši podgrafi dane velikosti

Definicija

Za neusmerjen graf $G = G(V, E)$ in parameter $k \in \mathbb{N}$ naj $\gamma^*(G, k)$ označuje **maksimalno vrednost povprečne stopnje** za vse inducirane podgrafe G, ki imajo k točk, tj.

$$\gamma^*(G, k) =_{def} max(\bar{d}(G[U])|U \subseteq Vin|U| = k).$$

Najgostejši podgrafi dane velikosti

Optimizacijski problem:

- Problem: Gost k-podgraf
- Vhodni podatki: Graf G, parameter $k \in \mathbb{N}$
- Izhodni podatki: Množica točk grafa G, ki realizira $\gamma^*(G, k)$

V nasprotju z najgostejšimi grafi je ta problem računsko težak. Jasno je, da je gost k-podgraf $\mathcal{NP}$-težak. Največ, na kar lahko upamo, je algoritem s polinomsko zahtevnostjo z zmernim aproksimacijskim razmerjem. Naravni pristop k aproksimaciji $\gamma^*(G, k)$ temelji na požrešni metodi.

Najgostejši podgrafi dane velikosti

Izrek

Naj bo G graf z n točkami in naj bo $k \in \mathbb{N}, k \leq n$. Naj $A(G,k)$ označuje povprečno stopnjo podgrafov grafa G, ki nastanejo zaradi množice točk, ki so izhodi podatki algoritma požrešna metoda. Potem je

$$\gamma^*(G,k) \leq \frac{2n}{k} \cdot A(G,k).$$

Izrek

Gost k-podgraf je lahko aproksimiran v polinomskem času z razmerjem $\mathcal{O}(n^{\frac{1}{3}-\varepsilon})$, za nek $\varepsilon > 0$.

Parametrizirana gostota

Definicija

Funkcija $\gamma : \mathbb{N} \to \mathbb{Q}_+$ je **začetna gostota** natanko tedaj, ko je γ izračunana v polinomskem času in je $\gamma(k) \leq k - 1$ za vsak $k \in \mathbb{N}$.

Za vse začetne gostote γ, je γ-gost podgraf grafa $G = (V, E)$ vsaka podmnožica $U \subseteq V$, za katero je $\bar{d}(G[U]) \geq \gamma(|U|)$.

Parametrizirana gostota

Optimizacijski problem:

- Problem: γ-gost podgraf
- Vhodni podatki: Graf G, parameter $k \in \mathbb{N}$
- Izhodni podatki: Ali obstaja γ-gost podgraf velikosti k v grafu G?

Če izberemo $\gamma(k) = k - 1$, za vsak $k \in \mathbb{N}$, dobimo γ-gost podgraf, klika, ki je $\mathcal{NP}$-poln problem. Če pa izberemo $\gamma(k) = 0$, vsaka izbira k točk povzroči γ-gost podgraf in ta je rešen v polinomskem času.

Parametrizirana gostota

Izrek

Naj bo γ začetna gostota.

1. *Če $\gamma = 2 + \mathcal{O}(\frac{1}{k})$, potem je γ-gost podgraf rešen v polinomskem času.*
2. *Če $\gamma = 2 + \Omega(\frac{1}{k^{1-\varepsilon}})$, za $\varepsilon > 0$, potem je γ-gost podgraf $\mathcal{NP}$-poln.*

Posledica

Iskanje k-točk podgrafa s povprečno stopnjo vsaj 2, je lahko narejeno v polinomskem času. Tu ni algoritma za iskanje k-točk podgrafa s povprečno stopnjo vsaj $2+\varepsilon$, za vsak $\varepsilon > 0$, razen če $\mathcal{P} = \mathcal{NP}$

Preučili smo računske vidike pojmov lokalnih gostosti, tj. pojmov gostosti, definiranih samo v induciranih podgrafih, ki posledično odpravljajo omrežne strukture izven podgrup. Upoštevali smo strukture (N-plex, N-jedra) in statistične omilitve (η-goste podgrafe) koncepta klika.

Ugnezdenost, kot pomembna struktura znotraj grupe, izključuje hitre algoritme za računanje podgrup določene velikosti. Ta izključitev je podedovana z nekaj omilitvami. Vendar pa nimamo strogih dokazov za ta opazovanja v primeru splošne lokalno določene podgrupe.

LS množica (Luccio-Sami množica)

Definicija

Za graf $G = (V, E)$ je podmnožica točk $U \subseteq V$ **LS množica** natanko tedaj, ko za vse neprazne podmnožice $U' \subset U$ velja

$$|E(U', V - U')| > |E(U, V - U)|.$$

V je LS množica. Tudi množice singeltonov $\{v\}$ so LS množice v G za vsak $v \in V$. Struktura LS množice ima nekaj lepih lastnosti. Na primer, da ne ne-trivialno pokrivajo, tj. če sta U_1 in U_2 LS množici, $U_1 \cap U_2 \neq \emptyset$, potem je ali $U_1 \subseteq U_2$ ali $U_2 \subseteq U_1$. Poleg tega so LS množice precej goste.

Lambda množica

Definicija

Naj bo $G = (V, E)$ neusmerjen graf. Za točki $u, v \in V$ naj $\lambda(u, v)$ označuje število robnih disjunktnih poti med u in v v G, tj. $\lambda(u, v)$ meri robne povezanosti u in v v G. Podmnožica $U \subseteq V$ je **lambda množica** natanko tedaj, ko

$$min_{u,v\in U}\lambda(u, v) > max_{u\in U, v\in V-U}\lambda(u, v).$$

Lambda množica ne meri neposredno gostote podmnožic. Ima pa določen pomen, saj omogočajo algoritem za njihov izračun v polinomskem času. Algoritem je v bistvu sestavljen iz dveh delov, iz izračuna matrike robne povezanosti točk množice V ter iz združevanja točk na level-wise način, tj. točki u in v pripadata isti lambda množici (na ravni N) $\Leftrightarrow \lambda(u, v) \geq N$.

Normalna množica

Normalnost za omrežne podskupine je definirana v statističnem načinu preko slučajnih sprehodov na grafu. Slučajni sprehod je stohastični proces s katerim gremo čez graf, tako da naključno izberemo naslednjo točko, ki jo bomo obiskali, med sosedi sedanje točke.

Definicija

Naj bo $G = (V, E)$ neusmerjen graf. Za $d \in \mathbb{N}$ in $\alpha \in \mathbb{R}_+$ je podmnožica $U \subseteq V$ (d, α)-**normalna** natanko tedaj, ko je za vsako točko $u, v \in U$, za katero je $d_G(u, v) \leq d$, verjetnost, da bo slučajni sprehod, ki je začel v u, prej obiskal v kot pa $w \in V - U$, enaka vsaj α.