SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI SEMINAR

# DETEKCIJA DRUŠTVENIH ZAJEDNICA

Martina Majcen

Zagreb, lipanj 2015.

# Sadržaj

# Uvod

Od kad postoje podaci, postoji i potreba da od njih nastane korisna informacija, a ona nastaje kao rezultat obrade, manipulacije i organiziranja tih podataka. Raznim analizama podataka i primjenama algoritama možemo otkriti zanimljive činjenice koje nisu naizgled poznate sa surovim podacima. U ovom seminaru, naglasak će biti na podacima iz društvenih mreža, odnosno povezanosti samih čvorova u mreži. Postoji mnogo informacija koje se mogu steći analizom podataka iz velikih društvenih mreža. Najpoznatiji primjer društvene mreže je Facebook te „prijateljski“ odnos koji postoji na njemu. Međutim, postoje i druge vrste povezanosti čvorova koje povezuju osobe u društvenoj mreži. U nastavku, bit će prikazane različite tehnike za analizu takvih mreža. Osim analiziranja odnosa između čvorova, odnosno osoba u društvenoj mreži, važno je znati kako prepoznati „zajednice“, tj. podskupove čvorova koji su međusobno povezani više nego s ostalim čvorovima. Čvorovi osim što su međusobno povezani, rijetko ili skoro nikad nisu isključivo u samo jednoj zajednici. Ljudi iz jedne zajednice imaju tendenciju da se međusobno poznaju, ali se ljudi iz dviju različitih zajednica rijetko međusobno poznaju. Osobu se ne treba nužno dodijeliti samo jednoj zajednici, ali nema ni smisla grupirati sve ljude iz svih zajednica u jednu veliku. Često se zajednice preklapaju što dodatno komplicira pronalazak odgovarajućih grupa, te naglašava važnost jednih veza nad drugima. Otkrivanje zajednica je od velike važnosti u sociologiji, biologiji i računalnoj znanosti, disciplinama u kojima se sustavi često prikazuju u obliku grafova. Taj problem je vrlo složen te još uvijek nije riješen na zadovoljavajući način, unatoč golemom trudu velike zajednice znanstvenika koji rade na njemu. Upravo na tome je naglasak u ovom seminaru, te će se pokušati približiti tematika s jednostavnijim i složenijim primjerima obrade tih podataka.

U prvom poglavlju objasnit će se osnovni pojmovi kao što su entiteti i veze u društvenim mrežama, kakvim sve odnosima mogu biti vezani te kakve su im karakteristike općenito. U drugom poglavlju prikazat će se jednostavni graf društvene mreže na kojem će se objasniti osnovni principi udruživanja čvorova te kako i zašto se daje prednost jedne veze nad drugima. Nadalje, u trećem poglavlju čitatelj će se upoznati s nekim od osnovnih metoda za detekciju zajednica, a u posljednjem kroz navedene korake i primjere specifičnijih algoritama detaljnije sagledati još dvije metode.

# 1. Entiteti i veze u društvenim mrežama

Kad pomislimo na društvene mreže, mislimo na Facebook, Twitter, Google+, ili druge web stranice koje se zovu „društvena mreža“, i uistinu je takva vrsta mreža predstavnik šireg opisa mreže koje nazivamo društvenima. Za detaljniju analizu i shvaćanje povezanosti entiteta u mreži, važno je znati na koji sve se sve način entiteti mogu vezati, odnosno u kojim odnosima mogu biti.

## 1.1. Karakteristike društvene mreže

Postoji nekoliko karakteristika društvenih mreža [1] zbog kojih se nazivaju društvenima, a važne su nam za razne analize.

1. Postoji više entiteta koji sudjeluju u mreži. Tipično, ti entiteti su ljudi, ali oni mogu biti i nešto sasvim drugo, što slijedi u poglavlju 1.2.

2. Postoji barem jedan odnos između entiteta mreže koji označava vrstu veze između dva entiteta. Ponekad je odnos isključivo dvosmjeran, dvije osobe ili jesu prijatelji ili nisu, ali može biti i jednosmjeran, jedna osoba može pratiti (engl. *follow*) drugu osobu, što ne znači da se prate međusobno, kao što je slučaj na Twitteru i Instagramu. Međutim, u drugim društvenim mrežama, odnos može imati dodatnu karakteristiku kao što je to npr. prijateljstvo, obitelj ili poznanstvo.

3. Postoji pretpostavka neslučajnosti ili lokalnosti povezivanja entiteta u mrežama, odnosno entiteti imaju tendenciju da se grupiraju. To jest, ako je čvor A, povezan s dva čvora, B i C, veća je vjerojatnost da su B i C također povezani.

## 1.2. Vrste društvenih mreža

Postoje mnogi primjeri društvenih mreža, osim mreža u kojoj su entiteti povezani „prijateljskim“ odnosom. U nastavku slijede još neke društvene mreže za koje se također može računati broj zajednica, odnosno broj grupiranih čvorova [1].

### 1.2.1. Telefonske mreže

U telefonskoj mreži, čvorovi predstavljaju telefonske brojeve. Veza se uspostavlja između dva čvora ako je poziv uspostavljen između tih pokretnih uređaja u nekom vremenskom razdoblju. Veze mogu nositi težine brojeva poziva između pojedinaca tijekom nekog razdoblja. Zajednice u telefonskoj mreži će se formirati od skupine ljudi koji često komuniciraju, na primjer skupina prijatelja, članova kluba ili ljudi koji rade u istoj tvrtki.

### 1.2.2. E-mail mreže

Čvorovi predstavljaju e-mail adrese, a vezu predstavlja činjenica da je najmanje jedan email u najmanje jednom smjeru poslan između dviju adresa. Alternativno, možemo samo staviti prednost ako postoje e-mailovi u oba smjera. Na taj način, možemo izbjeći gledanje *spamera* kao pouzdanu osobu. Drugi pristup je da se veze označe kao slabe ili jake. Jake veze predstavljaju komunikaciju u oba smjera, a slabe veze pokazuju da komunikacija postoji samo u jednom smjeru. Zajednice se grupiraju na isti način spomenut kod telefonskih mreža.

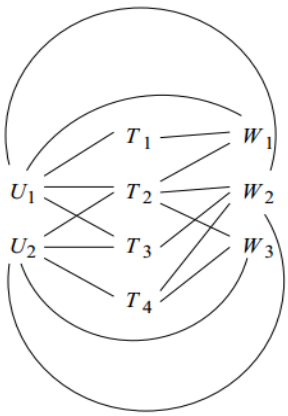
### 1.2.3. Mreža znanstvenika/znanstvenih radova

Čvorovi u ovoj mreži predstavljaju pojedince koji su objavili znanstvene radove. Postoji veza između dvije osobe koji su objavili jedan ili više radova zajedno. Po želji, možemo označiti veze s brojem zajedničkih publikacija. Zajednice u toj mreži su autori koji rade na određenoj temi. Alternativni pogled na iste podatke je graf u kojem su čvorovi radovi. Dva rada povezuje veza, ako imaju najmanje jednog zajedničkog autora. Zajednice se mogu formirati kao zbirke radova na tu temu.

### 1.2.4. Mreže s nekoliko vrsta čvorova

Postoje i druge društvene pojave koje uključuju entitete različitih vrsta. Na primjer, korisnici na stavljaju oznake na nekoj web stranici. Tada postoje tri različite vrste entiteta: korisnici, oznake i stranice. Moglo bi se pomisliti da su korisnici na neki način povezani i ako su skloni da koriste iste oznake često, ili ako su skloni označiti iste stranice. Slično tome, oznake se mogu smatrati povezanima ako se pojavljuju na istim stranicama ili su korištene od istih korisnika, a stranice se smatrati povezanima, ako postoji mnogo istih oznaka ili su mnoge označene od istih korisnika.

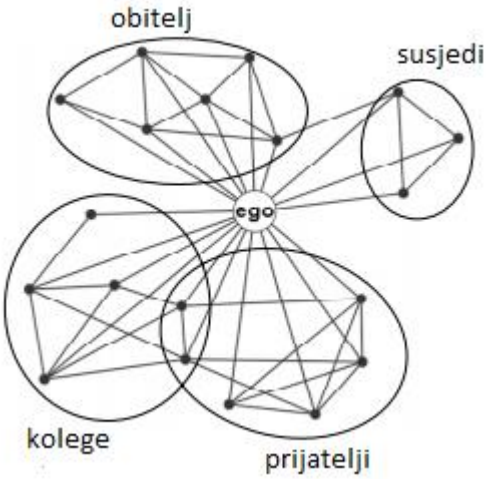
Na Slici 1. je primjer tripartitnog grafa (slučaj $k$ = 3 od k-partitnog grafa). Postoje tri seta čvorova: korisnici {U1, U2}, oznake {T1, T2, T3, T4} i web stranice {W1, W2, W3}. Sve veze povezuju čvorove iz dva različita seta te ovaj graf predstavlja podatke o tri vrste entiteta. Na primjer, veza (U1, T2) govori da je korisnik U1 stavio oznaku T2 na barem jednoj stranici. Međutim, na grafu se ne vidi važan detalj, a to je da se ne vidi tko je postavio koje oznake na kojoj stranici. Prikaz takvih ternarnih veza, zahtijeva složeniji prikaz, kao što je relacija u bazi podataka s tri stupca koji odgovara korisnicima, oznakama i stranicama.




*Slika 1. Tripartitni graf – 3 seta čvorova [1]*

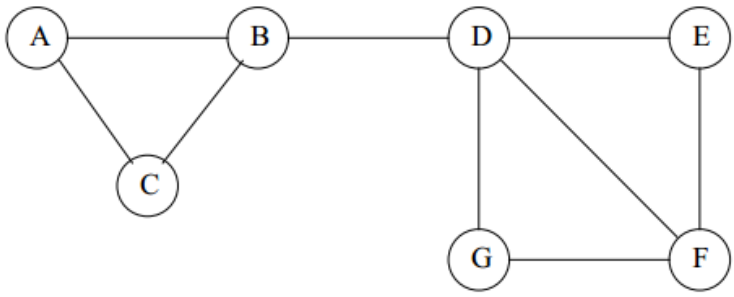
## 2. Primjer zajednice unutar jednostavne društvene mreže

Društvene mreže su prirodno modelirane po uzoru na grafove, koji se mogu nazvati društvenim grafovima. Kao što je već spomenuto, entiteti su čvorovi, a brid spaja dva čvora ako su čvorovi vezani uz odnos koji karakterizira mrežu. Najčešće su društveni grafovi neusmjereni, kao na Facebooku, no oni mogu biti i usmjereni grafovi, kao na primjer na Twitteru, Instagramu ili Google+. Otkrivanje društvenih krugova za određenog korisnika može se formulirati kao problem ego mreže. Ego mreža korisnika je mreža prijateljstva među prijateljima korisnika koji se promatra. Ilustracija ego mreže je prikazana na slici 2. kod koje su u ovom slučaju korisnikovi prijatelji podijeljeni u četiri kruga. Krug kolega s posla ili fakulteta mogu se preklapati s krugom dobrih prijatelja [3].




*Slika 2. Primjer ego mreže [3]*

Na slici 3. [1] je primjer jednostavne društvene mreže. Entiteti su čvorovi od A do G. Veza između čvorova je neusmjerena, te se pretpostavlja da je odnos između njih „prijateljstvo“. Prema slici, B je prijatelj s A, C i D.




*Slika 3. Jednostavni graf društvene mreže [1]*

Može se uočiti da graf ima devet veza, odnosno $\binom{7}{2}$=21 potencijalnih parova čvorova koji bi mogli biti prijatelji. Ako zamislimo da su X, Y, Z neki od čvorova ove mreže i da je X povezan s Y i X sa Z, trebamo izračunati vjerojatnost prijateljstva između Y i Z. Ako bi graf bio velik, vjerojatnost bi bila vrlo blizu slučaju da parovi čvorova imaju vezu između sebe, odnosno, 9/21 = 0.429 u ovom slučaju. Međutim, budući da je graf mali, tu je vidljiva razlika između prave vjerojatnosti i omjera broja veza u broju parova čvorova. Budući da već znamo da postoje veze (X, Y) i (X, Z), postoji još samo sedam veza. Tih sedam veza mogu se povezati između bilo kojih od preostalih 19 parova čvorova. Dakle, vjerojatnost veze (Y, Z) je 7/19 = 0,368.

Shodno tome, treba se izračunati vjerojatnost da veza (Y, Z) postoji u slici, odnosno broj čvorova koji mogu biti Y i Z. S obzirom na bridove (X, Y) i (X, Z), ta veza se nalazi na grafu. Postoji 9 pozitivnih primjera i 7 negativnih primjera te veze, te je omjer između njih 9/16 = 0.563. Može se zaključiti da slika 3. doista ima svojstvo neslučajnosti, odnosno lokalnosti.

# 3. Opći pristupi grupiranju čvorova u društvenoj mreži

Važan aspekt društvenih mreža je da sadrže zajednice osoba koje su povezane brojnim vezama. To obično odgovara skupinama prijatelja u školi ili skupinama znanstvenika zainteresiranih za istu temu. U nastavku će se razmotriti standardne metode grupiranja čvorova u grafu u svrhu identificiranja zajednica.

Neki od općih pristupa grupiranju su: hijerarhijski [1][2], *point-assignment* pristup [1] i particioniranje grafa [1][2]. Da bi se primijenile standardne tehnike grupiranja na društvenom grafu, prvi korak bio bi odrediti „udaljenost“ dvaju čvorova, ovisno o tome postoji li direktna ili posredna veza između dva čvora. Konkretno, koristit će se udaljenost unutar zajednice kao minimalna udaljenost između čvorova dvije zajednice.

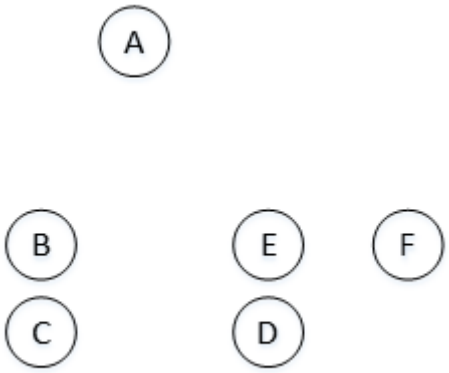
## 3.1 Primjena hijerarhijskog pristupa grupiranju čvorova

Hijerarhijsko grupiranje (engl. *hierarchical-based clustering)* počinje identificiranjem svih čvorova u skupinama, a oni koji su u neposrednoj blizini dvije zajednice, iterativnom metodom naknadno se dodaju. Dakle, hijerarhijsko grupiranje grafa društvene mreže kreće od kombinacije dvaju čvorova koji su povezani.

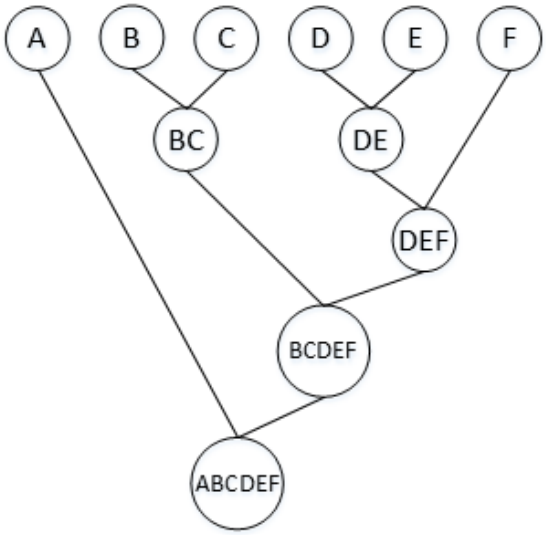
Na prethodnoj slici 3., na prvi pogled vidi se da postoje dvije zajednice {A, B, C} i {D, E, F, G}. Međutim, također se može vidjeti da su {D, E, F} i {D, F, G} dvije podzajednice od {D, E, F, G}. Te dvije podzajednice preklapaju se u dva člana, a tako nikad ne bi mogle biti identificirane od strane algoritama za grupiranje zajednica. Na posljetku, može se razmotriti svaki par pojedinaca koji su povezani kao zajednice veličine 2, ali su takve zajednice nezanimljive.

Problem s hijerarhijskim grupiranjem čvorova nastaje ako se u nekom trenutku odluči kombinirati čvorove koji sigurno spadaju u različite zajednice, kao što su to čvorovi B i D. Razlog zbog kojeg bi se mogli kombinirati B i D je da je D, i svaka zajednica koja ga sadrži, „blizu“ B koliko i svaka zajednica koja ga sadrži, kao što su A i C prema B. Vjerojatnost da prva stvar koja se učini bude kombiniranje B i D u jednu zajednicu je čak 1/9. Postoje stvari koje se mogu učiniti kako bi se smanjila vjerojatnost pogreške. Hijerarhijsko grupiranje pokreće se nekoliko puta te se prikupljaju grupiranja koja nam daju najviše koherentno grupiranje. No, bez obzira na postupak, u velikom grafu s mnogim zajednicama postoji značajna šansa da se u početnim fazama koriste neke veze koje povezuju dva čvora koji ne pripadaju zajedno u bilo kojoj većoj zajednici.

Na slikama 4. i 5. prikazan je općeniti slijed hijerarhijskog grupiranja [4]. Na slici 4. nalaze se čvorovi koji su više ili manje udaljeni jedni od drugih i shodno tome se i hijerarhijski pridjeljuju od sve manjih u sve veće zajednice što prikazuje slika 5.




*Slika 4. Udaljenost čvorova prije hijerarhijskog grupiranja*




*Slika 5. Rezultat hijerarhijskog grupiranja*

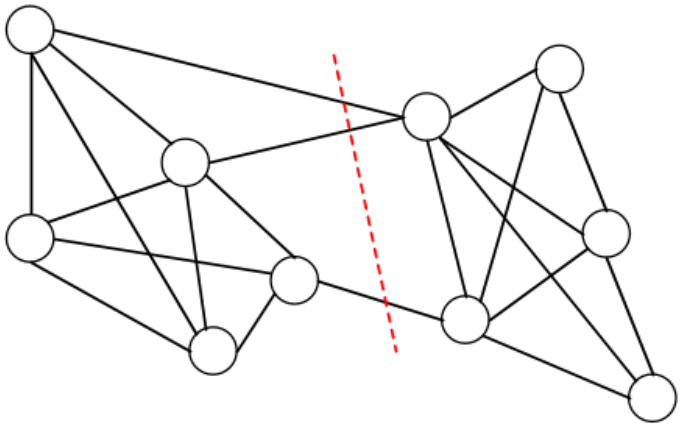
## 3.2 Primjena *point-assignment* pristupa za grupiranje čvorova

*Point-assignment* pristup odjednom uzima u obzir sve čvorove i dodijeljuje ih u zajednicu u kojoj najbolje odgovaraju. Činjenica da su svi bridovi na istoj udaljenosti će dovesti do niza slučajnih čimbenika koji će dovesti do toga da neki čvorovi budu dodijeljeni u krivu zajednicu.

Tradicionalna tehnika ovakvog pristupa je *K-means* grupiranje [5] [6], kojim se graf dijeli na dvije zajednice ako je $k$ = 2. Ako se odaberu dva čvora nasumce, oni mogu biti u istoj zajednici. Nasumično se odabere jedan čvor, a drugi se uzme što je moguće dalje od prvo izabranog čvora. Međutim, pretpostavimo da dobijemo dva odgovarajuća početna čvora, kao što su B i F. Zatim se dodijele A i C čvoru B te isto tako čvorovi E i G čvoru F. D je blizak čvoru B kao što je blizak i čvoru F, tako da bi mogao otići na bilo koju stranu, čak i ako je očito da D spada u F. Ako se odluka o tome gdje smjestiti D odgodi sve dok se ne dodijele neki drugi čvorovi zajednici, onda će se vjerojatno donijeti prava odluka. Na primjer, ako se dodijeli čvor u zajednicu s najkraćom prosječnom udaljenosti prema svim čvorovima zajednice, tada bi D trebao biti dodijeljen zajednici koja sadrži F. Međutim, u velikim grafovima će sigurno biti grešaka u smještanju čvorova u zajednice.

## 3.3. Dijeljenje grafa u particije

Problem particioniranja grafa (engl. *graph partitioning*) sastoji se od dijeljenja vrhova u G skupina unaprijed određene veličine, tako da broj bridova između skupina bude minimalan. Broj bridova koji se nalaze između zajednica naziva se veličina reza. Slika 6. predstavlja rješenje problema za graf s dest vrhova, za G = 2 i zajednice jednake veličine.



*Slika 6. Minimalna veličina reza u grafu s dvije zajednice*

Potrebno je navesti broj zajednica koje se traže minimalnim rezom jer bi u suprotnom mogla postojati zajednica s minimalnom veličinom reza koja bi odgovarala svim čvorovima koji su u istoj zajednici, tako poništavajući veličinu reza.

Također je potrebno odrediti veličinu zajednica, jer bi se u suprotnom, najvjerojatnije rješenje sastojalo od odvajanja čvora najnižeg stupnja od ostatka grafa, što je sasvim nezanimljivo.

Particioniranje grafa je temeljni problem u paralelnom računarstvu te u izradi mnogih serijskih algoritama, uključujući tehnike za rješavanje parcijalnih diferencijalnih jednadžbi i rijetkih linearnih sustava jednadžbi.

# 4. Specijalizirane metode grupiranja čvorova u društvenoj mreži

Budući da postoje problemi sa standardnim metodama grupiranja čvorova, nekoliko specijaliziranih tehnika za grupiranje je razvijeno kako bi se pronašle zajednice u društvenim mrežama. U ovom poglavlju detaljnije će se razmotriti metoda Girvan-Newman i metoda koja se zasniva na pronalasku trokuta u grafu. Također, te dvije metode će se usporediti po određenim kriterijima.

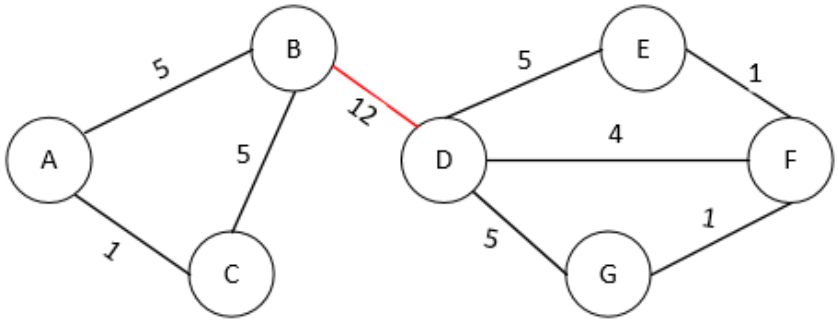
## 4.1 Metoda Girvan-Newman

Jedan od najčešće korištenih metoda za pronalaženje zajednica je Girvan-Newman algoritam [1][2]. Ovaj algoritam identificira bridove u mreži koje leže između zajednica te ih uklanja, ostavljajući iza sebe odvojene zajednice. Identifikacija se obavlja primjenom mjere *betweenness*, koju treba znati izračunati prije samog izračuna zajednica ovom metodom.

### 4.1.1. B*etweenness* brida

*Betweenness* brida (a, b) [5] određuje se na način da se gledaju parovi čvorova X i Y, brid (a, b) koji leže na najkraćem putu između X i Y u odnosu na ukupan broj najkraćih putova u mreži. Može biti nekoliko najkraćih putova između X i Y pa se bridu (a, b) pripisuje udio onih najkraćih putova koji uključuju brid (a, b). Rezultat metode koja koristi ***betweenness*** se ponaša slično kao **mjerenje udaljenosti čvorova** na grafu. To nije baš točna mjera udaljenosti jer nije definirana za parove čvorova koji su nepovezani, a moguće je da ne zadovoljava nejednakost trokuta čak i kad je definirana. Međutim, može se napraviti grupiranje uzimajući bridove s rastućim poretkom vrijednosti *betweennessa* i dodati ih jednom u graf. U svakom koraku, povezane komponente grafa čine neke zajednice. Što se veći *betweenness* razmatra, postoji više bridova i zajednice su veće. Ova metoda predlaže uklanjanje bridova. Prvo se uklone bridovi s najvećim *betweennessom*, dok se graf ne razloži u odgovarajući broj spojenih komponenti, odnosno zajednica.

Na Slici 7. brid (B, D) označen crvenom bojom ima najveći *betweenness*. Zapravo, taj je brid na svakom najkraćem putu između bilo kojeg od čvorova A, B i C prema bilo kojem čvoru D, E, F i G. Njegov *betweenness* je 3*4 = 12. S druge strane, brid (D, F) nalazi se na samo četiri najkraćih putova: one od A, B, C i D prema F**.**




*Slika 7. Graf s označenim betweennessom na bridovima*

### 4.1.2. Algoritam Girvan-Newman

Opisat će se metoda pod nazivom Girvan-Newman (GN), koja posjećuje svaki čvor X jednom i izračunava broj najkraćih puteva od X prema svim drugim čvorovima koji idu kroz svaki od bridova.

**Prvi korak** algoritma započinje s *breadth-first* pretraživanjem (BFS) na grafu, odnosno pretraživanje se vrši „ulaskom“ u svaku razinu od korijena prema listovima i prelazi po svim čvorovima na toj razini. Važno je istaknuti da je razina svakog čvora u BFS prezentaciji duljina najkraćeg puta od X do tog čvora. Dakle, bridovi koji su između čvorova na istoj razini nikada ne mogu biti dio najkraćeg puta od X. Bridovi između razina se zovu DAG (engl. *directed, acyclic graph*) bridovi jer se nalaze na usmjerenom acikličnom grafu. Svaki DAG brid će biti dio najmanje jednog najkraćeg puta od korijena X. Ako postoji DAG brid (Y, Z), u kojoj je Y na nivou iznad Z (tj., bliže korijenu), onda je Y roditelj Z-a, a Z dijete Y-a.

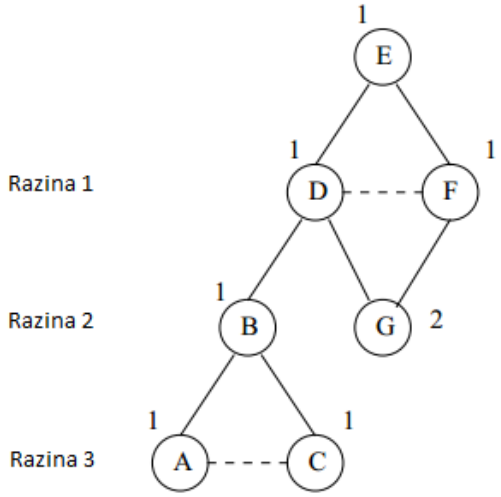
**Drugi korak** GN algoritma je označiti svaki čvor s brojem najkraćih puteva s kojima se može stići iz korijena. Korijen se označi s 1. Zatim, od vrha prema dolje, označi se svaki čvor Y sa zbrojem oznaka svojih roditelja.

**Treći korak** je za svaki brid *e* izračunati zbroj svih čvorova Y kojima se prolazi kroz najkraće putove iz korijena X do Y koji idu putem *e*. Ovaj izračun uključuje računanje zbroja čvorova i bridova od dna prema korijenu. Svaki čvor osim korijena ima oznaku barem 1, što predstavlja najkraći put do tog čvora. Ta vrijednost može se podijeliti između čvorova i bridova koji su iznad jer bi moglo biti nekoliko različitih najkraćih puteva do čvora. Pravila za izračun su:

1. Svaki čvor list u DAG dobiva oznaku 1.
2. Svaki čvor koji nije list dobiva oznaku jednak 1, plus zbroj oznaka od DAG bridova od tog čvora na razini ispod.
3. DAG brid *e* ulazeći u čvor Z s razine iznad je dao dio oznake Z proporcionalno udjelu od najkraćih putova od korijena do Z koji idu putem e. Neka su roditelji čvora Z: Y1, Y2,..., Yk. Neka $p_i$ bude broj najkraćih putova od korijena do Yi; taj broj je izračunat u koraku 2. Zatim oznaka za brid ($Y_i$, Z) je oznaka

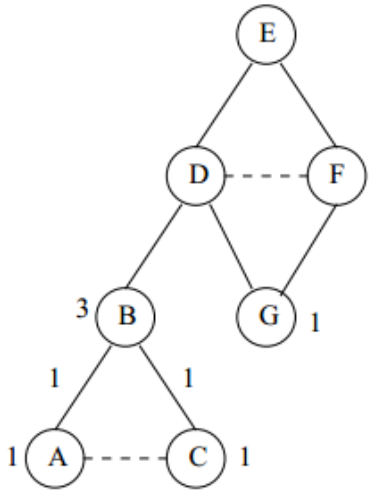
Z puta $p_i$ podijeljena sa sumom svih $p_i$. Nakon provedenog izračuna oznaka sa svakim čvorom kao korijenom, mogu se zbrojiti oznake za svaki brid. Zatim, budući da će svaki najkraći put biti otkriven dva puta (jedanput kada je svaki od krajnjih točaka korijen), moramo podijeliti oznake svakog brida s 2.

Slijedi **primjer** [1] će se pokazati kako zaista izračunati i ilustrirati rezultate pojedinih koraka GN algoritma. Ako postoji graf prikazan na slici 8., može se početi od razine 3 i nastaviti prema korijenu. Prvo se svaki čvor označi sa zbrojem oznaka svojih roditelja. Bridovi označeni punom linijom su DAG bridovi, a iscrtkanom linijom su označeni čvorovi povezani na istoj razini.




*Slika 8. Rezultat drugog koraka GN algoritma*

Zatim je potrebno označiti svaki čvor s brojem najkraćih puteva iz korijena X do Y koji idu putem brida *e*. A i C, kao listovi, dobivaju oznaku 1. Svaka od tih čvorova ima samo jednog roditelja, tako da su njihove oznake dane bridu (B, A) i (B, C), respektivno. Na razini 2, G je list, tako da dobiva oznaku 1. B nije list, tako da se dobiva oznaku 1 plus oznake na DAG bridovima odozdo. Kako oba brida imaju oznake 1, oznaka od B je 3. Intuitivno 3 predstavlja činjenicu da svi najkraći putovi iz E u A, B, C prolaze kroz B. Slika 9. pokazuje oznake dodijeljene do sada.

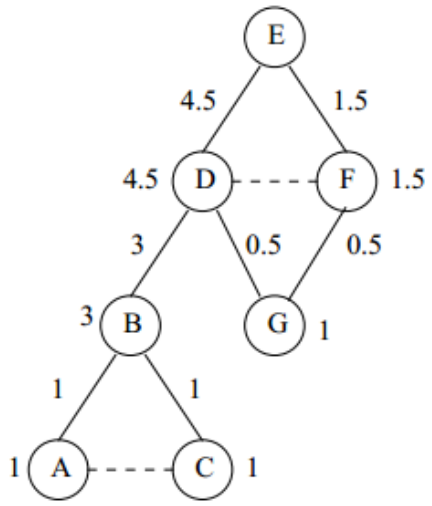



*Slika 9. Međurezultat trećeg koraka GN algoritma*

Nastavljajući prema razini 1, vidi se da B ima samo jednog roditelja, D, tako da brid (D, B) dobiva cijelu oznaku od B, koji je 3. Međutim, G ima dva roditelja, D i F. Stoga je potrebno podijeliti oznaku 1 između čvorova (D, G) i (F, G). Podijelit ćemo ih u omjeru 1:1, jer i D i F imaju oznaku 1, što govori da postoji jedan najkraći put od E do svakog od tih čvorova. Dakle, daje se pola oznake G svakom od tih čvorova; tj. njihove oznake za svakog su 1 / (1 + 1) = 0.5. Kad bi oznake D i F bile 5 i 3, što znači da bi bilo bilo pet najkraćih puteva do D i samo tri do F, onda bi oznaka brida (D, G) bila 5/8, a (F, G) bi bila 3/8.

Na posljetku se dodijele oznake do čvorova na razini 1. (D, E) dobiva oznaku 1 plus oznake bridova koji ulaze od ispod, koji su 3 i 0.5. Dakle, oznaka od D je 4,5. Oznaka od F je 1 plus oznaka od brida (F, G), ili 1,5. Konačno, čvorovi (E, D) i (E, F) dobivaju oznaku od D i F, respektivno, kako svaki od tih čvorova ima samo jednog roditelja. Oznake na svakom od bridova na Slici. 10 su doprinos *betweennessa* tog brida zbog najkraćih puteva iz E. Na primjer, ovaj doprinos za brid (E, D) je 4,5.

Da bi se dovršio račun *betweenness*, treba ponoviti izračun za svaki čvor kao korijen i zbrojiti sve doprinose. Na kraju, kao što je već rečeno, treba se podijeliti s dva da bi se dobio pravi *betweenness* jer svaki najkraći put će biti otkriven dva puta, jednom za svaki od njegovih krajnjih čvorova.




*Slika 10. Rezultat trećeg koraka GN algoritma*

## 4.2. Pronalazak trokuta u grafovima

Jedan od najkorisnijih svojstava društveno-mrežnih grafova je izračun trokuta [1] i drugih jednostavnih podgrafova u mreži. Da je izračun trokuta od velike pomoći prilikom izračunavanja zajednica u društvenoj mreži pokazat će se u nastavku.

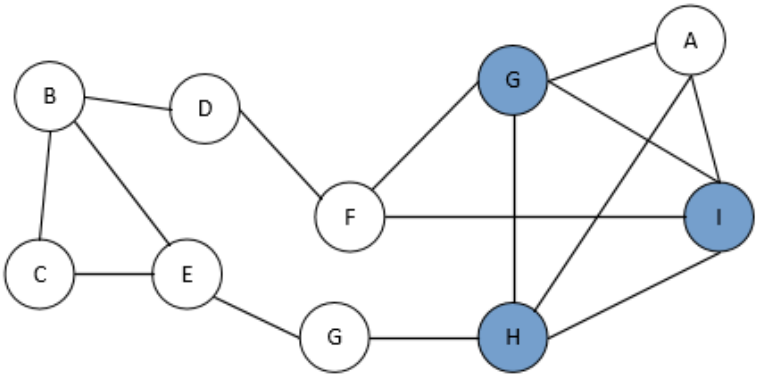
### 4.2.1. Povezanost trokuta u grafu sa zajednicama u društvenim mrežama

Ako imamo graf s $n$ čvorova i dodamo nasumično još $m$ bridova, postoji očekivani broj trokuta u grafu. Taj broj može se izračunati bez previše poteškoća. Gledajući aproksimirane vrijednosti, postoji $n^3/6$ mogućih trokuta, odnosno međusobno povezanih triju čvorova. Vjerojatnost da su bilo koja dva čvora povezana je $2m/n^2$, a vjerojatnost da su bilo koja tri čvora povezana, odnosno da između njih postoji trokut je aproksimirano $(2m/n^2)^3 = 8m^3/n^6$. Konačno, očekivani broj trokuta u grafu s $n$ čvorova i $m$ bridova je $(8m^3/n^6)(n^3/6) = 4/3(m/n)^3$.

Za graf društvene mreže s $n$ sudionika i $m$ parova „prijatelja“ očekivali bismo da je broj trokuta mnogo veći od vrijednosti za slučajni graf. Razlog je u tome što, ako su čvorovi A i B prijatelji, i A je također prijatelj s C, šanse da su B i C su također prijatelji veći su od prosjeka. Dakle, računanje broja trokuta nam pomaže izmjeriti koliko neki graf nalikuje na društvenu mrežu. Kada se zajednica društvene mreže tek formira, ljudi se povezuju sa svojim istomišljenicima, ali je broj trokuta je relativno mali. Ako se A poveže s prijateljem B i C, može se očekivati da B i C ne poznaju. Kako zajednica sazrijeva, B i C mogu komunicirati zbog članstva u zajednici. Dakle, postoji dobra šansa da će se stvoriti trokut {A, B, C} [1].

### 4.2.2. Algoritam pronalaska trokuta u društvenom grafu

Objasnit će se algoritam [1] koji se temelji na pronalasku trokuta u grafu. Pretpostavimo da imamo graf s $n$ čvorova i $m \geq n$ bridova. Radi lakšeg snalaženja, pretpostavimo da su čvorovi cijeli brojevi 1, 2, ..., $n$. Kažemo da je čvor *heavy hitter* ako je njegov stupanj barem $\sqrt{m}$, odnosno ako je to važan čvor u grafu. Na slici 11. prikazan je graf s 10 čvorova u kojem tri čvora koja su označena plavom bojom imaju stupanj = 4, što je dovoljno da se deklariraju kao *heavy hitter* čvorovi, s obzirom da je $3 < \sqrt{10} < 4$. *Heavy hitter* trokut je trokut u kojem su svi čvorovi *heavy hitter*, te takvih nema više od $2\sqrt{m}$ jer bi inače suma stupnjeva takvih čvorova bila veća od $2m$. Budući da svaki brid doprinosi stupnju od samo dva čvora, trebalo bi biti više od $m$ bridova. Koriste se različiti algoritmi za brojanje *heavy hitter* trokuta i ostalih trokuta.




*Slika 11. Graf s heavy hitter i ostalim čvorovima*

Za ulazne parametre čvorova i njihovih povezanosti, koraci algoritma su sljedeći:

1. Izračuna se stupanj svakom čvoru. Ovaj dio zahtijeva da se ispita svaki brid i doda 1 stupnju svakog od dva čvora koji ga čine.

2. Napravi se indeks na bridovima, čiji par čvorova čine ključ. To jest, indeks omogućava odrediti, s obzirom na dva čvora, postoji li brid između njih. Dovoljna je *hash* tablica.

3. Napravi se još jedan indeks bridova, ovaj put s ključem koji pripada jednom čvoru. S obzirom na čvor *v*, možemo dohvatiti čvorove susjedne čvoru *v*. I u ovom slučaju može se koristiti *hash* tablica. Čvorovi se poslože na način da se prvo poredaju čvorovi po stupnju. Zatim, ako *v* i *z* čvorovi imaju isti stupanj, a on može biti samo cijeli broj, poredaju se numerički. To jest, može se reći da je $v \prec u$ ako i samo ako je:

(i) Stupanj *v* je manji od stupnja *u*, ili

(ii) stupnjevi *u* i *v* su isti, i $v < u$.

**Trokuti *heavy hitter*:** Postoji samo $O(\sqrt{m})$ *heavy hitter* čvorova, pa se mogu uzeti u obzir svi skupovi od tri od tih čvorova. Postoje $O(m^3/2)$ mogućih *heavy hitter* trokuta, te se pomoću indeksa na čvorovima može provjeriti postoje li sva tri brida u $O(1)$ vremenskoj složenosti. Stoga, $O(m^3/2)$ je vrijeme potrebno da se pronađu svi takvi trokuti.

**Ostali trokuti** pronalaze se na drugačiji način. Uzimaju se u obzir bilo koji bridovi ($v_1$, $v_2$), osim ako su $v_1$ i $v_2$ su *heavy hitter* čvorovi, koji se zanemaruju. Recimo, međutim, da $v_1$ nije *heavy hitter* i štoviše da je $v_1 \prec v_2$. Neka su $u_1$, $u_2$, ..., $u_k$ čvorovi susjedni čvoru $v_1$. Treba se prisjetiti da je $k < \sqrt{m}$. Ovi čvorovi se mogu naći pomoću indeksa na čvorovima, u $O(k)$ vremenskoj složenosti, odnosno $O(\sqrt{m})$.
Za svaki $u_i$ možemo koristiti prvi indeks kako bi provjerili postoji li brid ($u_i$, $v_2$) u $O(1)$ vremena. Također se može odrediti stupanj $u_i$ u $O(1)$ vremena jer su se prije izbrojili svi stupnjevi čvorova. Uzimamo u obzir trokut $\{v_1, v_2, u_i\}$ ako i samo ako brid ($u_i$, $v_2$) postoji, a $v_1 \prec u_i$. Na taj način, trokut je pronađen samo jednom. Dakle, vrijeme da se obrade svi čvorovi susjedni čvoru $v_1$ je $O(\sqrt{m})$. Budući da postoji m bridova, ukupno vrijeme brojanja *ostalih* trokuta je $O(m^3/2)$ [1].

## 4.3. Usporedba Girvan-Newman metode i metode traženja trokuta

Prethodno opisane metode, Girvan-Newman i pronalazak trokuta, mogu se usporediti s više parametara. Jedan od njih je memorijska složenost algoritma koja je bila bitnija u samim počecima numeričkih izračuna na računalima, kada memorija nije bila tako lako dostupna. Za obje metode, potrebno je obrađivati sve čvorove i bridove i njihovu povezanost, što kod većih mreža rezultira jako velikim matricama i može ograničiti znanstvenike koji nemaju snažna računala s velikom memorijom. No, danas je ipak značajnija analiza vremenske složenosti, koja daje ocjenu o broju osnovnih operacija koje algoritam mora obaviti, prema čemu se onda može odrediti vrijeme potrebno za provođenje

algoritama [6]. Girvan-Newman algoritam vraća rezultate razumne kvalitete te je popularan jer je implementiran u nizu standardnih programskih paketa. Ipak, njegovo izvršavanje teče usporeno, s vremenskom složenosti od $O(m^2n)$ na mreži od $n$ vrhova i $m$ bridova, što nije baš praktično za mreže s više od nekoliko tisuća čvorova. Sličnu vremensku složenost ima i metoda pronalaska trokuta, $O(m^3/2)$ kod koje složenost ne ovisi o broju čvorova već samo o broju bridova. Također za obje metode, nije unaprijed poznato niti moguće tražiti određen broj zajednica na nekom skupu čvorova [6][7]. Prednost oba algoritma je što se nad njima može vršiti paralelizacija, odnosno primjenjiva su za model *Map Reduce* koji se temelji na raspodijeljenoj obradi podataka sa svojstvom linearnog razmjernog rasta. U tablici 1. preglednije su prikazane sličnosti i razlike ove dvije metode. Treba napomenuti i da obje metode imaju i svoje poboljšane verzije u kojima su složenosti izvršavanja manje, te sam rezultat kvalitetniji.

*Tablica 1. Usporedba metoda Girvan-Newman i pronalaska trokuta*

| **Metoda** / **Parametar** | **Girvan-Newman** | **Pronalazak trokuta** |
|---|---|---|
| **vremenska složenost** | $O(m^2n)$ | $O(m^3/2)$ |
| **pogodnost za paralelizaciju** | pogodan | pogodan |
| **ulazni parametri** | matrica povezanosti čvorova s ili bez težine bridova | matrica povezanosti čvorova s ili bez težine bridova |

# Zaključak

Društvene mreže danas pružaju bogat izbor izvora informacija; osim samog sadržaja i pristupačne vrste komunikacije, tu je i širok raspon informacija bez sadržaja na raspolaganju, kao što su veze između ljudi i njihovo djelovanje putem mreža. Te veze između ljudi definiraju zajednice u koje se ljudi međusobno po prirodi grupiraju. Detekcija zajednica značajna je osim u društvenim mrežama i u ostalim sustavima u kojima se entiteti i veze prikazuju u obliku grafa, kao što je primjer u biosustavima, biologiji i općenito računarstvu. Za pronalazak zajednica u mrežama, potrebno je koristiti razne algoritme, koji s većim ili manjim uspjehom računaju broj ljudi u nekoj zajednici ovisno o raznim parametrima. Analizom dvaju algoritama, Girvan-Newman i pronalaskom trokuta u grafu, može se zaključiti da je vremenska složenost algoritma jako bitna ako broj čvorova prelazi nekoliko tisuća, te da je bitna pogodnost za paralelizaciju jer ubrzava cijeli proces.

# Literatura

[1] J. Leskovec, A. Rajaraman, Jeffrey D. Ullman. Mining of Massive Datasets: Mining Social-Network Graphs, 2014.

[2] S. Fortunato. Community detection in graphs: Traditional methods. Physics Reports, 2010

[3] Y. Wang, L. Gao. An Edge-based Clustering Algorithm to Detect Social Circles in Ego Networks. Journal of computers, Vol. 8, No. 10, 2013

[4] Mario Komljenović, Zajednice u mrežama, Seminar, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Zagreb, svibanj 2009

[5] M. E. J. Newman. A measure of betweenness centrality based on random walks. Social networks, 2005

[6] Y. Chen, C. Huang, K. Zhai. Scalable Community Detection Algorithm with MapReduce. Commun. of ACM, 2009

[7] M. E. J. Newman, Fast algorithm for detecting community structure in networks. Physical review E, 2004 - APS

# Sažetak

U radu Detekcija društvenih zajednica objašnjeno je što je društvena mreža, što su entiteti u grafu i kakvim vezama se mogu povezivati. Također, predstavljen je model društvene zajednice u obliku grafa te na jednostavnijim i složenijim primjerima prikazan princip udruživanja čvorova u zajednice te razlog zbog kojeg je potrebno detektirati zajednice unutar mreža. Opisane su standardne metode kao što su hijerarhijski, *point-assignment* pristup i particioniranje grafa. Osim standardnih, opisane su i dvije specifične metode, Girvan-Neman i pronalazak trokuta u grafu. Opisani su koraci algoritma, primjeri izračuna te je dana usporedba po nekim kriterijima.