

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Petra Smole

**Analiza soizbirnosti predmetov na
Fakulteti za računalništvo in
informatiko**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Petra Smole

**Analiza soizbirnosti predmetov na
Fakulteti za računalništvo in
informatiko**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Lovro Šubelj

Ljubljana, 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Sodobna analiza omrežij nudi številne pristope za proučevanje zgradbe velikih kompleksnih omrežij. V diplomskem delu predstavite metode za razvrščanje vozlišč omrežja v skupine ter izbrane metode preizkusite na omrežju soizbirnosti predmetov na Fakulteti za računalništvo in informatiko. Tako dobljene skupine predmetov primerjajte z obstoječimi študijskimi moduli, kritično ovrednotite rezultate ter podajte predloge za nadaljnje delo.

IZJAVA O AVTORSTVU ZAKLJUČNEGA DELA

Spodaj podpisana Petra Smole, vpisna številka 63120241, avtorica pisnega zaključnega dela študija z naslovom:

Analiza soizbirnosti predmetov na Fakulteti za računalništvo in informatiko

IZJAVLJAM

1. da sem pisno zaključno delo študija izdelal samostojno pod mentorstvom doc. dr. Lovra Šublja;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil/-a;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

V Ljubljani, dne 6. septembra 2016

Podpis študenta/-ke:

Iskreno se zahvaljujem svojemu mentorju doc. dr. Lovru Šublju za strokovno vodenje, hitro odzivnost in vso ostalo pomoč pri izdelavi diplomskega dela. Zahvala gre tudi Fakulteti za računalništvo in informatiko, ki mi je omogočila dostop do podatkov, uporabljenih v diplomskem delu.

Posebno zahvalo pa namenjam svoji družini in prijateljem, ki so mi v času študija stali ob strani in me podpirali.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Analiza omrežij	3
2.1	Grafi in omrežja	4
2.2	Utežena omrežja	7
2.3	Dvodelna omrežja	9
2.3.1	Enodelne projekcije dvodelnih omrežij	9
3	Metode za razvrščanje vozlišč	11
3.1	Jedrno-obrobna zgradba	12
3.2	Optimizacija modularnosti	15
3.2.1	Požrešna optimizacija	17
3.2.2	Metoda Louvain	17
3.3	Izmenjava oznak	18
3.4	Bločno modeliranje	21
3.5	Stohastično bločno modeliranje	28
4	Orodja za analizo omrežij	31
4.1	Gephi	32
4.2	Visone	35
4.3	Pajek	37

5	Rezultati in diskusija	41
5.1	Podatki o soizbirnosti predmetov	42
5.2	Priprava podatkov	42
5.3	Rezultati in diskusija	45
5.3.1	1. letnik visokošolskega študijskega programa	45
5.3.2	Vsi študijski programi	46
5.3.3	Magistrski študijski program	49
5.3.4	1. letnik magistrskega študijskega programa	54
5.3.5	2. letnik magistrskega študijskega programa	57
5.3.6	Visokošolski študijski program	61
5.3.7	2. letnik visokošolskega študijskega programa	63
5.3.8	3. letnik visokošolskega študijskega programa	64
5.4	Sklepne ugotovitve	68
6	Sklep in zaključek	69
	Literatura	71
A	Grafi in tabele	75

Seznam uporabljenih kratic

kratica	slovensko
UI	Umetna inteligenca
MM	Multimedija
SO	Strojna oprema
OIV	Omrežja in varnost
PO	Programska oprema
ISSU	Informacijski sistemi in sistemi za upravljanje
ALG	Algoritmika
RM	Računske metode
SPL	Splošni izbirni predmet
STR	Strokovni izbirni predmet
OBV	Obvezni predmet
RAC	Računalništvo
INF	Informatika
VŠŠ	Visokošolski študijski program
MAG	Magistrski študijski program

Povzetek

Analiza omrežij in s tem številne omrežne mere omogočajo vpogled v lastnosti omrežja ter imajo pri razumevanju le-tega pomembno vlogo. V diplomskem delu je podana analiza omrežja visokošolskega in magistrskega študijskega programa Fakultete za računalništvo in informatiko. Namen naloge je poiškati skupine predmetov glede na to, kako jih študentje soizbirajo. To smo dosegli z uporabo petih že uveljavljenih pristopov, namenjenih iskanju skupin v omrežju in s pomočjo prosto dostopnih programskih orodij Gephi, Visone ter Pajek. Za iskanje skupin uporabimo pristop jedrno-obrobne zgradbe, optimizacijo modularnosti, izmenjavo oznak, bločno modeliranje ter stohastično bločno modeliranje. Rezultate, dobljene s pravkar predstavljenimi pristopi, primerjamo in v zadnjem delu podamo interpretacijo tistih, ki so z našega stališča zanimivi ter uporabni. Predmeti posameznega študijskega modula oziroma predmeti izvajalcev istega laboratorija se ne dodelijo le eni od skupin, vendar pa se v nekaterih primerih temu dobro približajo.

Ključne besede: analiza omrežij, skupine vozlišč, skupnosti vozlišč, jedrno-obrobna zgradba.

Abstract

Network analysis and consequently numerous network measurements allow an insight into the characteristics of network and therefore have an important role in its understanding. In this thesis the analysis of the professional and masters study programme of the Faculty of Computer and Information Science is conducted. The purpose is to find groups of courses based on how the students co-enroll them. We have achieved this by using five already established approaches intended for revealing characteristic groups in a network and with the help of widely-accessible software tools such as Gephi, Visone and Pajek. For revealing groups we use the approaches for core-periphery structure, modularity optimization, label propagation, block-modeling and stochastic blockmodeling. The results obtained with the introduced approaches are compared and, in the last part of the thesis, we provide the interpretation of those we find interesting and useful from our point of view. Courses of a particular study module or courses of instructors from the same laboratory are not assigned to just one of the groups, but they do, however, get close in some cases.

Keywords: network analysis, groups of nodes, communities, core-periphery structure.

Poglavje 1

Uvod

Tako z majhnimi kot tudi z velikimi omrežji se vsakodnevno srečujemo na različnih področjih. Le-ta so predstavljena z objekti in povezavami med njimi, ki predstavljajo različne odnose. Dobro poznan primer so spletna družabna omrežja, kot je Facebook, kjer objekte predstavljajo posamezniki, povezave pa ustrezajo prijateljstvom oziroma sodelovanjem med njimi. Zanimiva so predvsem velika omrežja, pridobljena iz različnih virov podatkov, ki so na voljo v računalniško berljivi obliki. Računalnik te podatke prebere in jih shrani v obliko, primerno za kasnejšo uporabo. Govorimo o omrežjih z več deset tisoč, sto tisoč vozlišči in povezavami. Zaradi tega nadaljnja obdelava in analiza brez uporabe računalnika nista mogoči. Na voljo je veliko prosto dostopnih programskih orodij, ki omogočajo vizualizacijo in analizo omrežnih podatkov. Vsako od orodij ima svoj nabor omrežnih mer, ki jih ponuja. Na primer, pri analizi omrežja lahko računamo lastnosti omrežja na ravni vozlišč oziroma povezav ali pa iščemo skupine vozlišč oziroma povezav s skupnimi lastnostmi.

V diplomskem delu analiziramo omrežje visokošolskega in magistrskega študijskega programa Fakultete za računalništvo in informatiko s podatki med leti 2012 in 2015. Omrežje sestavljajo pari identifikator študenta in šifra predmeta. Študent je povezan s predmeti, ki jih je izbral. Vsakemu paru vozlišč v omrežju sta dodana tudi atributa študijski program in letnik.

Tako lahko celotno omrežje ločimo glede na študijska programa in nato še po posameznih letnikih obeh študijskih programov. Dobimo osem različnih omrežij, nad katerimi izvajamo analizo. Namen diplomskega dela je na pravkar omenjenih omrežjih poiskati skupine predmetov glede na to, kako jih študentje soizbirajo. Za iskanje skupin uporabimo že uveljavljene pristope, opisane v Poglavju 3, in prosto dostopna programska orodja, predstavljena v Poglavju 4. Tako bi si študent, ki se vpisuje v enega od letnikov visokošolskega ali magistrskega študijskega programa, lahko pomagal pri izbiri predmetov. Na primer študent je za večino predmetov že odločen, da jih bo izbral, problem pa ima pri izbiri še zadnjih nekaj predmetov. Najprej bi pogledal, v katero skupino so dodeljeni predmeti, ki jih želi imeti. Tako bi dobil seznam ostalih predmetov, dodeljenih isti skupini in s tem predloge za izbiro preostalih predmetov.

V nadaljevanju je diplomsko delo razdeljeno na štiri poglavja. Poglavje 2 predstavi osnovne pojme o omrežjih, ki so potrebni za razumevanje diplomskega dela. V Poglavju 3 je posebna pozornost usmerjena k opisu uporabljenih pristopov, ki so namenjeni iskanju skupin v omrežju. V Poglavju 4 pozornost usmerimo k programskim orodjem, ki ponujajo omenjene pristope ter vizualni prikaz omrežja za nadaljnjo analizo. Poglavje 5 poda rezultate za obravnavana omrežja in diskusijo le-teh.

Poglavje 2

Analiza omrežij

Da bi razumeli načine, s katerimi lahko omrežje vpliva na lastnosti sistema, se moramo seznaniti s teorijo grafov, vejo matematike, ki se ukvarja z omrežji [1]. Njena zgodovina sega več kot 280 let nazaj, začenši z Leonhardom Eulerjem in njegovim prizadevanjem za sprehod, ki bi povezoval sedem mostov v mestu Königsberg [10]. Od takrat se je teorija grafov iz svoje najbolj osnovne oblike, kot medsebojne povezave med objekti, razvila v bogato in izrazito področje. Teorija grafov nam je v pomoč pri reševanju resničnih problemov, med drugim tudi pri različnih vrstah optimizacije. Problemi lahko postanejo bolj preprosti, če jih predstavimo z grafi, in s tem lažje rešljivi. V znanstveni literaturi se izraza graf in omrežje uporabljata izmenično, čeprav se omrežje nanaša na resnične sisteme, izraz graf pa uporabimo takrat, ko govorimo o matematični predstavitvi tega omrežja. Predmet analize so pogosto transportna, socialna, biološka omrežja, internet in podobno.

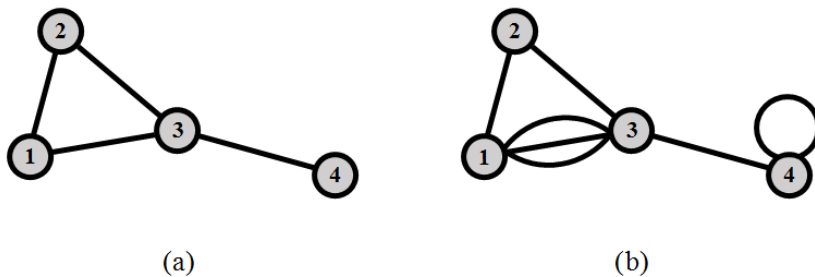
V tem poglavju bomo navedli nekaj osnovnih pojmov iz teorije grafov in omrežij, s katerimi se srečamo v nadaljevanju diplomskega dela.

2.1 Grafi in omrežja

Teorija grafov (angl. graph theory) je matematična in računalniška disciplina, ki proučuje značilnosti grafov ter zagotavlja pomembna orodja za zajem različnih vidikov omrežne strukture [9]. *Omrežje* (angl. network) je predstavitev oziroma matematični model realnega, lahko tudi umetnega kompleksnega sistema.

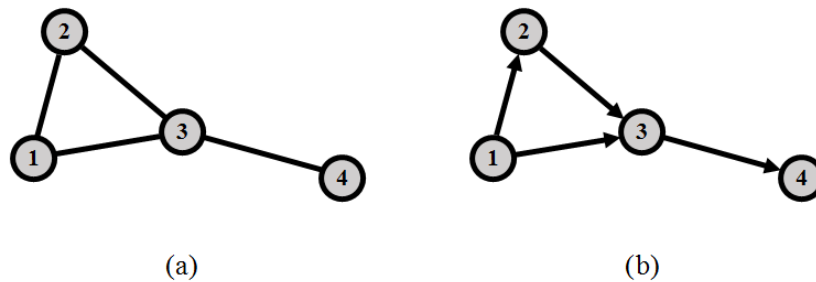
Graf G sestavlja neprazna končna množica elementov, ki jih imenujemo *točke* grafa ali *vozlišča* (angl. vertices), in seznam neurejenih parov teh elementov, ki jih imenujemo *povezave* grafa (angl. edges). Množico točk grafa G pogosto definiramo kot $V(G) = \{1, 2, \dots, n\}$, seznam povezav pa označimo z $E(G) \subseteq \{\{i, j\}; i, j \in V, i \neq j\}$. Število vozlišč ponavadi označimo z n , število povezav pa z m , matematično zapisano kot $n = |V|$ in $m = |E|$. *Omrežje* je definirano z množico *točk* (angl. nodes) in množico *povezav* med njimi (angl. links), ter je predstavljeno z grafom, definiranim zgoraj.

Povezavo, katere začetno in končno krajišče je enako, imenujemo *zanka* (angl. self-loop). Če je med istim parom vozlišč več povezav, le-tim pravimo *vzporedne povezave* (angl. parallel edges). V primeru, da graf ne vsebuje niti zank niti vzporednih povezav, je graf *enostaven* (angl. simple graph). V nasprotnem primeru graf imenujemo *multigraf* (angl. multigraph). Slika 2.1 prikazuje preprost primer enostavnega grafa ter multigrafa.



Slika 2.1: (a) Enostaven graf. (b) Multigraf, ki vsebuje zanko in vzporedne povezave.

Usmerjen graf (angl. directed graph or digraph) je graf, ki ima vse povezave usmerjene. Iz ene točke se premaknemo v drugo le v smeri povezave, nazaj pa po isti poti ne gre, razen, če je povezava dvosmerna. Množica povezav usmerjenega grafa vsebuje urejene pare vozlišč, med katerimi poteka povezava, in jo označimo z $E(G) \subseteq \{(i, j); i, j \in V\}$. Če povezave nimajo usmerjenosti, kar pomeni, da ne vemo, katera točka označuje začetek in katera konec povezave, tedaj je graf *neusmerjen* (angl. undirected graph). V tem primeru množico povezav zapišemo kot $E(G) \subseteq \{\{i, j\}; i, j \in V\}$. Na Sliki 2.2 je prikazan primer enostavnega neusmerjenega grafa, ki je enak grafu s Slike 2.1a, in primer enostavnega usmerjenega grafa.



Slika 2.2: (a) Enostaven neusmerjen graf. (b) Enostaven usmerjen graf.

Ključna lastnost vsakega vozlišča v grafu je njegova *stopnja* (angl. degree). Označimo jo s k_i , kjer je i opazovano vozlišče in predstavlja število povezav z ostalimi vozlišči. Za neusmerjen graf je stopnja i -tega vozlišča definirana kot $k_i = |\{j; \{i, j\} \in E\}|$. V primeru usmerjenega grafa obravnavamo dve vrsti stopenj vozlišča, vhodno in izhodno. Vhodna stopnja danega vozlišča je število povezav, katerih konec je v tem vozlišču. Izhodna stopnja pa je število povezav, katerih začetek je v opazovanem vozlišču. Vhodno stopnjo i -tega vozlišča definiramo kot $k_i^{in} = |\{j; (j, i) \in E\}|$, izhodno pa kot $k_i^{out} = |\{j; (i, j) \in E\}|$.

Omrežja lahko matematično predstavimo na več načinov. V nadaljevanju opišemo tri najpogosteje uporabljene.

Matrika sosednosti (angl. adjacency matrix): je kvadratna matrika A velikosti $n \times n$, kjer je n število vozlišč. Vozlišča pogosto označimo z naravnimi števili $1, 2, \dots, n$. Pri tem ni pomembno, katero oznako ima vozlišče, pomembno je le, da so oznake različne [21]. Če v grafu obstaja povezava med vozliščem j in vozliščem i , je element matrike A_{ij} enak 1, sicer je 0. Matrika sosednosti neusmerjenega grafa je simetrična. Za usmerjene grafe simetričnost v splošnem ne velja, izjemoma v primeru, da so vse povezave dvosmerne. Slika 2.3 prikazuje matriko sosednosti za enostaven neusmerjen graf, za enostaven usmerjen graf ter za multigraf. Vidimo, da sta leva in desna matrika simetrični, srednja pa ne, saj pripada usmerjenemu grafu. Če pogledamo matriko sosednosti s Slike 2.3c, vidimo, da njeni elementi niso več samo iz množice $\{0, 1\}$, ampak ustrezajo številu povezav med vozlišči.

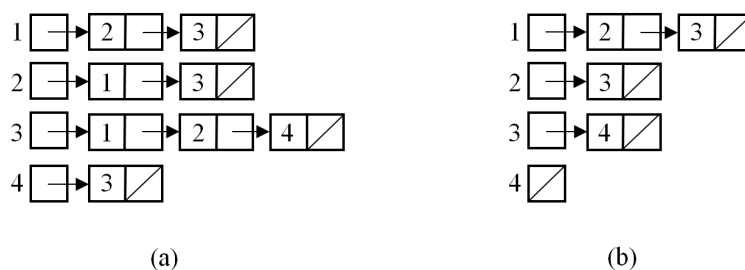
$$\begin{array}{ccc}
 \mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} & \mathbf{A} = \begin{pmatrix} 0 & 1 & 3 & 0 \\ 1 & 0 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)}
 \end{array}$$

Slika 2.3: (a) Matrika sosednosti za enostaven neusmerjen graf s Slike 2.2a. (b) Matrika sosednosti za enostaven usmerjen graf s Slike 2.2b. (c) Matrika sosednosti za multigraf s Slike 2.1b.

Seznam povezav (angl. edge list): vsebuje pare vozlišč, med katerimi poteka povezava v grafu. Pri usmerjenem grafu je potrebno paziti na vrstni red vozlišč v paru. Leva komponenta je vedno začetno vozlišče, desna pa končno. Vrstni red pri neusmerjenem grafu ni pomemben. Celotno omrežje lahko v tem primeru predstavimo tako, da podamo število vozlišč n in seznam povezav. Za graf s Slike 2.2b, kjer je $n = 4$, seznam

povezav zapišemo tako: $(1,2)$, $(1,3)$, $(2,3)$, $(3,4)$. Enak seznam povezav ustreza tudi neusmerjenemu grafu s Slike 2.2a, saj zaporedje elementov v paru tukaj ni ključnega pomena.

Seznam sosednosti (angl. adjacency list): za vsako vozlišče hrani seznam vseh tistih vozlišč, s katerimi je povezano. Na Sliki 2.4 je prikazan seznam sosednosti enostavnega neusmerjenega grafa in seznam sosednosti enostavnega usmerjenega grafa.



Slika 2.4: (a) Seznam sosednosti za enostaven neusmerjen graf s Slike 2.2a. (b) Seznam sosednosti za enostaven usmerjen graf s Slike 2.2b.

Vsaka od predstavitev omrežij je primerna za določene namene. Matriko sosednosti uporabljamo predvsem za elegantne analitične izpeljave. Seznam povezav za učinkovito shranjevanje in manipulacijo omrežij, saj je vsaka povezava shranjena samo enkrat. Seznam sosednosti pa se pogosto uporablja za učinkovite implementacije algoritmov.

V primeru, da povezavi priredimo število, govorimo o uteženih omrežjih, ki jih predstavimo v Poglavju 2.2.

2.2 Utežena omrežja

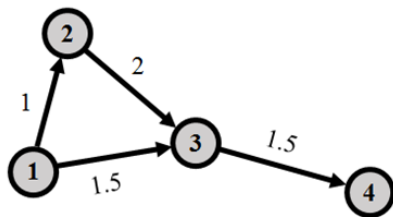
Uteženo omrežje (angl. weighted network) predstavimo z uteženim grafom, kjer je vsaki povezavi prirejeno edinstveno število w_{ij} , imenovano utež. Tudi med uteženimi grafi ločimo usmerjene in neusmerjene, kjer je v obeh primerih

množica vozlišč enaka $V(G) = \{1, 2, \dots, n\}$. Za neusmerjen utežen graf je množica povezav definirana kot $E(G) \subseteq \{\{i, j\}, w_{ij}; i, j \in V, w_{ij} \in \mathbb{R}_{>0}\}$. Tedaj za matriko sosednosti A velja, $A_{ij} = w_{ij}$, če in samo če obstaja povezava med vozliščema i in j . Pri usmerjenem uteženem grafu množico povezav zapišemo z $E(G) \subseteq \{((i, j), w_{ij}); i, j \in V, w_{ij} \in \mathbb{R}_{>0}\}$. Sedaj za matriko sosednosti A velja, $A_{ij} = w_{ij}$ natanko tedaj, ko se povezava začne v vozlišču i in konča v vozlišču j .

Kakšna je matrika sosednosti v primeru uteženega grafa, smo ravnokar predstavili, sedaj pa povejmo še, kako se v tem primeru spremenita drugi dve predstavitvi omrežij. Pri seznamu povezav parom vozlišč kot tretjo komponento dodamo utež. Vsakemu vozlišču seznama sosednosti pa dodamo vrednost njegove povezave.

Število w_{ij} ima več pomenov. V geografskem smislu je to lahko geografska razdalja. V primeru transportnega omrežja lahko predstavlja potreben čas, da se pride iz enega kraja v drugega, ali pa strošek poti med dvema krajema. Pri internetnem omrežju je to lahko količina podatkov, ki tečejo vzdolž povezav, v socialnem omrežju pa pogostost stikov med posamezniki [21].

Slika 2.5 prikazuje primer enostavnega uteženega grafa in pripadajočo matriko sosednosti.



(a)

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1.5 & 2 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \end{pmatrix}$$

(b)

Slika 2.5: (a) Utežen graf. (b) Matrika sosednosti za graf s Slike 2.5a.

2.3 Dvodelna omrežja

Dvodelno omrežje (angl. two-mode network) predstavimo z bipartitnim grafom, katerega vozlišča lahko razdelimo v dve disjunktni množici X in Y tako, da ima vsaka povezava eno krajišče v množici vozlišč X in drugo v Y . Iz tega sledi, da v dvodelnem grafu ne obstaja povezava, ki bi povezovala dve vozlišči iste množice. Za množico vseh vozlišč velja $V(G) = X \cup Y$ in $X \cap Y = \emptyset$. Množico povezav označimo z $E(G) \subseteq \{\{x, y\}; x \in X \wedge y \in Y\}$.

Mnogo sistemov je že naravno modeliranih kot dvodelno omrežje. Dobro poznan primer je omrežje hollywoodskih igralcev, v katerem ena množica vozlišč ustreza igralcem (X) in druga filmom (Y) [1]. Film je povezan z igralcem, če se le-ta pojavi v njem.

2.3.1 Enodelne projekcije dvodelnih omrežij

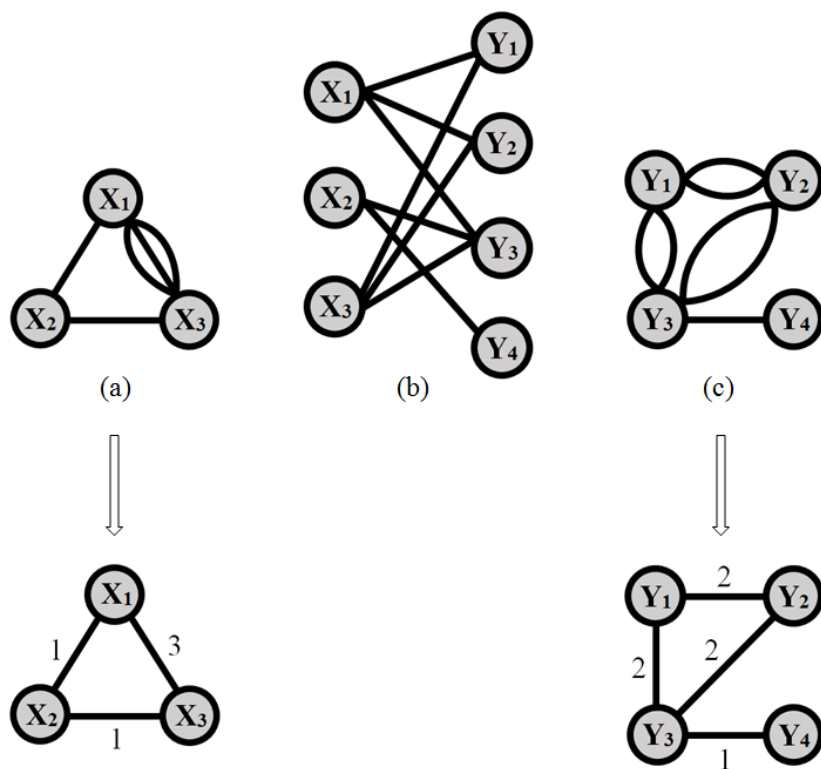
V zadnjem času je veliko pozornosti posvečene analizi in modeliranju dvodelnega omrežja. Zaradi razlogov enostavnosti dvodelno omrežje običajno projiciramo na enodelno omrežje [26]. Za vsako dvodelno omrežje lahko generiramo dve enodelni projekciji, kjer je rezultat posamezne projekcije pogosto multigraf. Prva projekcija poveže dve vozlišči množice X s povezavo, če sta v dvodelni predstavitvi povezani z istim vozliščem množice Y . Druga projekcija poveže dve Y vozlišči, ko imata vsaj eno skupno sosednje vozlišče v množici X .

Prva projekcija zgoraj že omenjenega omrežja hollywoodskih igralcev je omrežje igralcev, v katerem sta dve vozlišči povezani med sabo, če se igralca pojavita v istem filmu [1]. Druga projekcija je omrežje filmov, kjer sta dva filma povezana, če si delita najmanj enega igralca v zasedbi.

Najenostavnejši način projekcije je projiciranje dvodelnega omrežja na neuteženo, brez upoštevanja podatka, kolikokrat je bilo sodelovanje med pari vozlišč ponovljeno [26]. Kljub temu, da so nekatere topološke lastnosti kakovostno pridobljene iz te neutežene verzije, je enodelna projekcija vedno manj informativne narave kot originalna dvodelna predstavitev. Da se bolje

ohranijo prvotne informacije, je potrebna uporaba dvodelnega grafa in s tem ustrezna določitev uteži povezavam v projekcijskem grafu. Tedaj multigraf prevedemo na enostaven utežen graf, kjer utež pomeni število ponovljenih povezav med vozliščema.

Slika 2.6b prikazuje primer dvodelnega omrežja, Sliki 2.6a in 2.6c pa njegovi projekciji na enodelno omrežje. Pod vsako projekcijo se nahaja prevedba multigrafa na enostaven utežen graf.



Slika 2.6: (a) Enodelna projekcija na X in pripadajoč enostaven utežen graf. (b) Dvodelno omrežje. (c) Enodelna projekcija na Y in pripadajoč enostaven utežen graf.

Poglavje 3

Metode za razvrščanje vozlišč

Razložimo nekaj pojmov, ki jih uporabimo v tem poglavju. Veliko je govora o *razvrščanju* ali *gručenju* (angl. clustering) v *skupine* ali *gruče* (angl. clusters). Gre za združevanje elementov na tak način, da so si elementi iste skupine bolj podobni med sabo, kot pa z elementi drugih skupin. *Skupnosti* (angl. communities) po drugi strani pa so posebne vrste skupin vozlišč v realnih omrežjih. Zanje velja, da so vozlišča znotraj skupnosti tesno povezana med sabo in redko z vozlišči ostalih skupnosti.

Particija ali *razbitje* (angl. partition) omrežja pomeni, da vozlišča omrežja razdelimo v več nepraznih skupin in velja, da je vsako vozlišče del le ene skupine. Z drugimi besedami, unija vseh skupin je množica vozlišč, presek med dvema skupinama pa je prazen.

V tem poglavju predstavimo pet pristopov za razvrščanje vozlišč, ki jih uporabimo pri izdelavi diplomskega dela. Začnemo s predstavitvijo jedrno-obrobne zgradbe [8], nadaljujemo z optimizacijo modularnosti [1], izmenjavo oznak [22], bločnim modeliranjem [12] in končamo z opisom stohastičnega bločnega modeliranja [13]. Jedrno-obrobna zgradba je namenjena iskanju razbitja na dva razreda – jedro in obrobje. Optimizacija modularnosti, izmenjava oznak in stohastično bločno modeliranje iščejo delitev omrežja na skupnosti, bločno modeliranje pa odkriva skupine v omrežju.

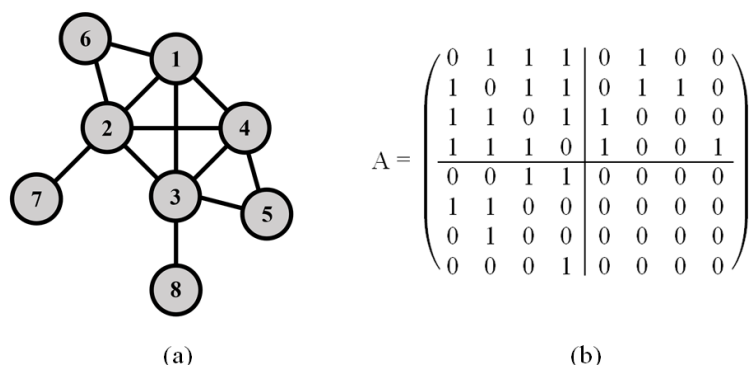
3.1 Jedrno-obrobna zgradba

Pogost, vendar neuraden pojem v analizi socialnih omrežij in na drugih področjih, je koncept *jedrno-obrobne zgradbe* (angl. core-periphery structure), ki ga je prvi razvil Krugman leta 1991 [8]. Z izračunom jedrno-obrobne zgradbe omrežja je potrebno ugotoviti, katera vozlišča so del gosto povezanega jedra in katera so del redko povezanega obrobja [23]. Jedro je tudi razmeroma dobro povezano z obrobniimi vozlišči, medtem ko obrobna vozlišča med sabo niso povezana. Kljub temu, da je pojem jedrno-obrobne zgradbe že precej razširjen, ni bil nikoli formalno opredeljen. Pomanjkanje definicij pomeni, da lahko avtorji izraz uporabljajo na različne načine, zaradi česar je težko primerjati sicer primerljive študije.

Jedrno-obrobno zgradbo lahko predstavimo kot pojem *dvorazrednih particij vozlišč* (angl. two-class partition of nodes), kjer en razred predstavlja jedro in drugi obrobje [8]. Za elemente matrike sosednosti A , kjer i in j pripadata jedru, velja, da je $A_{ij} = 1$. Če i in j pripadata obrobju, je $A_{ij} = 0$. Ostali elementi matrike A predstavljajo povezave med jedrom in obrobjem ter imajo lahko vrednost 1 ali 0. Ta vzorec predstavlja značilnost jedrno-obrobni zgradb.

Na Sliki 3.1a je podan graf, ki prikazuje primer jedrno-obrobne zgradbe. Poleg grafa je pripadajoča matrika sosednosti, kjer vzporednica in navpičnica ločujeta jedro od obrobja. Vozlišča $\{1, 2, 3, 4\}$ so del jedra, kar na matriki sosednosti predstavlja zgornji levi del, kjer so vsi elementi, razen diagonalnih, enaki 1. Obrobje predstavljajo elementi $\{5, 6, 7, 8\}$, kar je na matriki vidno kot ničelni predel spodaj desno.

Po tem, kar smo ravnokar povedali, je videti, kot da omrežje ne more imeti več kot eno jedro. Kljub temu pa drugi načini razmišljanja o jedrno-obrobni zgradbi vodijo k možnosti, da imamo več jeder, vsako s svojim lastnim obrobjem [24]. V vsakem primeru omejitev na eno samo jedro ni tako stroga kot je videti na prvi pogled, saj se lahko vedno odločimo analizirati podgraf omrežja, kjer menimo, da je le eno jedro [8].



Slika 3.1: (a) Graf, ki prikazuje primer jedrno-obrobne zgradbe. (b) Matrika sosednosti za graf s Slike 3.1a.

Borgatti in Everett sta predlagala algoritme za odkrivanje diskretne ter zvezne različice jedrno-obrobne zgradbe v uteženih neusmerjenih grafih [23]. Ločita se glede na število jeder in ju predstavimo v nadaljevanju.

Diskretni model (angl. discrete model): v tem primeru je jedrno-obrobni model sestavljen iz dveh razredov vozlišč [8]. Iz povezanega podgrafa – jedra, v katerem so udeleženci povezani med seboj v maksimalnem smislu in razreda udeležencev – obrobja, kjer so vozlišča razmeroma dobro povezana z jedrom, med sabo pa ne.

Matrika sosednosti na Sliki 3.2 predstavlja tako imenovan idealen vzorec, vendar ne obstaja veliko možnosti, da bi ga opazili v empiričnih podatkih. Zlahka lahko upoštevamo, da se realne strukture le približajo temu vzorcu. To pomeni, da ima del matrike sosednosti, ki predstavlja povezave med jedrom v večini vrednosti 1, vendar se pojavijo tudi 0. Podobno se v delu matrike A , ki predstavlja povezave med obrobjem, pojavijo tudi vrednosti 1, poleg prevladujočih 0.

Preprosta mera, ki pove, kako dobro realna struktura aproksimira idealno, je podana z enačbo (3.1). A_{ij} kaže na prisotnost oz. neprisotnost povezave med vozliščema j in i , medtem ko δ_{ij} označuje prisotnost oziroma neprisotnost povezave v idealnem vzorcu. Elementi δ_{ij} sestavljajo

$$\Delta = \left(\begin{array}{cccc|cccc} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array} \right)$$

Slika 3.2: Matrika sosednosti, ki predstavlja idealen vzorec.

matriko Δ , ki jo imenujemo tudi *vzorčna matrika* (angl. pattern matrix). Mera doseže svojo najvišjo vrednost, če in samo če sta matriki A in Δ enaki, kar se zgodi, ko ima A popolno jedrno-obrobno zgradbo. Velja, da večji kot je ρ , bolj se naši podatki približajo idealnemu vzorcu.

$$\rho = \sum_{i,j} A_{ij} \delta_{ij} \quad (3.1)$$

$$\delta_{ij} = \begin{cases} 1, & \text{če je } i \text{ ali } j \text{ v jedru} \\ 0, & \text{sicer} \end{cases} \quad (3.2)$$

Zvezni model (angl. continuous model): ena od omejitev zgoraj omenjenega pristopa je pretirana preprostost definiranja le dveh razredov vozlišč – jedra in obrobja [8]. Da bi to odpravili, lahko uvedemo večrazredno particijo. Ta pristop je izvedljiv, vendar je navajanje idealnega vzorca, ki najbolje zajame pojem jedrno-obrobne zgradbe več razredov relativno težko. Zahtevnost problema se povečuje eksponentno, z večanjem števila razredov.

Alternativni pristop je opustitev diskretnega modela v prid zveznemu, kjer se vsakemu vozlišču dodeli mera, imenovana *jedrna središčnost* (angl. coreness). Središčnost predstavlja funkcijo bližine vozlišča do jedra. Še vedno je dober pristop, da za oceno prileganja uporabljamo korelacijo med realnim in idealnim modelom. Vendar elemente matrike Δ spremenimo, kot prikazuje enačba (3.3), kjer c_i predstavlja stopnjo

središčnosti vozlišča i . Vzorčna matrika Δ ima velike vrednosti za pare vozlišč, kjer imata obe vozlišči visoko središčnost, srednje vrednosti za pare, kjer ima eno vozlišče visoko središčnost in drugo nizko, ter nizke vrednosti za pare vozlišč, kjer sta obe vozlišči obrobni.

$$\delta_{ij} = c_i c_j \quad (3.3)$$

Razložimo še pomen k -jedra (angl. k -core). k -jedro je največji podgraf z minimalno stopnjo k [18]. To pomeni, če k -jedru dodamo vozlišče, potem nima več minimalne stopnje k .

Identifikacija jedrno-obrobne zgradbe ima številne možnosti uporabe [25]. Jedrna vozlišča v omrežju imajo lahko različno vlogo od obrobni in sposobnost razlikovanja jedra od obrobja nam lahko poda nov pristop pri delovanju omrežnih sistemov. Razlikovanje med jedrom in obrobjem lahko privede do bolj informativne ter hitreje razumljive vizualizacije omrežij.

3.2 Optimizacija modularnosti

Priljubljena in učinkovita metoda za odkrivanje skupnosti v omrežjih je *optimizacija modularnosti* (angl. modularity optimization) [15]. Temelji na optimizaciji količine, ki sta jo prva predstavila Girvan ter Newman in se imenuje *modularnost* (angl. modularity). Le-ta meri kakovost delitve omrežja na skupnosti in se označi s Q .

Predstavljajmo si omrežje z n vozlišči in m povezavami ter razbitje na K skupnosti [1]. Skupnost C ima n_c vozlišč, med seboj povezanih z m_c povezavami, kjer je $c = 1, \dots, K$. Če je m_c večji od pričakovanega števila povezav med n_c vozlišči glede na zaporedje stopenj omrežja, tedaj podgraf, inducirani na teh vozliščih, predstavlja skupnost.

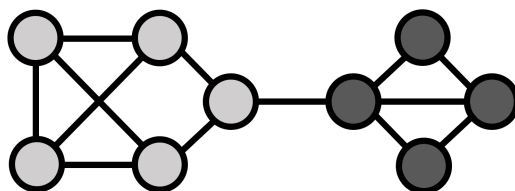
Modularnost razbitja definiramo kot vsoto po vseh K skupnostih. Za vsako skupnost izračunamo razliko med številom povezav znotraj te skupnosti in pričakovanim številom povezav v naključno povezanem grafu enake velikosti ter enakega zaporedja stopenj. Razlika ustreza enačbi (3.4), kjer

m_c predstavlja skupno število povezav znotraj skupnosti C , k_c pa je vsota stopenj vseh vozlišč v tej skupnosti.

$$Q = \sum_{c=1}^K \left[\frac{m_c}{m} - \left(\frac{k_c}{2m} \right)^2 \right] \quad (3.4)$$

Optimizacija modularnosti nam pove, katero razbitje ustreza optimalni strukturi skupnosti. Izberemo tisto razbitje, ki ima največjo vrednost Q , $Q \leq 1$. Velja torej, da višja kot je modularnost razbitja, boljša je ustrezna struktura skupnosti. Lahko se zgodi, da je vrednost Q negativna ali enaka 0. Če pri grafu na Sliki 3.3 vsako vozlišče predstavlja svojo skupnost, je $Q < 0$; v primeru, da cel graf določa eno skupnost, je $Q = 0$. Za graf na Sliki 3.3, kjer vozlišča enake barve določajo skupnost, izračunajmo modularnost:

$$Q = \left[\frac{7}{13} - \left(\frac{15}{2 \cdot 13} \right)^2 \right] + \left[\frac{5}{13} - \left(\frac{11}{2 \cdot 13} \right)^2 \right] \doteq 0.41$$



Slika 3.3: Optimalno razbitje grafa na dve skupnosti.

Načeloma bi lahko najboljše razbitje prepoznali s preverjanjem Q za vsa možna razbitja in izbrali tisto z največjo modularnostjo Q . Glede na to, da se število možnih razbitij povečuje hitreje kot eksponentno z velikostjo omrežja, je izčrpna optimizacija računsko neizvedljiva že pri razmeroma majhnih omrežjih [15]. Zato vsi algoritmi, ki optimizirajo modularnost, temeljijo na približni optimizaciji [19]. V nadaljevanju predstavimo dva pristopa, ki najdeta razbitje s skoraj maksimalno modularnostjo Q , brez da bi bilo potrebno pregledati vsa možna razbitja.

3.2.1 Požrešna optimizacija

Prvi pristop za optimizacijo modularnosti, ki ga predlagal Newman, iterativno združuje pare skupnosti, če poteza poveča modularnost razbitja [1]. Gre za *požrešno optimizacijo* (angl. greedy optimization) in algoritem, ki temelji na tej metodi, sledi naslednjim korakom:

1. vsako vozlišče določimo svoji skupnosti. Začnemo torej z n skupnostmi posameznih vozlišč;
2. za vsak par skupnosti, povezan z vsaj eno povezavo, izračunamo dobljeno razliko modularnosti ΔQ v primeru, da skupnosti združimo. Par, za katerega je razlika modularnosti ΔQ največja, združimo v eno skupnost. Upoštevamo, da se modularnost vedno računa za celotno omrežje;
3. drugi korak ponavljamo, dokler niso vsa vozlišča združena v eno skupnost in si sproti zapomnimo Q za vsak korak;
4. izberemo razbitje, katerega modularnost Q je največja.

3.2.2 Metoda Louvain

Pred kratkim je pristop, ki ga najdemo pod imenom *metoda Louvain* (angl. Louvain method), pridobil na popularnosti, saj je njegova točnost primerljiva z natančnostjo drugih pristopov za razvrščanje vozlišč, predvsem pa je izjemno hiter. Zato ga lahko uporabimo za identifikacijo skupnosti v zelo velikih omrežjih. Sestavljen je iz dveh korakov, ki se ponavljata iterativno, in sicer:

1. začnemo z uteženim omrežjem z n vozlišči in vsako vozlišče dodelimo različni skupnosti. Za vsako vozlišče i ocenjujemo razliko modularnosti ΔQ , če damo vozlišče i v skupnost enega od svojih sosedov j . Vozlišče i premaknemo v skupnost, za katero je razlika modularnosti največja, vendar le, če je ta razlika pozitivna. V nasprotnem primeru i ostane v

svoji prvotni skupnosti. Ta korak se izvede za vsa vozlišča, dokler ni več mogoče doseči izboljšanja;

2. zgradimo novo omrežje, katerega vozlišča so skupnosti, ugotovljene v koraku 1. Teža povezave med dvema vozliščema je vsota uteži povezav med vozlišči v ustrezni skupnosti. Povezave med vozlišči iste skupnosti vodijo do uteženih zank.

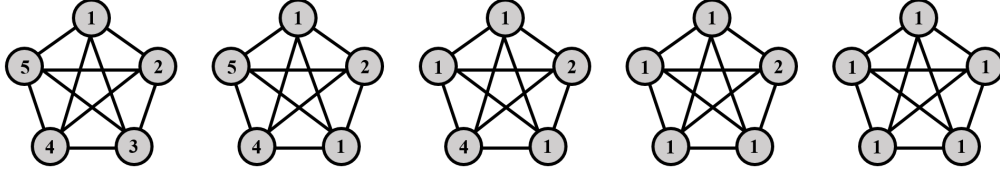
Oba koraka ponavljamo toliko časa, dokler ne obstaja več sprememb in je dosežena največja modularnost.

3.3 Izmenjava oznak

Algoritem izmenjave oznak (angl. label propagation algorithm), ki so ga predlagali Raghavan, Albert ter Kumara, je preprost, učinkovit in hiter algoritem, ki se uporablja za odkrivanje in analiziranje omrežnih skupnosti [22]. Pri izvajanju uporablja samo strukturo omrežja za usmerjanje njegovega napredka in ne zahteva nobene v naprej definirane funkcije, ki bi jo bilo potrebno optimizirati. Glavna prednost je ta, da ne potrebuje vnaprejšnjih informacij o številu skupnosti in o njihovi velikosti. Najdene skupnosti so visoke kakovosti, merimo jih lahko z že poznano mero – z modularnostjo [2].

Na začetku algoritma vsakemu vozlišču v omrežju dodelimo edinstveno oznako, ki označuje, kateri skupnosti vozlišče pripada [19, 22]. Torej imamo n med sabo različnih oznak. Na vsakem koraku vsa vozlišča posodobijo svojo oznako v najpogostejšo oznako v svoji soseščini. Kadar je več oznak enako pogostih, je nova oznaka izbrana naključno, lahko pa se obdrži trenutna oznaka, če je le-ta med najpogostejšimi.

Vrstni red, v katerem so vsa n vozlišča v omrežju posodobljena na vsakem koraku, je izbran naključno. Slika 3.4 prikazuje algoritem izmenjave oznak na primeru polnega grafa na petih vozliščih. Začetni graf je skrajno levi z različnimi oznakami, končni pa je skrajno desni, kjer vidimo, da vsa vozlišča dobijo enako oznako.



Slika 3.4: Preprost primer algoritma izmenjave oznak, kjer se oznake vozlišč posodablja od leve proti desni.

Postopek posodobitve je lahko bodisi sinhron bodisi asinhron [22]. Pri *sinhronem* posodabljanju vozlišče x v t -ti ponovitvi posodobi svojo oznako na novo, ki temelji na oznakah svojih sosedov v $(t - 1)$ -i ponovitvi. Zato je $l_x(t) = f(l_{x_1}(t - 1), \dots, l_{x_k}(t - 1))$, kjer $l_x(t)$ predstavlja oznako vozlišča x v t -ti ponovitvi, funkcija f pa vrne oznako, ki se pojavlja z najvišjo frekvenco med sosedi. Da bi se izognili cikličnemu nihanju oznak nekaterih vozlišč in s tem zagotovili konvergenco algoritma, avtorji predlagajo uporabo asinhronnega pristopa izmenjave oznak. Pri *asinhronem* posodabljanju se v danem trenutku posodobi samo eno vozlišče [11]. Nova oznaka vozlišča x v t -ti ponovitvi temelji na oznakah sosedov v t -ti ponovitvi, ki so pred x v nekem naključnem vrstnem redu in na oznakah sosedov v $(t - 1)$ -i ponovitvi, ki sledijo x v tem vrstnem redu [16]. V tem primeru lahko posodobitev oznake vozlišča x zapišemo kot $l_x(t) = f(l_{x_1}(t), \dots, l_{x_m}(t), l_{x_{(m+1)}}(t - 1), \dots, l_{x_k}(t - 1))$, kjer so x_1, \dots, x_m sosedje x , ki so že bili posodobljeni v trenutni iteraciji, medtem ko so $x_{(m+1)}, \dots, x_k$ sosedje, ki še niso bili posodobljeni [22].

Za vsako vozlišče $x \in V$, lahko pravilo posodabljanja njegove oznake zapišemo z enačbo, kjer l_x^{new} označuje novo oznako vozlišča x , n je število vozlišč v omrežju, A_{ux} predstavlja elemente matrike sosednosti, l_u je trenutna oznaka vozlišča u in δ je Kroneckerjev delta [2, 19].

$$l_x^{new} = \arg \max_l \left(\sum_{u=1}^n A_{ux} \delta(l_u, l) \right) \quad (3.5)$$

V idealnem primeru bi se iterativen proces moral nadaljevati, dokler nobeno vozlišče v omrežju ne spremeni svoje oznake [22]. Vendar pa lahko obstajajo vozlišča, ki imajo enako maksimalno število sosedov v dveh ali več skupnostih. Ker oznake izbiramo naključno med možnimi kandidati, bi se lahko oznake na teh vozliščih spreminjale skozi iteracije, čeprav oznake njihovih sosedov ostanejo nespremenjene. Zato iterativen postopek izvajamo toliko časa, dokler nima vsako vozlišče vsaj toliko sosedov v svoji skupnosti, kot jih ima z vsako od preostalih skupnosti. Če so l_1, \dots, l_p oznake, ki so trenutno aktivne v omrežju in je $k_i^{l_j}$ število sosedov vozlišča i z oznako l_j , potem se algoritem ustavi, ko za vsako vozlišče i velja: če ima i oznako l_m , potem je $k_i^{l_m} \geq k_i^{l_j} \forall j$. S širitvijo oznak gosto povezane skupine vozlišč hitro dosežejo soglasje glede skupne oznake. Ob koncu iterativnega procesa so vozlišča z enako oznako med sabo združena kot skupnosti [19, 22].

Zgoraj opisan algoritem izmenjave oznak lahko krajše opišemo z naslednjimi koraki:

1. vsakemu vozlišču dodelimo edinstveno oznako – $\forall x \in V : l_x(0) = x$;
2. nastavimo $t = 1$;
3. vozlišča v omrežju uredimo v naključnem vrstnem redu;
4. za vsak $x \in V$, izbran v tem specifičnem zaporedju, naj bo $l_x(t) = f(l_{x_1}(t), \dots, l_{x_m}(t), l_{x_{(m+1)}}(t-1), \dots, l_{x_k}(t-1))$;
5. če ima vsako vozlišče oznako, ki jo ima največje število njegovih sosedov, potem algoritem končamo. Sicer nastavimo $t = t + 1$ in se vrnemo na korak 3;
6. vozlišča z enako oznako tvorijo skupnost.

Prednost algoritma je njegova časovna zahtevnost, ki je skoraj linearna glede na velikost omrežja [16]. Število potrebnih korakov, da se algoritem konča, je težko napovedati, vendar avtorji pristopa trdijo, da pet iteracij zadošča, da se vsaj 95% vozlišč razvrsti pravilno [22]. Očitno je, da se s

povečevanjem števila iteracij povečuje število vozlišč, ki so klasificirani pravilno. Njegova slabost pa je občutljivost na vrstni red, v katerem so oznake vozlišč posodobljene na vsakem koraku [19]. Zato se lahko rešitve ob različnih izvajanjih algoritma med sabo razlikujejo. Poleg tega algoritem lahko odkrije le disjunktne skupnosti [16].

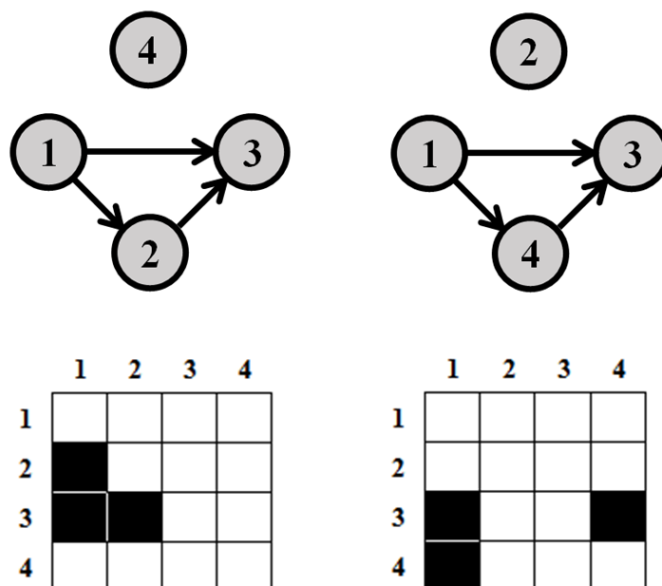
3.4 Bločno modeliranje

Več omrežnih tehnik je občutljivih na izjeme in že ena sama povezava lahko povzroči uvrstitev v slabšo skupino [12]. Ker so empirični podatki v večini nepopolni, potrebujemo orodje za preverjanje strukturnih značilnosti omrežja, ki omogoča izjeme in napake. Takšno prilagodljivo orodje, ki se uporablja predvsem za analizo socialnih omrežij, je *bločno modeliranje* (angl. blockmodeling). Pred definiranjem bločnega modela in bločnega modeliranja je potrebno razložiti še nekatere pojme. Najprej bomo povedali nekaj o matriki, ki jo bločno modeliranje uporablja kot sredstvo za predstavitev omrežij, in permutacijah, nadaljevali bomo s pojmom enakovrednosti in končali s tehniko bločnega modeliranja.

Matrika je učinkovit pripomoček za predstavitev manjšega socialnega omrežja. Poleg tega ponuja tudi vizualno predstavo o strukturi manjših in gostih omrežij, zaradi česar jo uporabljamo pri bločnem modeliranju. Spomnimo se matrike sosednosti, ki prikazuje sosednost vozlišč v omrežju in je sestavljena iz posameznih elementov, ki jih imenujemo celice. Do zdaj smo za pojav povezave v omrežju ustrezen element matrike nastavili na 1 in v nasprotnem primeru na 0. Od sedaj naprej pa prisotnost povezave označimo s polnimi celicami matrike in neprisotnost povezave s praznimi celicami.

Matrika razkriva omrežno strukturo le, če so njena vozlišča premišljeno urejena, na primer glede na eno izmed lastnosti omrežja. Prerazporeditev ali razvrščanje vozlišč v omrežju se imenuje *permutacija* (angl. permutation) omrežja. Gre za seznam, ki za vsako vozlišče pove zaporedno številko vozlišča, pri čemer se struktura omrežja ne spremeni.

Na Sliki 3.5 vidimo, da sta matriki videti drugačni, vendar opisujeta enako strukturo. To pomeni, da lahko isto omrežje predstavimo z več različnimi matrikami. Permutacije uporabimo zato, da bi našli matriko, ki najbolje razkriva strukturo omrežja.



Slika 3.5: Dve omrežji z enako strukturo in permutiranimi vozlišči ter pripadajoči matriki sosednosti.

Pri analizi socialnih omrežij želimo identificirati posameznike oziroma akterje s podobnimi vzorci odnosov, da bi ugotovili, ali so povezani z določeno vlogo. Če za primer vzamemo položaj inštruktorja na univerzi, sta njegovi vlogi na primer mentorstvo študentom in posvetovanje s sodelavci. Vzorec odnosov je drugačen do študentov, do sodelavcev in nadrejenih. Za akterje s podobnimi vzorci odnosov lahko rečemo, da so relacijsko *enakovredni* (angl. *equivalent*), da predstavljajo *razred enakovrednosti* (angl. *equivalence class*) ali da zasedajo *enakovredne položaje* (angl. *equivalent positions*) v omrežju. Pogledali si bomo dve vrsti enakovrednosti – najprej strukturno in kasneje regularno, saj je za slednjo potrebno definirati še nekaj dodatnih pojmov.

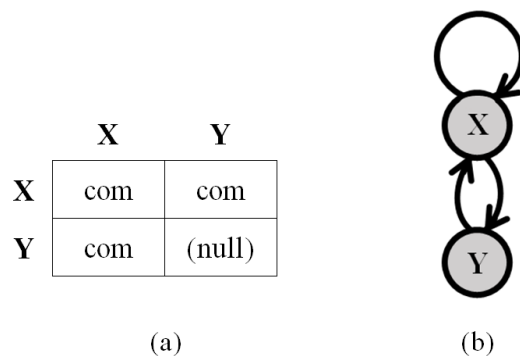
Strukturna enakovrednost (angl. structural equivalence): dve vozlišči sta strukturno enakovredni takrat, kadar sta enako povezani s samim seboj, drug z drugim in z vsemi preostalimi vozlišči. Strukturno enakovredni vozlišči je torej mogoče zamenjati brez posledic za strukturo omrežja. Pri matriki sosednosti so strukturno enakovredna vozlišča videti tako, da imajo enake vrstice in enake stolpce, se pravi enake sosede, kar je razvidno s Slike 3.6. Različnost dveh vozlišč se lahko izrazi z indeksom, ki se giblje od 0 do 1, kjer 0 predstavlja strukturno enakovredni vozlišči, 1 pa povsem različni vozlišči.

	X_1	X_2	X_3	Y_1	Y_2
X_1					
X_2					
X_3					
Y_1					
Y_2					

Slika 3.6: Matrika sosednosti, ki prikazuje strukturno enakovrednost in hkrati idealno jedrno-obrobno zgradbo. Strukturno enakovredna vozlišča so $\{X_1, X_2, X_3\}$ in tudi $\{Y_1, Y_2\}$.

Pri jedrno-obrobni zgradbi smo matriko sosednosti razmejili z vodoravno in navpično črto. Teh črt je lahko več in delijo matriko na pravokotnike – tako imenovane *bloke* (angl. blocks). Blok torej vsebuje celice matrike sosednosti, ki pripadajo prerezu enega ali dveh razredov oziroma skupin. Strukturno omrežja lahko opišemo z analizo blokov matrike sosednosti. Bloki vzdolž diagonale izražajo odnose znotraj razredov, medtem ko bloki izven diagonale predstavljajo odnose med razredi.

Matrika sosednosti omrežja, ki vsebuje razrede strukturne enakovrednosti, ima posebno značilnost, in sicer so njeni bloki bodisi *polni* (angl. complete) bodisi *prazni* (angl. empty), če odmislimo celice na diagonali. To izhaja iz kriterija strukturne enakovrednosti, da imajo enakovredna vozlišča enake vrstice in stolpce. Sedaj lahko matriko sosednosti poenostavimo s skrčitvijo vsakega razreda na eno novo vozlišče in določimo, kateri tipi blokov so dovoljeni ter kje se lahko pojavijo. V primeru strukturne enakovrednosti je tip bloka lahko poln – *com* ali prazen – *(null)*. Ta skrčena matrika oziroma matrika bločnega modela vsebuje vse informacije, ki so bile prisotne v originalni matriki. Slika 3.7 prikazuje matriko bločnega modela preproste jedrno-obrobne zgradbe in grafični prikaz odnosov med razredi enakovrednosti. Usmerjena povezava predstavlja poln blok, odsotnost usmerjene povezave pa ustreza praznemu bloku.



Slika 3.7: (a) Matrika bločnega modela jedrno-obrobne zgradbe s Slike 3.6. (b) Pripadajoče skrčeno omrežje.

Bločni model (angl. block model) opisuje splošno strukturo omrežja in položaj vsakega vozlišča v tej strukturi. Vozlišča dodeljuje razredom in določa dovoljene tipe relacij med razredi. Za njegovo opredelitev torej potrebujemo particijo in matriko bločnega modela. Particija dodeli vozlišča razredom enakovrednosti in matriko sosednosti razdeli na bloke. Matrika bločnega modela pa določa tipe odnosov med razredi.

Do sedaj smo predpostavili, da poznamo bločni model omrežja, se pravi razbitje vozlišč na razrede in matriko bločnega modela, ki navaja dovoljene tipe blokov. V okviru raziskovanja imamo ponavadi dano omrežje in želimo poiskati bločni model, ki najboljše zajame strukturo omrežja. Tehnika za pridobitev iskanega bločnega modela se imenuje *bločno modeliranje* (angl. blockmodeling). V splošnem je bločno modeliranje sestavljeno iz naslednjih treh korakov:

1. določimo število razredov v omrežju;
2. izberemo tipe blokov, ki se lahko pojavijo in, če je mogoče, tudi lokacije teh tipov v matriki bločnega modela;
3. algoritem razdeli vozlišča na določeno število razredov v skladu s pogoji, ki jih določa model in, če je potrebno, izbere končno matriko bločnega modela za model. Tako je bločni model končan.

Prva dva koraka določata matriko bločnega modela, vendar še ne vemo, katera vozlišča spadajo v določen razred. Včasih tudi ni mogoče natančno povedati, kateri tip bloka bo na voljo v katerem delu matrike bločnega modela. To se reši v tretjem koraku. Razumljivo je, da moramo imeti nekaj znanja in pričakovanj o omrežju, da izberemo ustrezno število razredov in opredelimo tip razmerij med razredi, ki so smiselni.

Empirična omrežja redko ustrezajo idealu, ki je predstavljen z matriko bločnega modela. Zato pride do napak, vendar jih je mogoče enostavno preveriti. Recimo, da vemo, katero vozlišče spada v posamezen razred, nato pa preverimo, ali je vsak blok matrike ustreznega tipa v skladu z matriko bločnega modela. Dobimo *oceno napake* (angl. error score), ki kaže, kako dobro realno omrežje ustreza idealni matriki. Ocena napake se uporablja za vrednotenje različnih bločnih modelov za isto omrežje, kjer nižje vrednosti pomenijo boljše prileganje in vrednost 0 predstavlja popolno prileganje. Poglejmo izračun napake na primeru strukturne enakovrednosti s Slike 3.8. Preštejemo manjkajoče celice v blokih, ki naj bi bili polni, in celice v blokih,

ki bi morali biti prazni. Napake pri matriki so označene s križcem in ocena napake je torej enaka 2.

	X_1	X_2	X_3	Y_1	Y_2
X_1					
X_2			×		
X_3					×
Y_1					
Y_2					

Slika 3.8: Napake pri strukturi enakovrednosti.

Tretji korak bločnega modeliranja izvira iz iskanja delitev vozlišč v enakovredne razrede, ki dajo najnižjo oceno napake. Algoritem najprej naključno dodeli vozlišča razredom, določenim v prvem koraku. Izračuna se ocena napake, nato pa jo poskuša zmanjšati z naključnim premikanjem izbranih vozlišč iz ene v drugo skupino ali z zamenjavo dveh vozlišč v različnih skupinah. Ta proces se ponavlja toliko časa, dokler se ocena napake ne izboljšuje več.

Strukturalna enakovrednost zahteva, da imajo enakovredni udeleženci enake sosede. Ker pa je merilo izjemno strogo, je bila definirana druga vrsta enakovrednosti.

Regularna enakovrednost (angl. regular equivalence): vozlišča, ki so regularno enakovredna, ni potrebno, da so povezana z istimi vozlišči, vendar morajo biti povezana z vozlišči v istih razredih. Zaznamo jo lahko s pomočjo bločnega modeliranja, saj obstaja poseben tip bloka, povezan z regularno enakovrednostjo, ki se imenuje *regularni* (angl. regular) blok. Regularni blok vsebuje vsaj eno povezavo v vsaki vrstici in vsakem stolpcu. Regularna enakovrednost omogoča regularne – *reg* in prazne bloke. Ker pa je poln blok vedno tudi regularen blok, to pomeni,

da je strukturna enakovrednost posebna vrsta regularne enakovrednosti. Na Sliki 3.9a vidimo matriko, ki vsebuje dva regularna bloka in pet napak, 3.9b prikazuje pripadajočo matriko bločnega modela in 3.9c matriko napake.

	X	Y ₁	Y ₂	Y ₃	Y ₄	Z ₁	Z ₂	Z ₃
X					X			
Y ₁								
Y ₂								
Y ₃	X							X
Y ₄								X
Z ₁								
Z ₂								
Z ₃						X		

(a)

	X	Y	Z
X	(null)	(null)	(null)
Y	com	reg	(null)
Z	(null)	reg	(null)

(b)

	X	Y	Z
X	0	1	0
Y	1	0	2
Z	0	0	1

(c)

Slika 3.9: (a) Matrika sosednosti, ki vsebuje primer regularne enakovrednosti. (b) Matrika bločnega modela (c) Matrika napake.

Namesto uporabe ene vrste enakovrednosti lahko uporabimo katero koli kombinacijo dovoljenih tipov za opredelitev omrežja. Za vsak posamezen blok navedemo tip, ki je dovoljen. Ta način je znan kot *posplošeno bločno modeliranje* (angl. *generalized blockmodeling*).

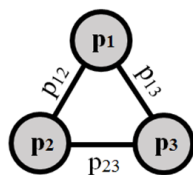
Bločno modeliranje je močna tehnika za analizo predvsem gostih omrežij, vendar potrebuje pravi prispevek raziskovalca, da dobimo zanimive rezultate. Ponavadi je vredno uporabiti več nekoliko različnih bločnih modelov, na primer z različnim številom razredov ali drugimi omejitvami o odnosih med bloki. Število bločnih modelov, ki se lahko prilegajo socialnemu omrežju, je visoko, zato moramo k bločnemu modeliranju pristopiti z jasnimi predstavami in pričakovanji o celotni strukturi omrežja.

3.5 Stohastično bločno modeliranje

Stohastični bločni model (angl. stochastic block model) je dobro uveljavljen in pogosto uporabljen verjetnostni ali generativni model za naključna modularna omrežja [13]. Uporablja se za odkrivanje strukture v omrežnih podatkih, s prileganjem modela opazovanim podatkom [25]. Parametri prileganja nam povedo najboljšo delitev omrežja na skupnosti.

Stohastični bločni model temelji na *stohastični enakovrednosti* (angl. stochastic equivalence) [14]. Vozlišči v istem bloku sta *stohastično enakovredni* (angl. stochastically equivalent), če lahko zamenjamo njune parametre, ne da bi spremenili katero od verjetnosti porazdelitve.

Vsako vozlišče je dodeljeno natanko enemu bloku in verjetnost povezave med dvema vozliščema je odvisna izključno od njunega članstva v bloku [13]. Stohastični bločni model predpostavlja, da je vsako vozlišče $i \in \{1, \dots, n\}$ v samo enem izmed R razredov. Vsako vozlišče i ima oznako l_i in $l_i = r$ nakazuje članstvo vozlišča i v razredu $r \in \{1, \dots, R\}$. Verjetnost, da ima vozlišče i povezavo do vozlišča j , je odvisna od l_i in l_j , torej $p_{ij} = p_{l_i l_j}$. Verjetnost, da je vozlišče i element razreda r , označimo s $P(l_i = r) = p_r$. Sedaj lahko definiramo $R \times R$ matriko z verjetnostmi povezav med razredi. Element te matrike je p_{rs} in predstavlja verjetnost povezave med razredom r in razredom s . Model grafa na Sliki 3.10a prikazuje verjetnosti, da je vozlišče dodeljeno enemu izmed treh razredov in verjetnosti povezav med temi razredi. Slika 3.10b podaja matriko z verjetnostmi povezav med posameznimi razredi.



(a)

	1	2	3
1	p_{11}	p_{21}	p_{31}
2	p_{12}	p_{22}	p_{32}
3	p_{13}	p_{23}	p_{33}

(b)

Slika 3.10: (a) Graf na treh vozliščih z označenimi verjetnostmi. (b) Matrika z verjetnostmi povezav med razredi.

Ustvarimo naključen usmerjen graf G na n vozliščih, z elementom matrike sosednosti $A_{ij} = 1$, če obstaja povezava od j do i , sicer $A_{ij} = 0$. Verjetnost generiranja specifične $\{0, 1\}^{n \times n}$ matrike sosednosti A in s tem grafa G , skupaj z dodelitvami vozlišč v razrede l glede na nabor parametrov $\theta = \{R, p_r, p_{rs}\}$ je podana z enačbo 3.6. V primeru neusmerjenega omrežja produkt poteka po parih, kjer je $i < j$.

$$P(A, l | \theta) = \prod_{i \neq j} \left[p_{l_i l_j}^{A_{ij}} (1 - p_{l_i l_j})^{(1 - A_{ij})} \right] \prod_i p_{l_i} \quad (3.6)$$

Stohastični bločni model lahko realizira široko paleto različnih omrežnih struktur. Na primer diagonalna verjetnostna matrika ponazarja omrežje z nepovezanimi komponentami, medtem ko bi dodajanje elementov izven diagonale ustvarilo običajno strukturo skupnosti – s pogostimi povezavami znotraj skupnosti in redkimi medsebojnimi povezavami.

Poglavje 4

Orodja za analizo omrežij

Zanimanje za metode vizualne analize, predvsem socialnih omrežij, se povečuje. Oblikovani so bili številni pristopi [7]. Veliko standardnih omrežnih algoritmov je časovno in prostorsko zahtevnih in zaradi tega neprimernih za analizo velikih omrežij s tisoče vozlišči in povezavami [4]. Taka omrežja ni mogoče obravnavati učinkovito z uporabo standardnih orodij za analizo omrežij, ki večinoma temeljijo na matrični predstavitvi in so s tem omejeni na omrežja srednje velikosti, kvečjemu nekaj deset ali sto vozlišči.

Opisali bomo tri prosto dostopna programska orodja za analizo in vizualizacijo velikih omrežij, ki jih uporabimo v diplomskem delu. Od pristopov, predstavljenih v Poglavju 3, Gephi omogoča uporabo metode Louvain za optimizacijo modularnosti, Visone ponuja iskanje jedrno-obrobne zgradbe ter metodo Louvain za optimizacijo modularnosti, Pajek pa omogoča metodo Louvain za optimizacijo modularnosti, pristop bločnega modeliranja in s tem iskanje jedrno-obrobne zgradbe. V nadaljevanju sledi podrobnejša predstavitev teh orodij.

4.1 Gephi

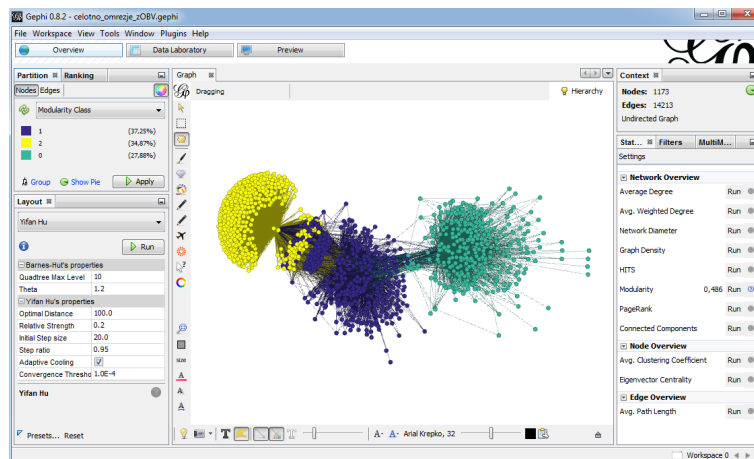
*Gephi*¹ je prosto dostopno programsko orodje za raziskovanje in vizualizacijo omrežij. Leta 2008 je z njegovim razvijanjem začela skupina študentov na Univerzi Compiègne, Francija [17]. Namenjeno je ljudem, ki želijo raziskovati ter razumevati grafe in je najbolj popularno programsko orodje, s katerim je moč analizirati vse vrste omrežij. Še posebej primerno je za omrežja z dodatnimi atributi, katerih vrednosti so povezane z vsakim vozliščem ali vsako povezavo. Dela lahko z velikimi omrežji, ki vsebujejo do sto tisoč vozlišč in milijon povezav. Omogoča enostaven in širok dostop do podatkov v omrežju [3]. Lahko ga uporabimo za analiziranje, filtriranje, gručenje, obdelavo, njegov poudarek je na dinamični vizualizaciji omrežij in interakciji s podatki v realnem času.

Gephi ima vključeno funkcijo za študij *časovno spremenljivih omrežij* (angl. time-varying networks), katerih struktura ali vsebina se spreminjata skozi čas [17]. S prilagajanjem vgrajene časovnice lahko proučujemo razvoj omrežnih struktur in razvoj lastnosti vozlišč ter povezav skozi čas. Omogoča delo s številnimi datotečnimi formati. Formati, s katerimi lahko definiramo grafe, so CSV, GDF, GEXF, GML, UCINET DL, GraphML, Pajek NET, VNA in njegov lasten format Gephi. Poleg teh ponuja tudi slikovne formate PNG, SVG in PDF. Uporabniški vmesnik Gephi se osredotoča na ustvarjanje slik omrežja v realnem času. Interakcija z omrežjem je preprosta, uporabniki eksperimentirajo z različnimi vizualnimi konfiguracijami in izid vsake akcije je viden takoj. Več algoritmov lahko poteka istočasno v ločenih delovnih prostorih brez vpliva na uporabniški vmesnik [3].

Glavne komponente znotraj Gephija so *pregled* (angl. overview), kjer lahko izračunamo razne omrežne mere in vzpostavimo vizualizacijo omrežja, *podatkovni laboratorij* (angl. data laboratory), kjer pregledamo ter uredimo omrežne podatke in *predogled* (angl. preview), kjer konfiguriramo nastavitve upodabljanja in dobimo predogled vizualizacije.

¹Gephi programsko orodje za analizo omrežij. Dostopno na: <https://gephi.org>

Okno *pregleda* razdelimo na šest delov, ki jih lahko razberemo s Slike 4.1 in jih predstavimo v nadaljevanju.



Slika 4.1: Grafični uporabniški vmesnik Gephi.

Zgoraj levo se nahaja modul za delo s *particijami* (angl. partition), kjer lahko spreminjamo barve vozlišč in povezav glede na njihove kategorične attribute. Poleg leži modul *rangiranja* (angl. ranking), kjer lahko spreminjamo barve in velikosti vozlišč, povezav oziroma oznak glede na numerične attribute.

Pod njima se nahaja modul *prikaza* (angl. layout), kjer si izberemo enega od razpoložljivih algoritmov prikaza in prilagodimo njegove attribute. Attribute omogočajo spremembo nastavitev prikaza med samim izvajanjem, kar omogoča nadzor algoritma in kot rezultat dobimo estetsko prijetno predstavitev.

Osrednji del uporabniškega vmesnika je okno *grafa* (angl. graph), ki prikazuje vizualizacijo omrežja in omogoča interaktivno raziskovanje omrežja. Na levem delu so orodja, ki omogočajo interaktivno izbiro vozlišč ter povezav, ročno spreminjanje barv ter velikosti vozlišč, dodajanje vozlišč ter povezav in spreminjanje atributov vozlišč. Na spodnjem levem delu so orodja, s katerimi lahko ponastavimo velikost, barve in

oznake v omrežju. Na dnu vizualizacijskega okna pa lahko spreminjamo barve, velikosti, pisave in druge značilnosti, ki vplivajo na vsa vozlišča, povezave ali oznake.

Desno od vizualizacijskega okna se nahaja modul *statistike* (angl. statistics) za izračun statistik omrežja, vozlišč ali povezav. Nekatere izmed mer, ki jih je mogoče izračunati, so povprečna stopnja (angl. average degree), povprečna utežena stopnja (angl. average weighted degree), premer omrežja (angl. network diameter), gostota grafa (angl. graph density), modularnost (angl. modularity), povezane komponente (angl. connected components), povprečni koeficient gručenja (angl. average clustering coefficient) ter povprečna dolžina poti (angl. average path length). Ko izvedemo izračune, so v podatkovnem laboratoriju na voljo novi atributi, ki jih lahko vključimo v vizualizacijo.

Poleg statistike se nahajajo številni *filtri* (angl. filters), ki lahko prikažejo množico vozlišč in povezav, za katere veljajo določene lastnosti.

Okno *podatkovnega laboratorija* vsebuje podatkovno tabelo vozlišč in tabelo povezav, skupaj s pripadajočimi atributi. Uvozimo lahko preglednice s podatki o vozliščih ter povezavah in upravljamo s stolpci, ki vsebujejo podatke.

Pri oknu *predogleda* lahko konfiguriramo nastavitve upodabljanja, kot so velikost, barva, debelina črte in druge lastnosti vozlišč, povezav oziroma oznak. Operacije se izvedejo na celotni vizualizaciji, lastnosti posameznih vozlišč pa se spreminja v oknu pregleda.

4.2 Visone

Programsko orodje *Visone*² je namenjeno interaktivni analizi ter vizualizaciji socialnih omrežij in je prosto dostopno za akademske ter raziskovalne namene. Razvit je s strani interdisciplinarne raziskovalne skupine, ustanovljene na Univerzi Konstanz, Nemčija, in leta 2002 je bila objavljena prva različica. Orodje je namenjeno raziskovanju in poučevanju socialnih omrežij, s posebnim poudarkom na vizualnih sredstvih za raziskovanje in analizo omrežnih podatkov [7]. Zasnovano in oblikovano je tako, da omogoča tako strokovnjakom kot tudi laikom uporabo inovativnih in naprednih vizualnih metod na preprost način ter z ustrezno natančnostjo. Zapletene vrste analiz in rokovanje s podatki je pregledno, lahko dostopno in enostavno za uporabo.

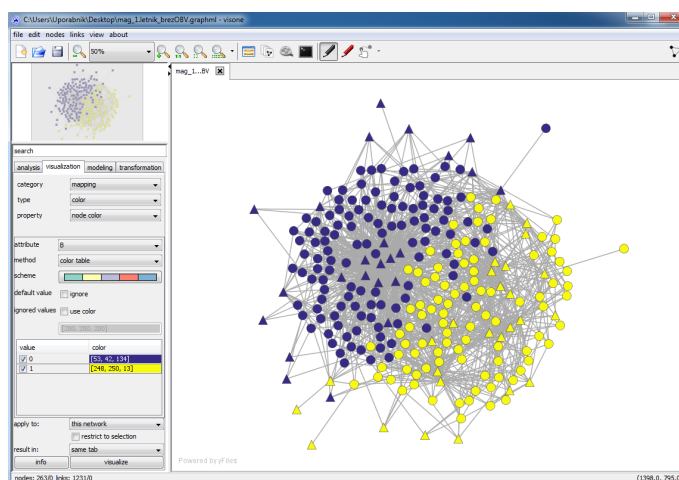
Visone omogoča številne standardne omrežne podatkovne formate, ki so pogosti pri analizi socialnih omrežij. Formati, ki jih je mogoče tako uvoziti kot tudi izvoziti, so na primer CSV, DOT, GML, GraphML, UCINET DL in Pajek NET. Poleg tega ponuja še izvoz v številne slikovne formate, kot so BMP, GIF, JPG, PDF, PNG ter SVG.

Interaktivni grafični uporabniški vmesnik je prilagojen socialnim omrežjem in vsebuje specializirane komponente za analizo ter vizualizacijo [7]. Uporablja terminologijo, skladno z literaturo socialnih omrežij, in ponuja različne poglede na podatke. Grafični uporabniški vmesnik je prikazan na Sliki 4.2 in ga lahko razdelimo na pet komponent.

Na najvišji ravni so *meniji* (angl. menus), ki med drugim nudijo osnovne operacije za delo z datotekami, izbiranje vozlišč, operacije z vozlišči ter povezavami in spreminjanje pogleda na omrežje.

Pod meniji se nahaja *orodna vrstica* (angl. toolbar), ki omogoča neposreden dostop do številnih pogosto uporabljenih operacij. Visone ponuja tudi preklapljanje med dvema načinoma – *načinom analize* (angl. analysis mode) in *načinom urejanja* (angl. edit mode).

²Visone programsko orodje za analizo omrežij. Dostopno na: <http://visone.info>



Slika 4.2: Grafični uporabniški vmesnik Visone.

Osrednji del grafičnega uporabniškega vmesnika je *območje omrežja* (angl. network area). Poleg prikazovanja omrežja omogoča tudi neposredno spreminjanje lastnosti vozlišč ter povezav, grupiranje vozlišč, ustvarjanje, brisanje in izbiranje vozlišč ter povezav.

Na levem delu Visone okna leži *območje predogleda* (angl. overview area). Prikazuje celotno omrežje glavnega okna, skupaj s sivim poljem, ki obdaja del trenutno prikazanega omrežja v osrednjem delu.

Zadnji, a vendar precej pomemben del Visone okna, so *zavihki* (angl. tabs). Nahajajo se pod območjem predogleda in omogočajo dostop do številnih operacij. Pomembni so predvsem trije zavihki – *analiza* (angl. analysis), *vizualizacija* (angl. visualization) in *transformacija* (angl. transformation).

Analiza razlikuje med dvema glavnima opraviloma – *indeksiranjem* (angl. indexing) in *grupiranjem* (angl. grouping). Pri indeksiranju računamo lastnosti na ravni vozlišč in na ravni povezav. Ločimo med več razredi: središčnost (angl. centrality), gostota (angl. density), razdalja (angl. distance) in hierarhija (angl. hierarchy) vozlišč ter redkost

(angl. sparsification), središčnost (angl. centrality) in umeščenost (angl. embeddedness) povezav. Pri grupiranju iščemo posebne skupine vozlišč ter povezav in lahko izbiramo med štirimi razredi: gručenjem (angl. clustering), enakovrednostjo (angl. equivalence), kohezivnostjo (angl. cohesiveness) in povezanostjo (angl. connectedness). Pri vsakem od naštetih razredov indeksiranja in grupiranja nato izberemo mero, ki jo želimo izračunati, in nastavimo dodatne attribute, ki so nam ponujeni.

Algoritmi vizualizacije spremenijo grafični izgled omrežja. Visone razlikuje med tremi glavnimi kategorijami vizualizacije – *prikazom* (angl. layout), *mapiranjem* (angl. mapping) in *geometrijo* (angl. geometry). Vsaka od kategorij omogoča nastavljanje dodatnih parametrov, s katerimi določimo dodatne lastnosti.

Zavihek transformacij omogoča spreminjanje strukture omrežja glede na neko določeno pravilo. Visone omogoča transformacije na ravni vozlišč, na ravni povezav in na ravni omrežja, kjer se pri vsaki ravni izbira med številnimi dodatnimi operacijami.

4.3 Pajek

*Pajek*³ je programsko orodje za analizo in vizualizacijo velikih omrežij [20]. Najnovejšo različico programa, dokumentacijo in spremljajoče gradivo je mogoče prenesti in uporabljati brezplačno za nekomercialno uporabo z domače spletne strani. Razvila sta ga Vladimir Batagelj in Andrej Mrvar iz Univerze v Ljubljani, začetki razvoja pa segajo v leto 1996. Pajek zagotavlja orodja za analizo ter vizualizacijo omrežij in je namenjen raziskovalcem iz različnih področij [5]. Čeprav je bil razvit predvsem za analizo velikih omrežij, se pogosto uporablja tudi za vizualizacijo majhnih omrežij. Kljub temu, da je na voljo že 20 let, je še vedno glavno prosto dostopno orodje, ki lahko obvladuje zelo velika omrežja [20]. Dela lahko z omrežji, ki vsebujejo

³Pajek programsko orodje za analizo omrežij. Dostopno na: <http://mrvar.fdv.uni-lj.si/pajek/>

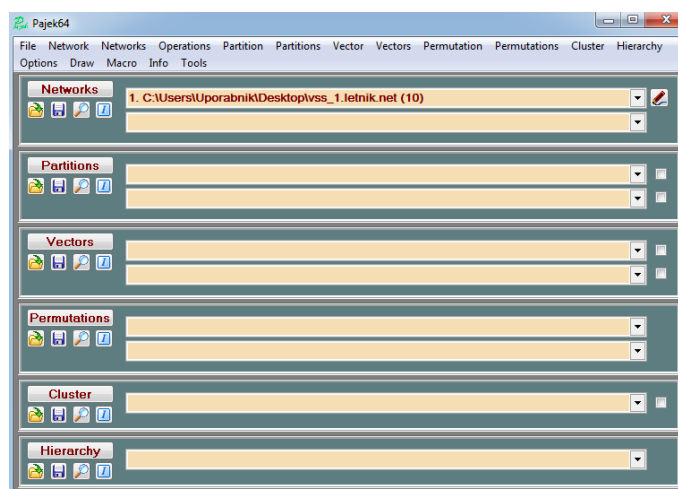
milijardo vozlišč, dočim pa glede števila povezav ni omejitev. Priporočljivo je, da vizualizacije omrežij ne vsebujejo več kot nekaj tisoč vozlišč, saj je za zelo velika omrežja postopek risanja zamuden ter slika omrežja vizualno neprivlačna [12].

Poleg lastnih dveh omrežnih datotečnih formatov – Pajek NET za omrežja in Pajek MAT za matrike, podpira tudi nekaj drugih. Vega in UCINET DL se uporabljata za grafe oziroma omrežja, GEDCOM za rodovniške podatke, formati Ball and Stick, Mac Molecule ter MDL MOL pa so bili razviti za kemijo in niso pogosto uporabljeni v analizi socialnih omrežij. Pajek podpira tudi več izhodnih grafičnih formatov, ki jih je mogoče proučevati s posebnimi 2D in 3D pregledovalniki datotek. Za izvoz dvodimenzionalnih objektov so na voljo EPS/PS, SVG, JPEG ter Bitmap, za tridimenzionalne objekte pa X3D, Kinemages, VRML in MDL MOL.

Pajek podpira običajna neusmerjena in usmerjena omrežja, dvodelna omrežja, ki smo jih predstavili že v Poglavju 2, *časovna omrežja* (angl. temporal networks), kjer gre za spreminjajoča se omrežja skozi čas, *označena omrežja* (angl. signed networks), katera se uporabljajo za shranjevanje odnosov, ki so bodisi pozitivni bodisi negativni, *večrelacijska omrežja* (angl. multi-relational networks), kjer je na isti množici vozlišč opredeljenih več odnosov) in *aciklična omrežja* (angl. acyclic networks), ki ne vsebujejo ciklov [20]. Za delo s temi vrstami omrežij so na voljo tudi posebne operacije.

Analiza in vizualizacija velikih omrežij sta implementirani z uporabo šestih podatkovnih struktur. Le-te so *omrežje*, ki je glavni objekt, *particija*, ki za vsako vozlišče pove, kateremu razredu pripada, *vektor*, ki za vsako vozlišče pove številčne lastnosti, *permutacija*, kjer gre za prerazporejanje vozlišč, *skupina*, ki določa podmnožico vozlišč, in *hierarhija*, kjer so vozlišča hierarhično urejena ter dobimo splošno drevesno strukturo na vozliščih. Struktura glavnega okna Pajka je organizirana glede na zgoraj omenjene podatkovne tipe objektov, ki jih sprejmejo kot vhod, kar lahko vidimo na Sliki 4.3 [5].

Do številnih operacij, ki za svoje izvajanje potrebujejo ustrezne podatkovne strukture, dostopamo preko menijev glavnega okna. Ker je operacij,



Slika 4.3: Grafični uporabniški vmesnik Pajek.

ki jih Pajek nudi res veliko, bomo navedli le del, več podrobnosti pa se nahaja v referenčnem priročniku [6]. Pajek omogoča povsem osnovne operacije, kot so branje, urejanje in izvoz vseh podatkovnih tipov. Poleg teh operacij ponuja tudi bolj kompleksne, na primer razne transformacije (angl. transform), naključna omrežja (angl. random networks), komponente (angl. components), številčenje (angl. numbering), soseščine (angl. k-neighbors), poti (angl. paths), maksimalni pretok (angl. maximum flow), štetje (angl. count), unija (angl. union), presek (angl. intersection), razlika (angl. difference), krčenje (angl. shrinking), odkrivanje skupnosti (angl. community detection), poenostavitve (angl. simplifications), različnosti (angl. dissimilarity), prerazporejanje (angl. reordering), jedra (angl. cores), triade (angl. triads), mere središčnosti (angl. centrality measures), barvanja (angl. coloring), bločno modeliranje (angl. blockmodeling), ustvarjanje particij (angl. create partition), gručenje (angl. clustering) in še mnogo drugih operacij, ki delajo nad omejenimi podatkovnimi strukturami.

Za pridobitev rezultata mora biti ponavadi več operacij izvedenih v zaporedju [20]. Pogosto uporabljena zaporedja elementarnih operacij lahko definiramo kot *makroje* (angl. macros) in jih izvajamo kot en sam ukaz. S tem

lahko orodje Pajek prilagodimo posebnim skupinam uporabnikov iz različnih področij. Pajek omogoča vizualizacijo omrežij z v naprej pripravljenimi *prikazi* (angl. layouts). Implementiranih je več standardnih algoritmov za avtomatsko risanje grafov. Poleg tega ponuja tudi orodja za ročno urejanje grafov, s katerimi lahko dodatno urejamo prikaze ali pa jih proučujemo v podrobnosti.

Poleg standardne različice Pajka obstaja tudi posebna različica, imenovana *PajekXXL* [20]. Gre za posebno izdajo orodja Pajek, kjer je poraba spomina precej nižja. Za enako redko omrežje potrebuje vsaj 2- do 3-krat manj fizičnega pomnilnika kot Pajek, poleg tega pa lahko obvlada omrežja z do dve milijardi vozlišč. PajekXXL se običajno uporablja za velika omrežja, ki se ne prilegajo razpoložljivemu spominu računalnika. Ko je nekaj zanimivih delov omrežja najdenih in pridobljenih, se za nadaljnjo analizo lahko uporablja standardna različica Pajka.

Poglavje 5

Rezultati in diskusija

Namen diplomskega dela je s pomočjo analize omrežij poiskati skupine predmetov glede na to, kako jih študentje soizbirajo. Nekateri od pristopov delajo dobro na enih omrežjih in drugi na drugih. Posledično se tudi rezultati razlikujejo od enega do drugega pristopa. Zato podamo in interpretiramo le tiste rezultate, ki so z našega vidika zanimivi ter uporabni.

Različna orodja za enak pristop vrnejo podobne rezultate. Zato se lahko omejimo na eno samo orodje, kjer je pristop na voljo. Z modularnostjo delamo v Gephiju, z jedrno-obrobno zgradbo v Visone in z bločnim modeliranjem v Pajku. Pri izmenjavi oznak [22] in stohastičnem bločnem modeliranju [25] uporabimo implementacije avtorjev pristopov in rezultate nato uvozimo v enega od obravnavanih orodij.

Zaradi enostavnosti bločno modeliranje izvajamo samo na enodelnih neuteženih omrežjih. Gre za to, da so v primeru dvodelnega omrežja matrike sosednosti zelo velike ter prazne in so tako manj informativne. Razlike med rezultati z in brez uteži so majhne, je pa matrika sosednosti bolj berljiva v primeru neutežene verzije. Uporabljamo strukturno enakovrednost, ker so pri regularni enakovrednosti omejitve manj stroge in s tem hitreje dosegljive. Stohastično bločno modeliranje zato izvajamo samo na enodelnih omrežjih.

5.1 Podatki o soizbirnosti predmetov

Ker je na visokošolskem in magistrskem študiju izbirnost veliko večja kot na univerzitetnem, je to dovolj očiten razlog, da se omejimo le na ta dva študijska programa. Delamo s podatki visokošolskega in magistrskega študija Fakultete za računalništvo in informatiko med leti 2012 in 2015. Le-ti vsebujejo študente in predmete, ki so jih študentje izbrali tekom študija. Poleg tega sta dodana še dva atributa – študijski program in letnik študija. Opisan nabor podatkov tako zadošča vsem potrebam te diplomske naloge.

Podatke lahko predstavimo z dvodelnim neusmerjenim omrežjem, kjer so študentje povezani s predmeti. Omrežje sestavljajo pari *identifikator študenta*¹ ter *šifra predmeta* in so določeni enolično. Identifikator študenta ni vpisna številka, šifra predmeta pa je določena s strani fakultete.

Študentje visokošolskega študija izbirajo samo med vnaprej ponujenimi predmeti. Na magistrskem študiju lahko poleg že ponujenih predmetov izberejo tudi predmete univerzitetnega študija. Kateremu študijskemu programu pripada posamezen predmet, je razvidno iz prvih treh števk šifre predmeta. Številke 637 označujejo predmete visokošolskega študija, 635 predmete magistrskega študija in številke 632 predmete univerzitetnega študija.

Kot smo že omenili, želimo poiskati skupine predmetov glede na to, kako jih izbirajo študentje. Poudarek je torej na predmetih, ki so skupaj izbrani s strani študentov in ne na študentih samih. Najti skupino študentov, ki so izbirali podobne predmete, za to diplomsko delo ni smiselno.

5.2 Priprava podatkov

Po pregledu vseh predmetov, ki se pojavijo v omrežju, odstranimo štiri izjeme, ki jih s šifro predmeta na spletni strani fakultete ni mogoče najti. Gre za eno pojavitev predmeta s šifro 63759, ki predstavlja Računalniški praktikum, dve pojavitvi predmeta 63760 Izbirni predmet druge fakultete, tri poja-

¹Zaradi varovanja osebnih podatkov so bili vsi podatki anonimizirani.

vitve predmeta 63758 Fizika in devet pojavitev predmeta 63751 Uporabniška programska oprema. Opazimo lahko, da je njihova stopnja izbranosti nizka, zato ne naredimo veliko škode, če jih zanemarimo. Poleg tega se je predmet 63730 Odkrivanje zakonitosti iz podatkov preimenoval v predmet 63765 Podatkovno rudarjenje, tako da vse njegove pojavitve ustrezno popravimo. Zaradi razloga enostavnosti predmete z več možnimi izvedbami prevedemo na predmete z eno samo izvedbo. To naredimo tako, da odstranimo veliko tiskano črko s konca šifre. Odstranimo tudi tiste pojavitve, kjer so študentje vpisali dodatno leto.

Sedaj imamo urejeno dvodelno omrežje, iz katerega najprej izločimo visokošolski ter magistrski študij in dobimo dve novi omrežji. Nato še oba študijska programa ločimo glede na letnike. Visokošolski študijski program ločimo na prvi, drugi in tretji letnik, magistrski pa na prvi in drugi letnik. Na teh osmih dvodelnih omrežjih izvedemo še projekcijo na enodelno omrežje predmetov, kjer dva predmeta povežemo, če ju je izbral isti študent. Rezultat projekcije je manjše omrežje. Zaradi tega so potrebne uteži, ki povedo, kolikokrat se posamezen par predmetov pojavi pri različnih študentih. Pare predmetov, katerih utež je enaka ena, odstranimo, saj nam nič ne povedo o soizbirnosti glede na to, da jih soizbere le en študent. Tako iz dvodelnih kot tudi iz enodelnih omrežij naredimo še eno različico omrežij, kjer odstranimo obvezne predmete. Med sabo lahko primerjamo dvodelna in enodelna omrežja ali pa omrežja z in brez obveznih predmetov.

Tabeli 5.1 in 5.2 prikazujeta osnovne podatke vseh omenjenih različic omrežij za dvodelno in enodelno omrežje z obveznimi predmeti. Tabeli A.1 in A.2 podajata osnovne podatke za dvodelno in enodelno omrežje brez obveznih predmetov. Od leve proti desni si sledijo vrsta omrežja, število vozlišč, število povezav in povprečna stopnja. Na teh omrežjih izvajamo pristope, opisane v Poglavju 3. Ob koncu vsakega pristopa shranimo vizualni prikaz ter tabelo, ki za vsako vozlišče pove, kateri skupini je dodeljeno. Rezultate prikažemo grafično na tri načine, predstavljene v nadaljevanju.

Porazdelitev predmetov glede na *laboratorije*, katerim pripadajo: predmet priredimo laboratoriju, kateremu pripada nosilec predmeta, pri čemer so imena laboratorijev anonimizirana. Narišemo stolpčni diagram, kjer so na abscisni osi laboratoriji, na ordinatni osi pa število predmetov.

Porazdelitev predmetov glede na *module*, v katere so dodeljeni: za predmete magistrskega študija so moduli že določeni. Predmete visokošolskega in univerzitetnega študija pa smo ločili na računalništvo, informatiko in drugo, glede na opis predmeta na spletu in so ekvivalent modulom. Narišemo stolpčni diagram, kjer so na abscisni osi moduli, na ordinatni osi pa število predmetov.

Porazdelitev predmetov glede na *število študentov*: za vsak predmet pogledamo, koliko študentov ga je izbralo. Narišemo stolpčni diagram, kjer je na abscisni osi število študentov, na ordinatni osi pa delež predmetov.

Pri vsakem od omenjenih prikazov skupine, ki jih dobimo z enim od pristopov, ponazorimo z različnimi barvami. Zaradi velikega števila vozlišč in povezav so rezultati bolj vidni v elektronski verziji. V Tabeli A.3 so za vsak predmet podani moduli, katerim je predmet določen in laboratorij, kateremu pripada nosilec tega predmeta.

	n	m	$\langle k \rangle$
Vsi študiji	1174	14322	24,2
MAG (vsi letniki)	335	2987	17,8
MAG (1. letnik)	267	2079	15,6
MAG (2. letnik)	205	908	8,8
VŠŠ (vsi letniki)	861	11335	26,1
VŠŠ (1. letnik)	539	5252	9,7
VŠŠ (2. letnik)	388	3333	17,1
VŠŠ (3. letnik)	386	2750	14,2

Tabela 5.1: Dvodavno omrežje z obveznimi predmeti.

	n	m	$\langle k \rangle$
Vsi študiji	112	2977	53,2
MAG (vsi letniki)	53	911	34,4
MAG (1. letnik)	51	640	25,1
MAG (2. letnik)	46	362	15,7
VŠŠ (vsi letniki)	59	1525	51,7
VŠŠ (1. letnik)	10	45	9,0
VŠŠ (2. letnik)	42	544	25,9
VŠŠ (3. letnik)	50	698	27,9

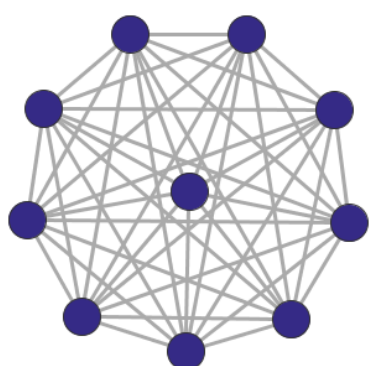
Tabela 5.2: Enodelno omrežje z obveznimi predmeti.

5.3 Rezultati in diskusija

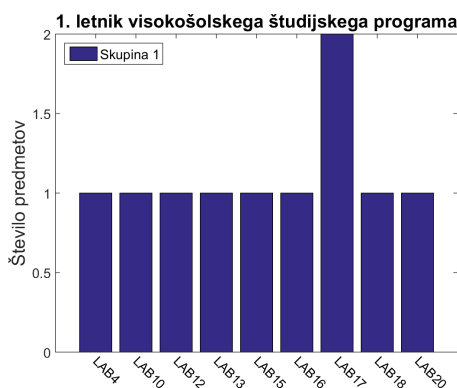
5.3.1 1. letnik visokošolskega študijskega programa

Na visokošolskem študijskem programu je izbirnost predmetov možna samo v drugem in tretjem letniku. Za prvi letnik visokošolskega študija noben pristop ne vrne uporabnih rezultatov. Vseh deset predmetov, ki se pojavijo v omrežju, je dodeljenih eni sami skupini. To jasno prikazuje primer vizualizacije iz Visone na Sliki 5.1a, kjer so vsa vozlišča iste barve. Ker so vsi predmeti prvega letnika obvezni, nas dobljeni rezultat ne preseneti. Gre za polni graf na desetih vozliščih. To pomeni, da velja: $\forall i \in V: k_i = 9$.

Če pogledamo še porazdelitev predmetov po laboratorijih, prikazano na Sliki 5.1b, vidimo, da omenjenih deset predmetov pripada devetim laboratorijem. To lahko interpretiramo tako, da fakulteta določi zelo širok nabor predmetov. S tem lahko vsak študent že v prvem letniku študija odkrije področja, ki ga zanimajo.



(a) Graf omrežja.



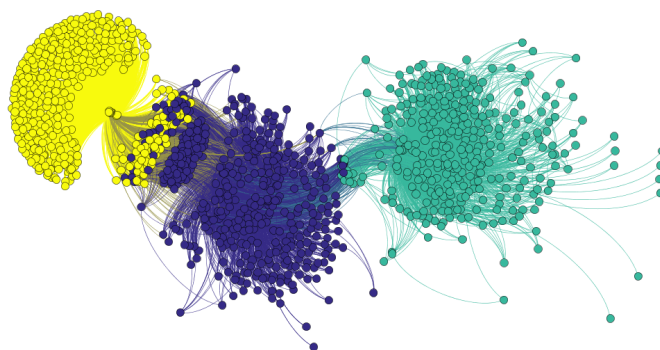
(b) Porazdelitev predmetov po laboratorijih.

Slika 5.1: 1. letnik visokošolskega študijskega programa.

5.3.2 Vsi študijski programi

Iz rezultatov, ki jih dobimo pri obdelavi podatkov celotnega omrežja, vidimo, kako se obravnavana študijska programa med sabo ločita. Čeprav bi želeli dobiti še kakšne dodatne informacije glede izbirnih predmetov, nam pristopi v tem primeru tega ne vrnejo. Ti rezultati za nas niso najbolj uporabni in informativni. Kljub temu je že to, da pristopi vrnejo razbitje na študijska programa, dovolj dober rezultat v primerjavi s tem, če bi dobili eno skupino.

Optimizacija modularnosti (glej Poglavje 3.2) tako za dvodelne kot tudi za enodelne različice celotnega omrežja z obveznimi predmeti vrne razbitje na tri skupnosti. Slika 5.2 prikazuje graf, ki ga kot rezultat dobimo v Gephiju po izvedbi optimizacije modularnosti na *dvodelnem omrežju z obveznimi predmeti*. Graf razpade na tri skupnosti. Rumeno obarvan del vsebuje vse obvezne predmete 1. letnika visokošolskega študijskega programa, modro obarvan del vsebuje preostale predmete visokošolskega študija, zeleni del pa vsebuje predmete, ki jih študentje izbirajo na magistrskem študijskem programu. Vidimo, da so skupnosti med sabo povezane, kar nakazuje na to, da študentje nadaljujejo študij iz letnika v letnik in z visokošolskega študijskega programa na magistrski program. Ker gre za dvodelno omrežje, Slika A.1a prikazuje graf, kjer različni barvi označujeta predmete in študente.



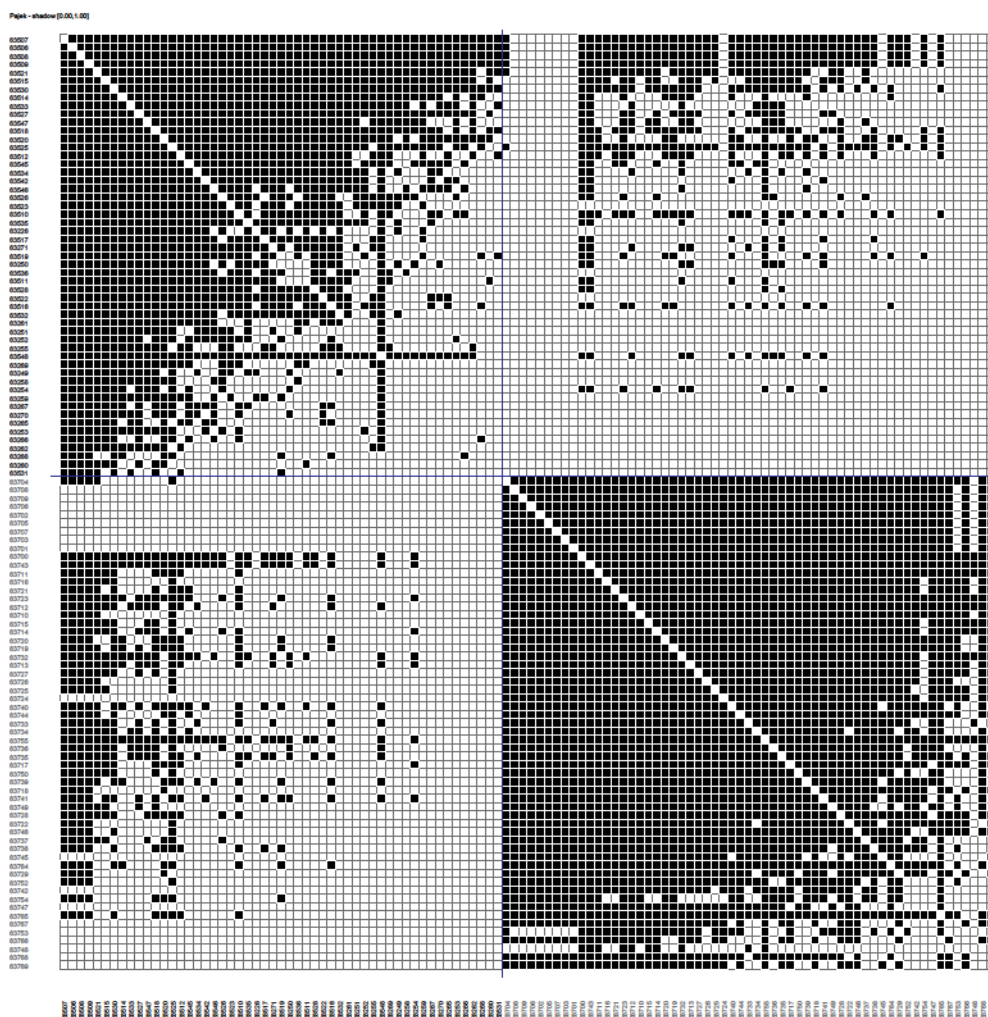
Slika 5.2: Optimizacija modularnosti na dvodelnem omrežju vseh študijskih programov z obveznimi predmeti.

Optimizacija modularnosti na *enodelnem omrežju z obveznimi predmeti* vrne zelo podoben rezultat. Razlikuje se le v tem, da sta v rumeno obarvani skupnosti poleg obveznih predmetov 1. letnika visokošolskega študija dodana še dva predmeta, ki sta bila prej dodeljena modri skupnosti. Gre za predmeta 63767 Tehnične veščine, ki so ga študentje izbrali 14-krat, in predmet 63769 Programski jezik C, ki je bil izbran 26-krat. Če to primerjamo z obveznimi predmeti, ki so v povprečju izbrani 525-krat, vidimo, da je izbranost veliko manjša. Zaradi tega tudi ni povsem jasno, zakaj se ta dva predmeta priključita obveznim predmetom 1. letnika visokošolskega študija.

Optimizacija modularnosti na *dvodelnem in enodelnem omrežju brez obveznih predmetov* razdeli omrežje na dve skupnosti. Rezultat je pričakovan glede na to, da smo obvezne predmete v tej različici omrežja odstranili. Dobimo delitev na visokošolski in magistrski študijski program. Enak rezultat dobimo tudi pri *dvodelnem in enodelnem omrežju z obveznimi predmeti*, kjer parametre pristopa nastavimo tako, da dobimo manjše število skupnosti, kot bi jih modularnost sicer vrnila. Primer delitve omrežja na dve skupnosti je prikazan na Sliki A.1b.

Rezultat izmenjave oznak (glej Poglavlje 3.3) in bločnega modeliranja (glej Poglavlje 3.4) na enodelni različici omrežja z obveznimi predmeti je enak kot pri modularnosti, kjer dobimo dve skupini. Poleg tega nam Pajek omogoča,

da rezultate prikažemo v obliki matrike sosednosti, kot prikazuje Slika 5.3. Blok zgoraj levo predstavlja predmete magistrskega študija, blok spodaj desno pa predmete visokošolskega študija.



Slika 5.3: Bločno modeliranje na enodelnem omrežju vseh študijskih programov z obveznimi predmeti.

5.3.3 Magistrski študijski program

Skupine, dobljene pri analizi magistrskega študijskega programa, ločijo pogosto izbrane predmete od redkeje izbranih. Rezultat je precej bolj informativen kot rezultati, predstavljeni doslej, saj dobimo več informacij o samih predmetih, ki jih študentje izbirajo na magistrskem študiju. S tem bi si študent, ki želi študij nadaljevati na drugi stopnji, lahko pomagal pri izbiri predmetov.

Rezultati bločnega modeliranja na enodelnem omrežju magistrskega študijskega programa z obveznimi predmeti so zelo podobni tistim, kjer obvezne predmete odstranimo. Razlika je ta, da se obvezni predmeti priključijo pogosto izbranim predmetom ter da je nekaj predmetov razvrščenih drugače. Osredotočimo se na *enodelno omrežje magistrskega študijskega programa z obveznimi predmeti*. Rezultat bločnega modeliranja, kjer se omrežje razdeli na dve skupini, prikazuje matrika sosednosti na Sliki 5.4. Omrežje razpade na štiri bloke. Tipi blokov in njihove napake so prikazane v Tabeli 5.3.

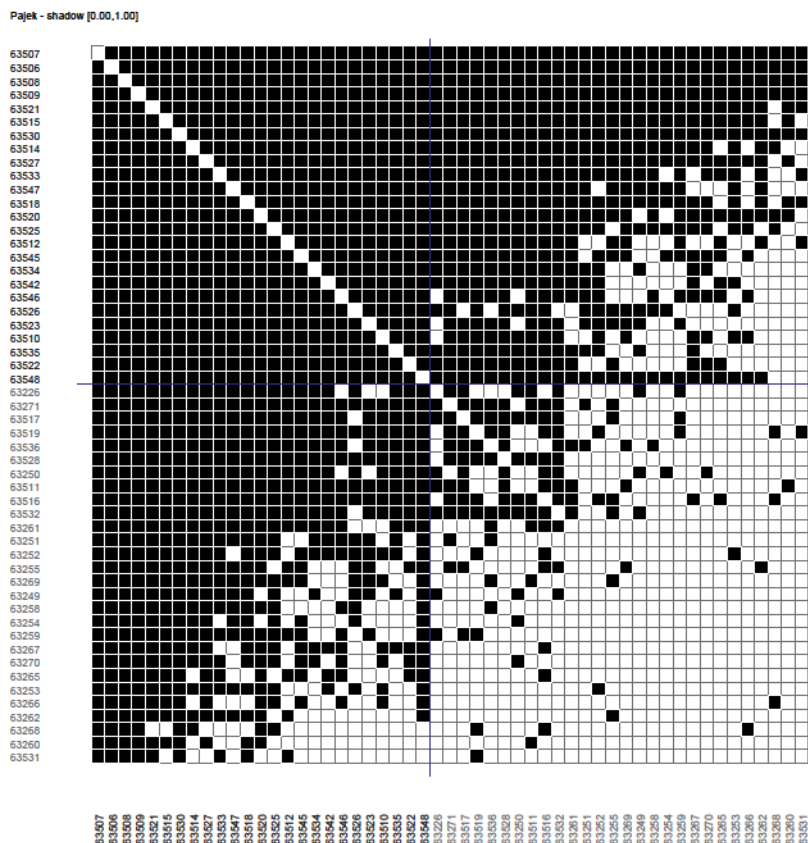
	1	2
1	com	com
2	com	(null)

	1	2
1	0	156
2	156	134

Tabela 5.3: Matrika bločnega modela in matrika napake za enodelno omrežje magistrskega študijskega programa z obveznimi predmeti.

To delitev matrike lahko obravnavamo kot jedrno-obrobno zgradbo. Zgornji levi blok predstavlja jedro, ki je poln blok, in njegova napaka je enaka nič. Je močno povezano in vsebuje predmete, ki so najpogosteje izbrani. Gre za vseh pet obveznih predmetov magistrskega študija; poleg le-teh so še pogosto izbrani predmeti magistrskega študija, katerih šifra se začne s števki 635. Blok spodaj desno predstavlja obrobje, je prazen blok in vsebuje redko izbrane predmete. Ti predmeti spadajo pod univerzitetni študij in njihova šifra se začne s 632. Poleg omenjenih predmetov je v obrobju tudi sedem predmetov magistrskega študijskega programa, ki niso izbrani prav pogosto. Povzamemo lahko, da študentje magistrskega študijskega programa v

večini soizbirajo predmete, ki so jim ponujeni. Redko soizbirajo predmete univerzitetnega študija, pogosteje pa soizbirajo predmete magistrskega in univerzitetnega študija.



Slika 5.4: Bločno modeliranje na enodelnem omrežju magistrskega študijskega programa z obveznimi predmeti.

Slika 5.6a prikazuje porazdelitev predmetov po modulih za rezultat, ki ga vrne bločno modeliranje pri delitvi omrežja na dve skupini. Moder del predstavlja tako imenovano jedro in vidimo, da se tukaj pojavijo samo moduli magistrskega študijskega programa in obvezni predmeti. Rumeno obarvani stolpci predstavljajo obrobje, kjer so vsi predmeti univerzitetnega študija. Če se spomnimo, smo predmete univerzitetnega študijskega programa razdelili na računalništvo in informatiko, magistrski pa se delijo na module. Hkrati je

v obrobju tudi nekaj predmetov magistrskega študija, kar nakazujejo rumeni stolpci nad modrimi. Interpretacija rezultatov je podobna kot v primeru predstavitve z matriko sosednosti.

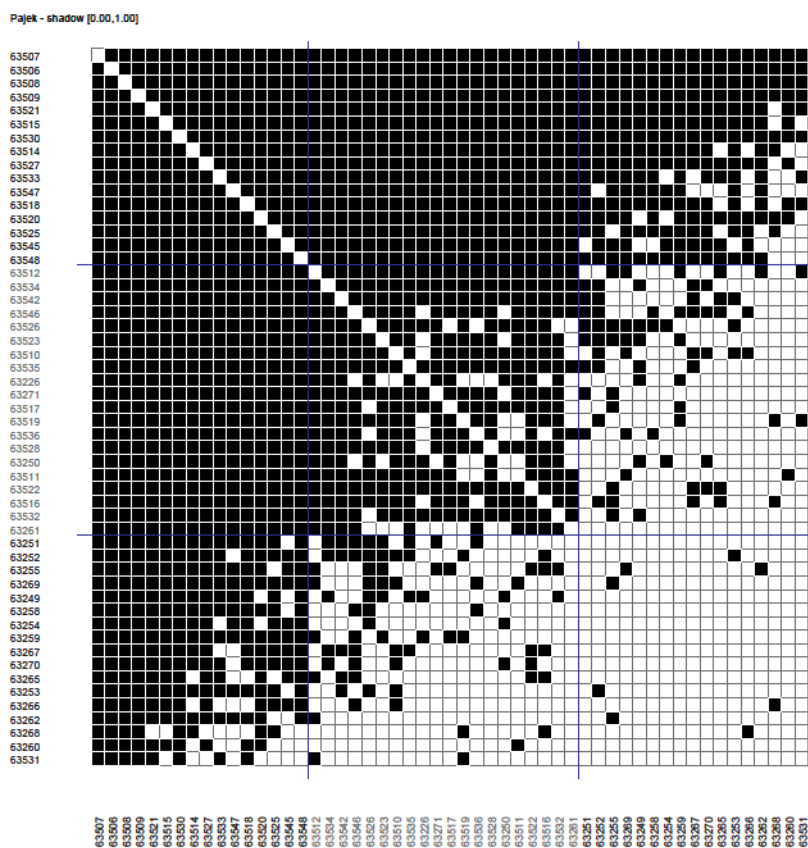
Matrika sosednosti, dobljena z bločnim modeliranjem, kjer se omrežje razdeli na tri skupine, je prikazana na Sliki 5.5. Razpade na devet blokov, katerih tipi in napake so podane v Tabeli 5.4. Štiri polne bloke v zgornjem levem delu lahko skupaj obravnavamo kot jedro. Zgornji levi blok vsebuje vseh pet obveznih predmetov in enajst najpogosteje izbranih predmetov magistrskega študija. Sredinski diagonalni blok v večini vsebuje predmete magistrskega študijskega programa in nekaj univerzitetnih. Izbranost teh predmetov je nekoliko nižja kot prej. Predmeti obeh blokov so pogosto soizbrani tako znotraj obeh skupin kot tudi med sabo. Spodnji desni blok predstavlja obrobje redko izbranih predmetov. Ti predmeti so soizbrani zelo poredko, se pa pogosteje pojavijo predvsem v paru z najpogosteje izbranimi predmeti.

	1	2	3
1	com	com	com
2	com	com	(null)
3	com	(null)	(null)

	1	2	3
1	0	0	44
2	0	56	77
3	44	77	8

Tabela 5.4: Matrika bločnega modela in matrika napake za enodelno omrežje magistrskega študijskega programa z obveznimi predmeti.

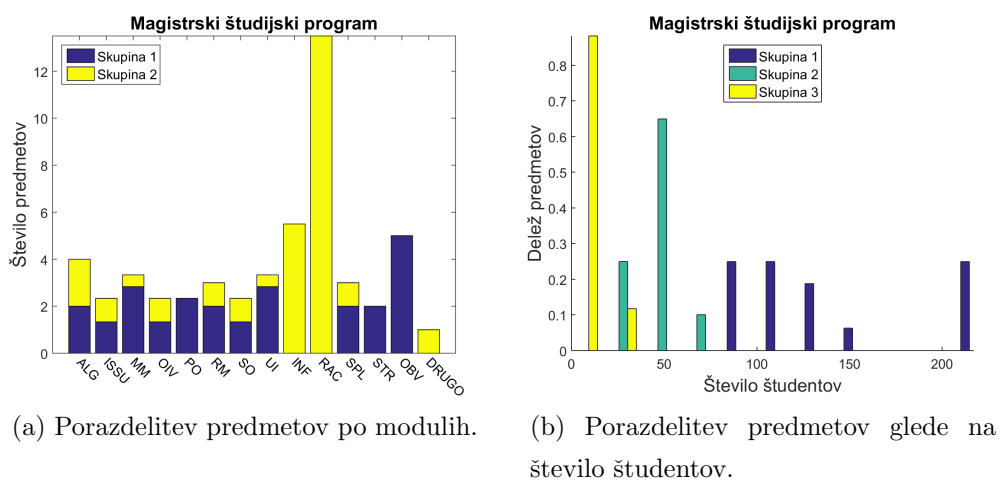
Slika 5.6b prikazuje porazdelitev predmetov glede na število študentov za primer, ko bločno modeliranje omrežje razdeli na tri skupine. Jasno so prikazane ravnokar predstavljene tri skupine. Modri stolpci ustrezajo jedru, rumeni obrobju, zeleni pa sredinskemu diagonalnemu bloku matrike sosednosti, ki ga lahko štejem kot del jedra. Do manjših prekrivanj pride pri rumeni in zeleni skupini. Razen tega se omrežje lepo loči na tri skupine glede na to, kako pogosto so bili predmeti izbrani.



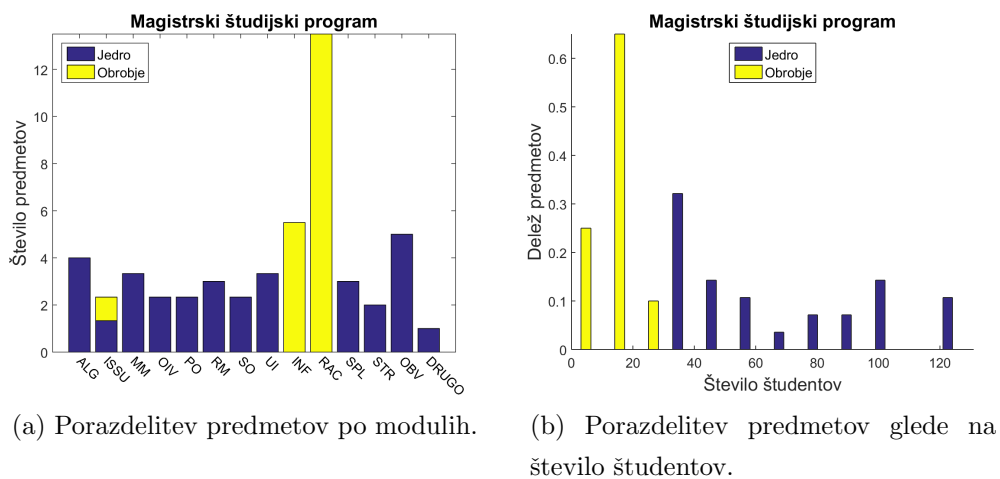
Slika 5.5: Bločno modeliranje na enodelnem omrežju magistrskega študijskega programa z obveznimi predmeti.

Jedrno-obrobna zgradba (glej Poglavlje 3.1) na omrežju magistrskega študijskega programa vrne lepše rezultate na enodelnih različicah omrežja, kot na dvodelnih. Jedro se lepše loči od periferije, kar pomeni, da so v jedru res pogosto izbrani predmeti ter v obrobju tisti, ki so redko izbrani. Posledično tudi ne prihaja do prekrivanj oziroma je le-teh malo. Poglejmo porazdelitev predmetov po modulih, ki jo vrne jedro-obrobna zgradba na Sliki 5.7a. Če jo primerjamo s porazdelitvijo, dobljeno pri bločnem modeliranju, vidimo jasno ločitev na predmete magistrskega in univerzitetnega študija. Le en predmet magistrskega študija 63531 Sodobni pristopi in arhitekture pri razvoju informacijskih sistemov, ki spada v modul ISSU, je razporejen v drugo skupino.

Ta predmet so izbrali le štirje študentje, zato je jasno, zakaj je bil dodeljen obrobju. Na Sliki 5.7b, ki jo kot rezultat vrne jedrno-obrobna zgradba na enodelnem omrežju magistrskega študijskega programa brez obveznih predmetov, se lepo loči jedro s pogosto izbranimi predmeti od obrobja, ki je redkeje izbrano.



Slika 5.6: Bločno modeliranje.



Slika 5.7: Jedrno-obrobna zgradba.

5.3.4 1. letnik magistrskega študijskega programa

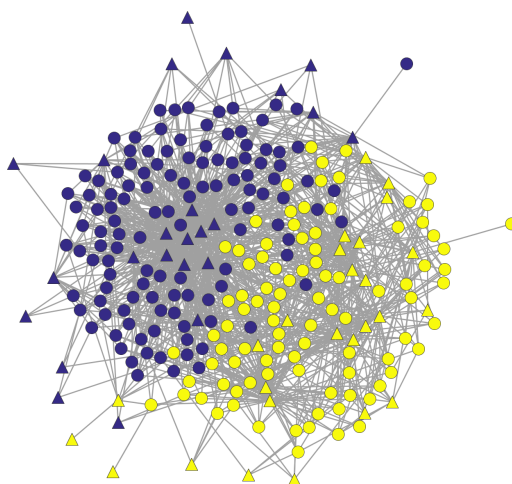
Algoritem izmenjave oznak na obravnavanih omrežjih ne dela najboljše. Rezultate, primerne za nadaljnjo analizo, dobimo samo pri dvodelnih omrežjih brez obveznih predmetov. V tem primeru omrežja razpadejo na dve ali več skupnosti, v ostalih primerih pa so vsa vozlišča dodeljena eni sami skupnosti. V nadaljevanju podrobneje analiziramo rezultate, dobljene s pristopom izmenjave oznak na *dvodelnem omrežju 1. letnika magistrskega študijskega programa brez obveznih predmetov*.

Graf na Sliki 5.8a prikazuje vizualizacijo tega omrežja, pridobljeno z orodjem Visone. Omrežje razpade na dve skupnosti, ki sta predstavljeni z različnima barvama. Vozlišča v obliki trikotnika predstavljajo predmete, vozlišča v obliki kroga pa študente. Ne moremo reči, da so vozlišča znotraj posamezne skupnosti močno povezana in redko povezana z vozlišči druge skupnosti. Če skupnosti ne bi bili različno obarvani, iz same strukture grafa ne bi bilo mogoče razbrati delitve na skupnosti.

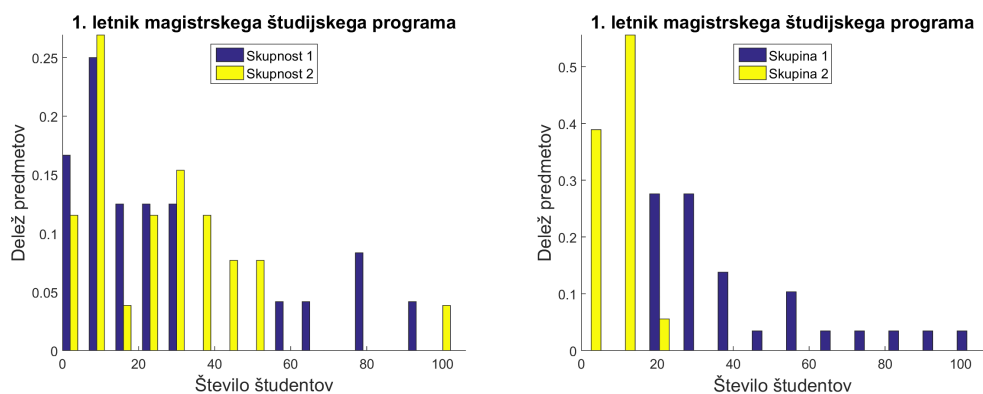
V osrednjem delu modre skupnosti so predmeti, ki so pogosteje izbrani, kar se vidi zaradi večjega števila povezav v tem delu grafa. Hkrati pa so del iste skupnosti tudi predmeti z obrobja grafa. Podobno velja tudi za drugo skupnost. Že s same slike tako opazimo, da pristop izmenjave oznak ne išče skupnosti glede na to, kako pogosto so predmeti izbrani, kot pristopi, ki smo jih uporabili do sedaj. To opažanje lahko potrdimo z izrisom porazdelitve predmetov glede na število študentov, ki jo prikazuje Slika 5.8b. Očitno je, da ni jasnih razlik med skupinama. Vsaka od skupnosti vsebuje tako predmete, ki so pogosto izbrani, kot tudi redko izbrane predmete. Dobljeno porazdelitev primerjajmo s porazdelitvijo, ki jo bločno modeliranje vrne za enodelno omrežje in je prikazana na Sliki 5.8c. Bločno modeliranje omrežje razdeli na močno povezano jedro in redko povezano obrobje, kot smo predvidevali na podlagi prejšnjih primerov. S tem vidimo, da izmenjava oznak v primerjavi z drugimi pristopi združuje vozlišča na povsem drugačen način.

Slika 5.9a prikazuje porazdelitev predmetov po laboratorijih, dobljeno s pristopom izmenjave oznak, kjer se omrežje razdeli na dve skupnosti. Večina

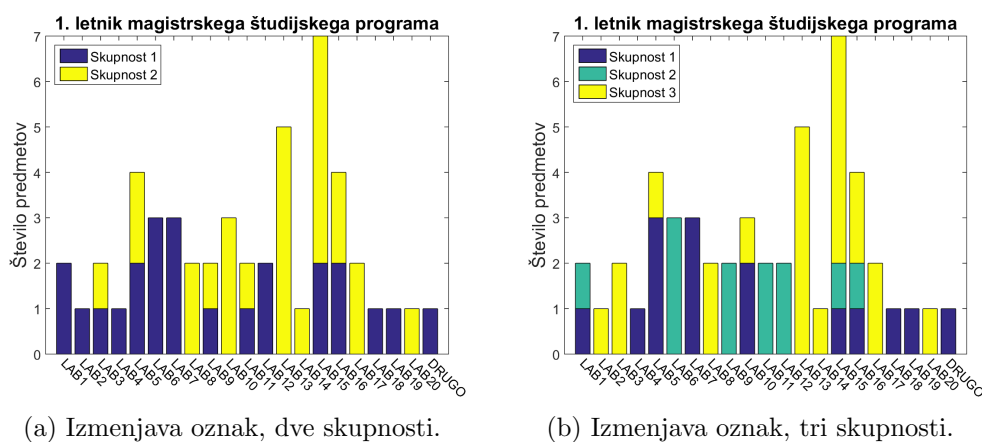
laboratorijev je prirejena eni od skupnosti. V prvo, modro skupnost, je dodeljenih osem laboratorijev in kategorija drugo, v rumeno skupnost pa je dodeljenih šest laboratorijev. Preostalih šest laboratorijev je razdeljenih med obe skupnosti. Porazdelitev predmetov po laboratorijih, kjer za enako omrežje algoritem izmenjave oznak vrne 3 skupnosti, prikazuje slika 5.9b. Rezultat je podoben kot v primeru dveh skupnosti. Večina laboratorijev je dodeljena eni od treh skupnosti, pet pa se jih deli med dve ali tri skupnosti.



(a) Graf omrežja, dobljen z izmenjavo oznak.



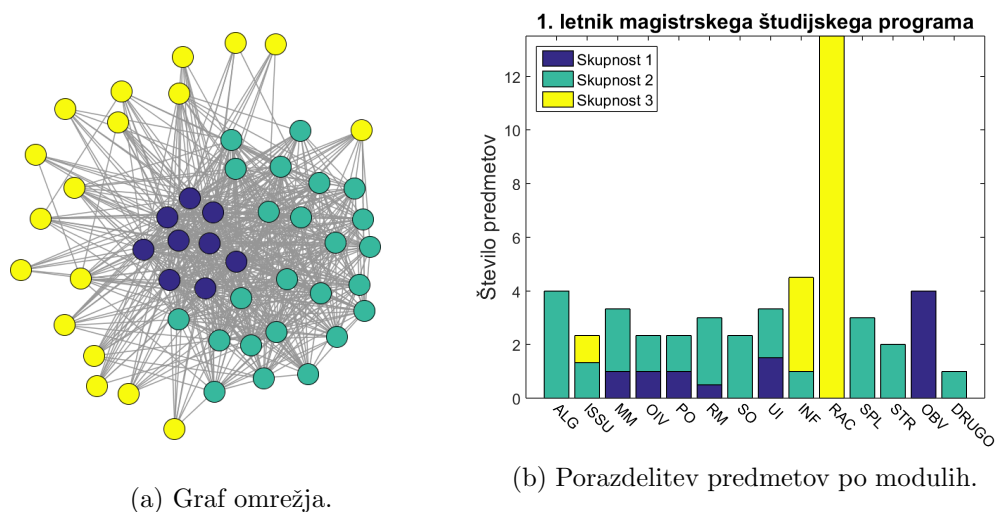
Slika 5.8: Omrežje 1. letnika magistrskega študijskega programa brez obveznih predmetov.



Slika 5.9: Dvodavno omrežje 1. letnika magistrskega študijskega programa brez obveznih predmetov.

Stohastično bločno modeliranje (glej Poglavje 3.5), ki ga sicer izvedemo na *enodelnem omrežju 1. letnika magistrskega študijskega programa z obveznimi predmeti*, vrne drugačne rezultate kot izmenjava oznak. Razlika je vidna že z grafa, ki je prikazan na Sliki 5.10a in je pridobljen z orodjem Visone. Poleg tega, da ima omrežje manj vozlišč, vidimo, da so tudi skupnosti razvrščene povsem drugače. Osrednji, modri, del predstavlja močno povezana vozlišča, zeleni del redkeje povezana vozlišča in rumeni del redko povezana vozlišča. Za modri in zeleni del lahko rečemo, da skupaj predstavljata jedro, rumeni del pa ustreza obrobju.

Porazdelitev predmetov po modulih je prikazana na Sliki 5.10b. Če na hitro pogledamo, vidimo, da omrežje razpade na tri skupnosti. Na obvezne predmete, ki na grafu predstavljajo osrednji del, na predmete magistrskega študija, ki so predstavljeni z zeleno, ter rumeno obarvano obrobje. Seveda so prisotna tudi manjša odstopanja. Tabela A.4 prikazuje, kateri skupnosti je dodeljen posamezen predmet in je urejena glede na izbranost predmetov.



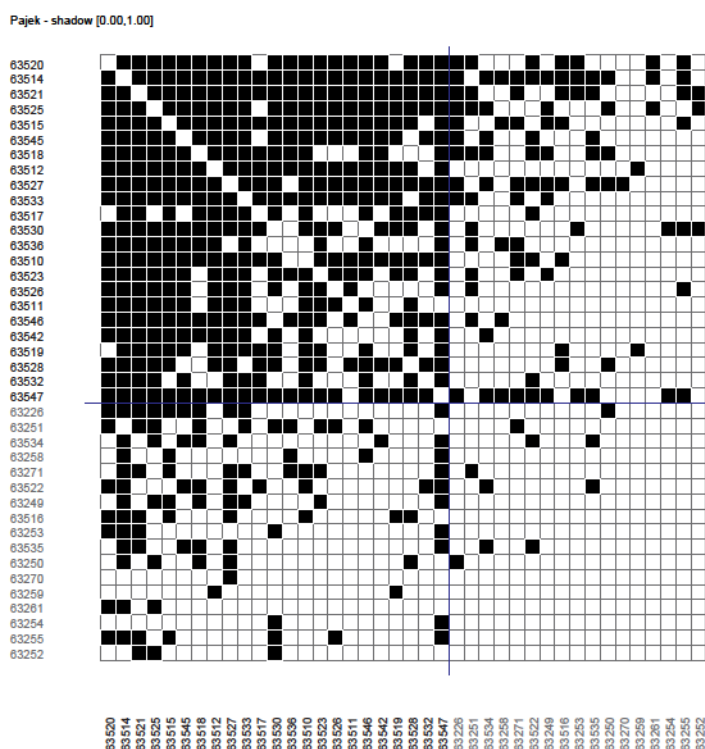
Slika 5.10: Stohastično bločno modeliranje.

5.3.5 2. letnik magistrskega študijskega programa

Poglejmo, kaj z bločnim modeliranjem dobimo na enodelnem omrežju 2. letnika magistrskega študijskega programa. Razlike med rezultati omrežij z obveznimi in brez obveznih predmetov so minimalne. Obvezni predmet 2. letnika se priključi pogosto izbranim predmetom, obvezni predmeti prejšnjega letnika pa redko izbranim. Osredotočimo se na *enodelno omrežje 2. letnika magistrskega študijskega programa brez obveznih predmetov*.

Slika 5.11 prikazuje dobljeno matriko sosednosti. Tipi blokov in njihove napake so prikazane v Tabeli 5.5. Omrežje razpade na dve skupini. V tem primeru ne gre za pravo jedrno-obrobno zgradbo, saj sta poln in prazen diagonalni blok med sabo redko povezana. To nakazujeta prazna bloka zgoraj desno in spodaj levo. V polnem bloku so najpogosteje izbrani predmeti magistrskega študijskega programa, katerih šifra se začne s 635. Spodnji desni blok predstavljajo redko izbrani predmeti. Gre za predmete univerzitetnega študijskega programa, katerih šifra se prične s 632. Poleg že omenjenih predmetov so del te skupine tudi štirje predmeti magistrskega študija, ki so jih

študentje izbrali manj kot 9-krat. Poln blok nakazuje, da študentje 2. letnika magistrskega študijskega programa v veliki meri soizbirajo predmete magistrskega študija. Predmeti univerzitetnega študija so soizbrani zelo redko. Tudi predmeti magistrskega in univerzitetnega študija niso pogosto soizbrani. To sporočajo trije prazni bloki matrike sosednosti.

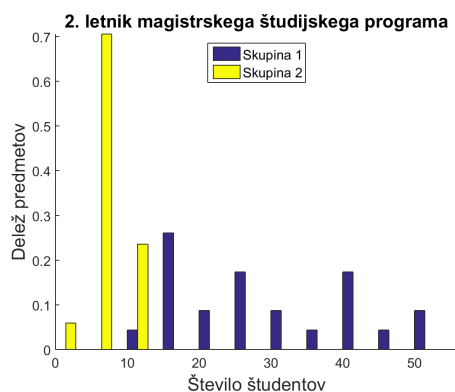


Slika 5.11: Bločno modeliranje na enodelnem omrežju 2. letnika magistrskega študijskega programa brez obveznih predmetov.

	1	2		1	2
1	com	(null)	1	92	100
2	(null)	(null)	2	100	10

Tabela 5.5: Matrika bločnega modela in matrika napake za enodelno omrežje 2. letnika magistrskega študijskega programa brez obveznih predmetov.

Slika 5.12 prikazuje porazdelitev predmetov glede na število študentov in potrjuje zgoraj opisana dejstva. Skupini se med sabo ločita glede na to, kako pogosto so predmeti izbrani. Modra porazdelitev ustreza polnemu bloku matrike sosednosti, rumena pa praznemu bloku spodaj desno. Do manjših odstopanj pride nekje med deset in petnajst študenti. Gre za predmet 63532 Diskretna matematika, ki je bil izbran 10-krat ter je dodeljen prvi skupini, in predmet 63271 Osnove oblikovanja, ki je bil izbran 12-krat ter je dodeljen drugi skupini. Tukaj nekje je meja, ki ločuje skupini in v tem primeru ni točno določena.



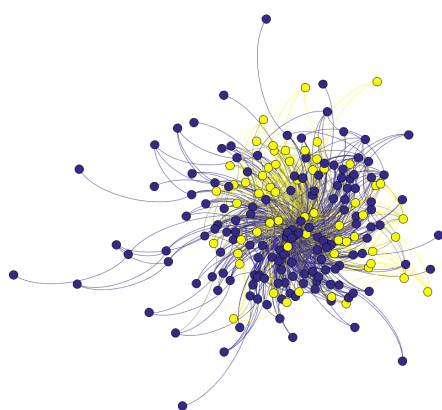
Slika 5.12: Porazdelitev predmetov glede na število študentov, dobljena z bločnim modeliranjem.

Pri optimizaciji modularnosti za večino obravnavanih omrežij iz same strukture omrežja ni mogoče razbrati delitve na skupnosti; ne glede na to, katerega od algoritmov prikaza uporabimo v Gephiju. Tudi po obarvanju posameznih skupnosti, ni mogoče opaziti jasne ločitve v omrežju. Le pri analizi celotnega omrežja z začetka poglavja se tako strukturno kot tudi barvno študijski programi lepo ločijo.

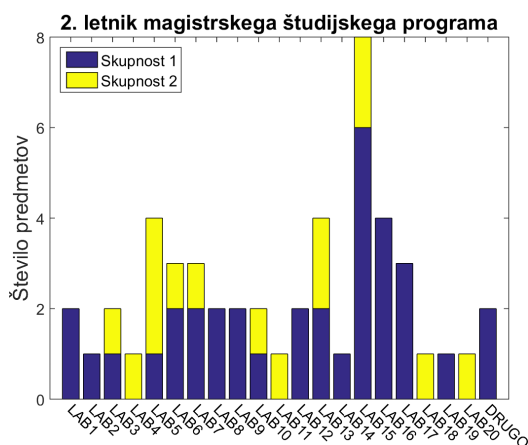
Na Sliki 5.13a je vizualni prikaz, ki ga optimizacija modularnosti vrne za *dvodelno omrežje 2. letnika magistrskega študijskega programa z obveznimi predmeti*. Slika potrjuje zgoraj predstavljena dejstva, da skupnosti med sabo niso jasno ločene. Sklepati kaj več o tem, na kakšen način so predmeti do-

deljeni skupnostim, s same slike ni mogoče. Slika A.2a v dodatku prikazuje graf, kjer različni barvi označujeta predmete in študente. Tabela A.5 za vsak predmet pove, kateri skupnosti je le-ta dodeljen. Urejena je glede na število študentov, ki je izbralo posamezen predmet. Lahko opazimo, da ni vzorca, ki bi glede na izbranost predmetov ločeval skupnosti. To je povsem razumljivo, saj optimizacija modularnosti išče razbitje na skupnosti, kjer je modularnost največja, in ne išče skupnosti glede na to, kako pogosto je bil predmet izbran.

Porazdelitev predmetov po laboratorijih, ki jih dobimo z optimizacijo modularnosti, prikazuje Slika 5.13b. Na prvi pogled je vidno, da je večji del predmetov dodeljen prvi, modri skupnosti. Devet laboratorijev in kategorija drugo so v celoti dodeljeni modri skupnosti, štirje laboratoriji pa so dodeljeni rumeni. Ostalih sedem laboratorijev je razdeljenih med obe skupnosti.



(a) Graf omrežja.



(b) Porazdelitev predmetov po laboratorijih.

Slika 5.13: Optimizacija modularnosti.

5.3.6 Visokošolski študijski program

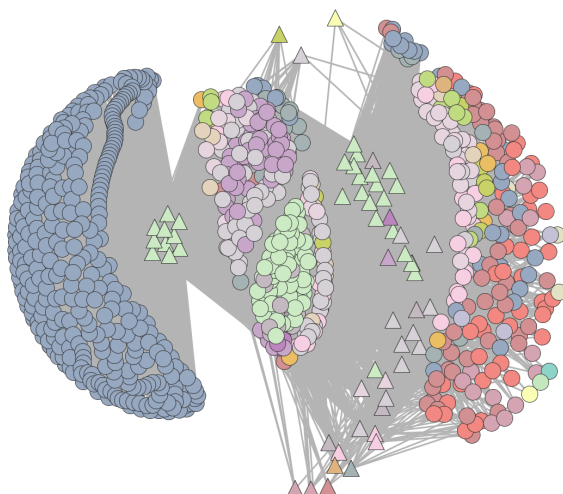
Poglejmo rezultat, ki ga vrne jedrno-obrobna zgradba za *dvodelno omrežje visokošolskega študijskega programa z obveznimi predmeti*. Vizualni prikaz omrežja, pridobljen z orodjem Visone, prikazuje Slika 5.14a. Vozlišča v obliki trikotnikov predstavljajo predmete, vozlišča v obliki krogov pa študente. Povsem razumljivo je, da je študentov veliko več, saj imajo vsi študentje na voljo enako množico predmetov, med katerimi lahko izbirajo. Z barvami so med sabo ločena k -jedra, kjer se k giblje od 4 do 23, vendar ne zavzame vseh vmesnih vrednosti. Jedrno število grafa je najmanjši k , tako da je $(k + 1)$ -jedro prazno. Ustrezno k -jedro se imenuje jedro grafa. V tem primeru jedro grafa predstavlja 23-jedro.

Iz strukture grafa vidimo, da omrežje sestavlja več delov. Skrajno levi, modro obarvan del, predstavlja študente, vpisane v 1. letnik visokošolskega študijskega programa. Ti študentje so povezani z vsemi obveznimi predmeti 1. letnika, ki jih prepoznamo kot skupino desetih zelenih trikotnikov. Sledi množica študentov 2. letnika visokošolskega študija in nato predmeti, ki jih študentje lahko izberejo tako v 2. kot tudi v 3. letniku. Skrajno desni del grafa v večini predstavlja študente 3. letnika. Je pa v notranjem delu tega območja še nekaj študentov 2. letnika.

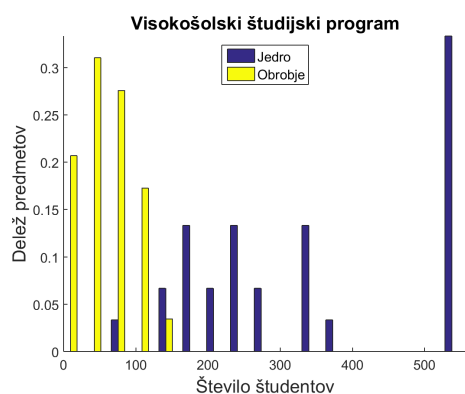
Jedro grafa predstavljajo zeleno obarvana vozlišča oziroma, ker nas zanimajo samo predmeti, zeleno obarvani trikotniki. V jedru so obvezni predmeti 1. letnika, saj imajo največjo stopnjo izbranosti. Poleg tega so v jedru tudi pogosto izbrani predmeti 2. in 3. letnika visokošolskega študijskega programa. Preostala vozlišča v obliki trikotnika obravnavamo kot obrobje.

Slika 5.14b prikazuje porazdelitev predmetov glede na število študentov. Lahko opazimo, da se jedro in obrobje ločita glede na to, kako pogosto so bili predmeti izbrani. Ker pride do manjših odstopanj, meja med jedrom in obrobjem ni točno določena. Predmet 63749 Izbrana poglavja iz računalništva in informatike, ki je bil izbran 63-krat, je dodeljen jedru, predmet 63744 Digitalno procesiranje signalov, izbran 136-krat, pa je dodeljen obrobju.

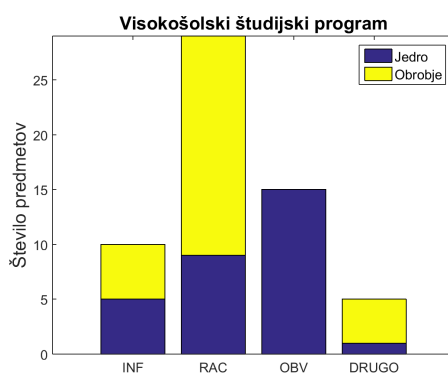
Porazdelitev predmetov po modulih prikazuje Slika 5.14c. Predmeti so enakomerno razdeljeni med jedro in obrobje. Opazimo, da je vseh petnajst obveznih predmetov visokošolskega študijskega programa dodeljenih jedru. To je pričakovano, saj gre za najpogosteje izbrane predmete. Predmeti informatike se enakomerno delijo med obe skupini, pri računalništvu in drugih predmetih pa je večji delež dodeljen obrobju. S slike je razvidno, da je na visokoškolskem študijskem programu veliko več računalniških predmetov kot predmetov informatike.



(a) Graf omrežja.



(b) Porazdelitev predmetov glede na število študentov.



(c) Porazdelitev predmetov po modulih.

Slika 5.14: Jedrno-obrobna zgradba.

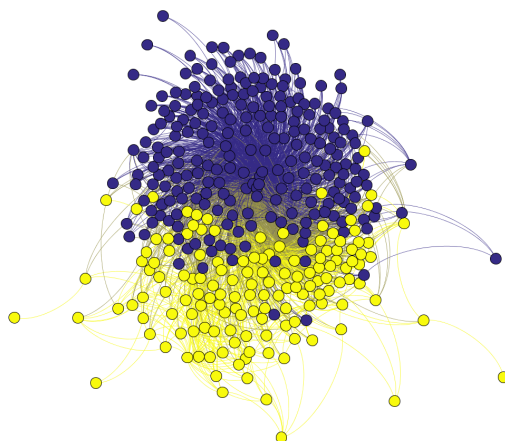
5.3.7 2. letnik visokošolskega študijskega programa

V enem izmed prejšnjih rezultatov smo omenili, da se iz večine vizualizacij, pridobljenih z optimizacijo modularnosti, delitve omrežja na skupnosti ne da razbrati. Tako s stališča strukture omrežja kot tudi po obarvanju posameznih skupnosti se le-te med sabo ne ločijo.

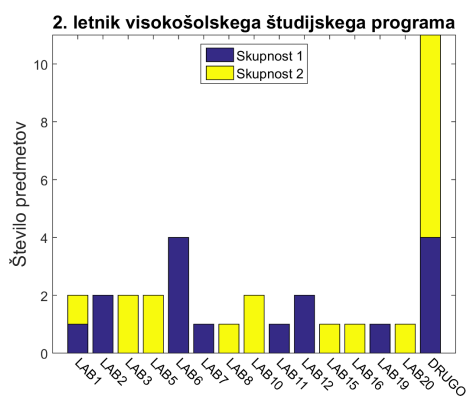
Eden izmed redkih primerov optimizacije modularnosti, ki temu nasprotuje, je rezultat, dobljen na *dvodelnem omrežju 2. letnika visokošolskega študijskega programa brez obveznih predmetov*. Graf, kjer se vozlišča po obarvanju skupnosti razdelijo na dva dela, je prikazan na Sliki 5.15a. V osrednjem delu modre skupnosti je večji del najbolj pogosto izbranih predmetov, kar vidimo po gosti povezani sredini. To potrjuje Tabela A.6, ki za vsak predmet podaja skupnost, kateri je le-ta dodeljen in je urejena glede na število študentov, ki so izbrali posamezen predmet. Tudi rumena skupnost vsebuje nekaj najpogosteje izbranih predmetov. Preostali, redkeje izbrani predmeti, se delijo med obe skupnosti. Optimizacija modularnosti ne išče skupnosti glede na to, kako pogosto so bili predmeti izbrani, ampak išče razbitje z največjo modularnostjo. Zato se skupnosti med sabo ne ločita glede na stopnjo izbranosti. Slika A.2b v dodatku prikazuje graf, kjer različni barvi označujeta predmete in študente.

Porazdelitev po laboratorijih, ki jo za to omrežje dobimo z optimizacijo modularnosti, prikazuje Slika 5.15b. Je ena izmed najbolj čistih porazdelitev, ki jih dobimo z obravnavanimi omrežji. Predmeti so dokaj enakomerno porazdeljeni med obe skupnosti in vsi laboratoriji razen dveh so v celoti dodeljeni eni sami skupnosti. V prvo skupnost spada šest laboratorijev, v drugo pa sedem. Le en laboratorij in drugi predmeti se delijo med obe skupnosti.

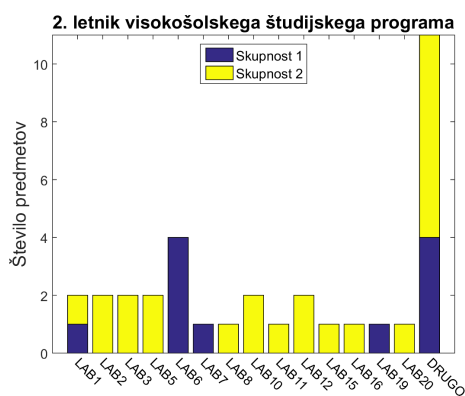
Podobno porazdelitev, kjer je velika večina laboratorijev dodeljena eni sami skupnosti, vrne tudi izmenjava oznak za enako omrežje. Porazdelitev je prikazana na Sliki 5.15c. V tem primeru je veliko večji delež predmetov dodeljen drugi skupnosti. V prvo skupnost spadajo trije laboratoriji, deset pa jih je dodeljenih drugi skupnosti. En laboratorij ter drugi predmeti so podobno kot pri optimizaciji modularnosti dodeljeni obema skupnostma.



(a) Graf omrežja, dobljen z optimizacijo modularnosti.



(b) Optimizacija modularnosti.



(c) Izmenjava oznak.

Slika 5.15: Dvodavno omrežje 2. letnika visokošolskega študijskega programa brez obveznih predmetov.

5.3.8 3. letnik visokošolskega študijskega programa

Nazadnje si oglejmo še analizo 3. letnika visokošolskega študijskega programa. Matrika sosednosti, dobljena z bločnim modeliranjem na *enodelnem omrežju* 3. letnika visokošolskega študijskega programa brez obveznih predmetov, je prikazana na Sliki 5.16. Tipe posameznih blokov in njihove napake prika-

zuje Tabela 5.6. Dobimo razbitje omrežja na tri skupine. Polne bloke skupaj s sredinskim diagonalnim blokom lahko obravnavamo kot jedrno-obrobno zgradbo. Zgornji levi blok ustreza jedru, je poln blok in predmeti tega bloka so pogosto soizbrani. Sredinski diagonalni blok predstavlja obrobje, je prazen blok in predmeti tega bloka so redko soizbrani. Tudi predmeti jedra in obrobja so pogosto soizbrani, kar predstavljata preostala polna bloka. Tretjo skupino predstavlja blok spodaj desno, katerega predmeti se med seboj skoraj ne izbirajo. Tudi s tako imenovanim jedrom in obrobjem so ti predmeti zelo redko soizbrani.

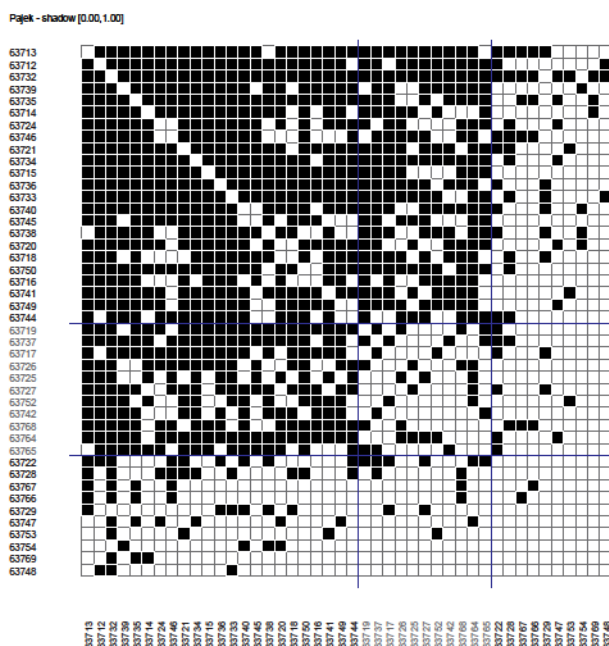
Tabela A.7 za vsak predmet poda skupino, kateri je le-ta dodeljen s pristopom bločnega modeliranja. Urejena je glede na to, kako pogosto so bili predmeti izbrani. Diagonalni blok spodaj desno predstavlja deset najredkeje izbranih predmetov. Meja, ki ločuje to skupino od ostalih dveh, je očitna. Petnajst najpogosteje izbranih predmetov pripada jedru, preostali redkeje izbrani predmeti pa se delijo med jedro in obrobje. Meja med jedrom in obrobjem, glede na število študentov, ki je predmete izbiralo, ni jasna. Kljub temu je to rezultat z najmanjšo napako, ki ga bločno modeliranje vrne za delitev tega omrežja na tri skupine.

	1	2	3
1	com	com	(null)
2	com	(null)	(null)
3	(null)	(null)	(null)

	1	2	3
1	68	61	52
2	61	32	14
3	52	14	2

Tabela 5.6: Matrika bločnega modela in matrika napake za enodelno omrežje 3. letnika visokošolskega študijskega programa brez obveznih predmetov.

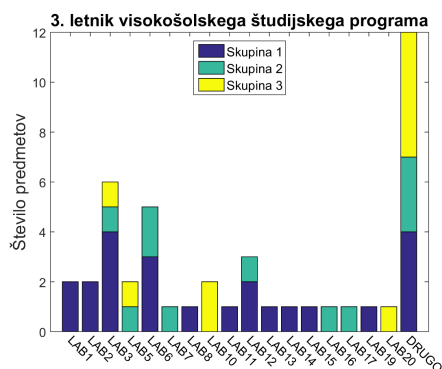
Porazdelitev predmetov po laboratorijih je prikazana s Sliko 5.17. V prvo skupino, jedro, je določenih osem laboratorijev. Obrobju so dodeljeni trije laboratoriji, tretji skupini pa pripadata dva. Preostali štirje laboratoriji in drugi predmeti so prirejeni dvema ali trem skupinam. Največ predmetov izvajajo ravno ti laboratoriji, tako da je delitev med več skupin pričakovana.



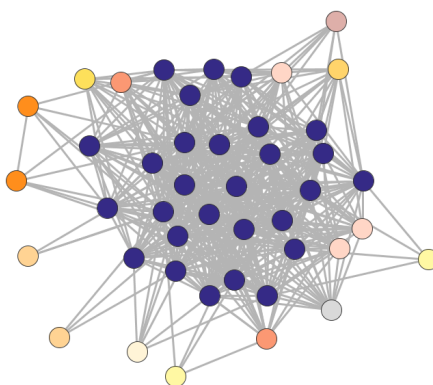
Slika 5.16: Bločno modeliranje na enodelnem omrežju 3. letnika visokošolskega študijskega programa brez obveznih predmetov.

Poglejmo še rezultat, ki ga za enodelno omrežje 3. letnika visokošolskega študijskega programa brez obveznih predmetov vrne jedrno-obrobna zgradba. Vizualizacija iz Visone je prikazana na Sliki 5.18a. Z barvami so med sabo ločena k -jedra. Jedro grafa je v tem primeru 19-jedro, čemur ustrezajo modra vozlišča. Preostali del grafa je dodeljen obrobju. Jedro je močno povezano med seboj, redkeje povezano z vozlišči v obrobju, obrobna vozlišča pa se med sabo povezujejo zelo redko. To je ravno definicija jedrno-obrobne zgradbe.

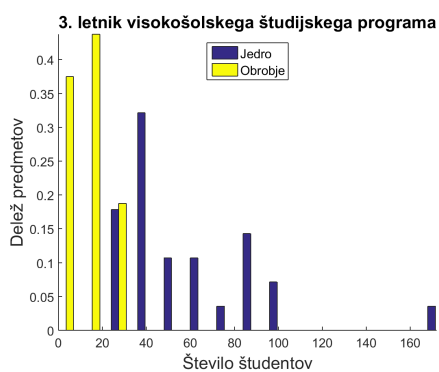
Porazdelitev predmetov po številu študentov, ki jo dobimo za dani rezultat, je prikazana s Sliko 5.18b. Pri meji med jedrom in obrobjem pride do manjših odstopanj, kjer sta dva predmeta, ki ju je izbralo devetnajst študentov, določena obrobju, predmet, ki ga je izbralo enako število študentov, pa je določen jedru. Slika 5.18c prikazuje porazdelitev predmetov po laboratorijih. Jedru je dodeljenih osem laboratorijev, obrobju pa šest. Ostali trije laboratoriji in drugi predmeti so dodeljeni tako jedru kot tudi obrobju.



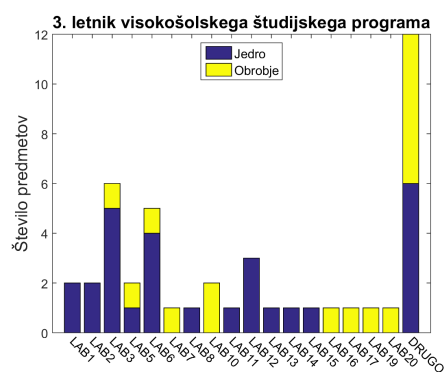
Slika 5.17: Bločno modeliranje.



(a) Graf omrežja.



(b) Porazdelitev predmetov glede na število študentov.



(c) Porazdelitev predmetov po laboratorijih.

Slika 5.18: Jedrno-obrobna zgradba.

5.4 Sklepne ugotovitve

Rezultati analize so odvisni od pristopa, s katerim želimo poiskati skupine, in od omrežja, na katerem izvajamo pristop. Na primer pri analizi 1. letnika visokošolskega študijskega programa in celotnega omrežja vseh pet pristopov vrne enak rezultat. V primeru 1. letnika visokošolskega študijskega programa dobimo eno samo skupino, pri analizi celotnega omrežja pa se omrežje loči glede na študijska programa.

Jedrno-obrobna zgradba v splošnem vrne lepo delitev na dva razreda, in sicer na gosto povezano jedro ter s tem na pogosto izbrane predmete in redko povezano obrobje oziroma redko izbrane predmete. Bločno modeliranje lahko obravnavamo kot jedrno-obrobno zgradbo. Vedno imamo vsaj en poln diagonalen blok, ki ustreza jedru in vsebuje predmete, ki so pogosto soizbrani. Poleg tega je en prazen diagonalen blok v spodnjem desnem kotu matrike sosednosti, ki ustreza obrobju in predstavlja redko izbrane predmete. Preostali bloki so lahko polni, le-ti predstavljajo povezave med jedrom in obrobjem ali prazni, kadar povezav med jedrom in obrobjem skoraj ni. Stohastično bločno modeliranje prav tako vrača rezultate, kjer so skupnosti med sabo ločene glede na to, kako pogosto so predmeti izbrani.

Algoritem izmenjave oznak in optimizacija modularnosti ne iščeta skupnosti glede na to, kako pogosto so predmeti izbrani, kar je opazno iz dobljenih rezultatov. Algoritem izmenjave oznak vrne rezultate z dvema ali več skupnostma le pri dvodelnih omrežjih brez obveznih predmetov. Pri optimizaciji modularnosti iz same strukture omrežja v večini primerov ni mogoče razbrati delitve na skupnosti. Redka primera, ki temu nasprotujeta, sta omrežje vseh študijskih programov in omrežje 2. letnika visokošolskega študijskega programa. Izmed vseh analiz smo prikazali in interpretirali porazdelitve po laboratorijih ter modulih, ki so najbolj čiste. S tem mislimo, da je čim manj laboratorijev oziroma modulov dodeljenih večim skupinam. V večini preostalih primerov so posamezni laboratoriji ter moduli dodeljeni večim skupinam, zato dobljeni rezultati za potrebe diplomskega dela niso dovolj reprezentativni.

Poglavje 6

Sklep in zaključek

Namen diplomskega dela je bil za omrežje visokošolskega in magistrskega študijskega programa Fakultete za računalništvo in informatiko poiskati skupine predmetov glede na to, kako jih študentje soizbirajo. Po ločitvi omrežja na študijske programe in letnike, smo na dobljenih omrežjih izvedli predstavljene pristope, namenjene iskanju skupin v omrežju. Zanimalo nas je, kako se predmeti posameznih skupin razporedijo glede na laboratorije in module.

Idealno bi bilo, da bi kot rezultat porazdelitev dobili čiste skupine. Se pravi, da bi bili predmeti posameznega laboratorija oziroma modula dodeljeni le eni od skupin. V večini primerov smo prišli do ravno nasprotnega rezultata. Hipotezo, da se predmeti po skupinah lepo ločijo glede na laboratorije ali module, smo zavrnil. V nekaterih primerih pa se rezultati dobro približajo želenim in le-te v diplomskem delu interpretiramo. Težko je reči, kateri od pristopov je dober in kateri ne, saj ima vsak svoj pogled na skupine v omrežju. Prav tako je rezultat odvisen tudi od omrežja, na katerem izvajamo pristope. Na primer analiza celotnega omrežja vrne ločitev glede na študijske programe. Pri preostalih omrežjih, ki smo jih obravnavali, se obvezni predmeti priključijo pogosto izbranim. Z rezultati tega diplomskega dela bi si študentje lahko pomagali pri izbiri predmetov pred vpisom v novo študijsko leto. Hkrati lahko služijo kot dodatne informacije Fakulteti za računalništvo in informatiko o izbiri predmetov, ki jih ponujajo.

V nadaljevanju podamo nekaj idej za izboljšavo in nadgradnjo v diplomskem delu predstavljenih analiz. Delali smo s podatki med leti 2012 in 2015. V primeru, da bi vzeli podatke za daljše časovno obdobje, bi bili s tem tudi rezultati bolj reprezentativni. Dobro bi bilo premisliti tudi o morebitni spremembi pragov. V našem primeru smo iz omrežja odstranili vse pare predmetov, ki jih je soizbral le en študent. Nastavljanje pragov ni enostavno, saj navadno ni jasno, kaj optimizirati, in se lahko razlikujejo od omrežja do omrežja. V diplomskem delu smo uporabili le eno vrsto metod za gručenje. Morda bi z drugačnimi pristopi dobili boljše rezultate. Poleg tega bi lahko analizo z enakimi pristopi izvedli tudi za podatke drugih študijskih programov Fakultete za računalništvo in informatiko ali pa za študijske programe drugih fakultet.

Literatura

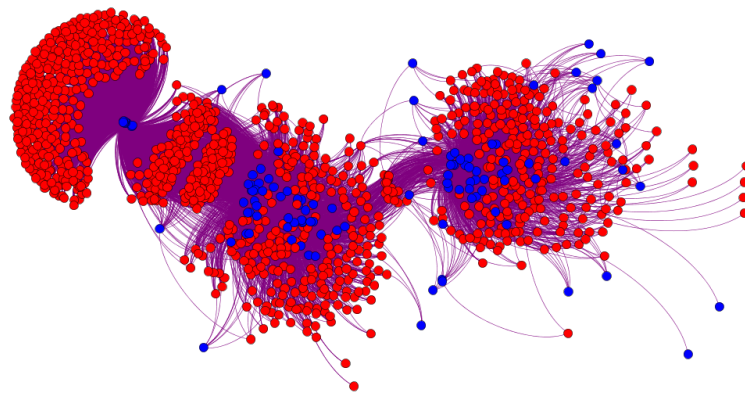
- [1] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.
- [2] Michael J Barber and John W Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.
- [3] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi: An open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.
- [4] Vladimir Batagelj and Andrej Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):47–57, 1998.
- [5] Vladimir Batagelj and Andrej Mrvar. *Pajek—analysis and visualization of large networks*. Springer, 2004.
- [6] Vladimir Batagelj and Andrej Mrvar. Program for analysis and visualization of large networks: reference manual, 2011.
- [7] Michael Baur, Marc Benkert, Ulrik Brandes, Sabine Cornelsen, Marco Gaertler, Boris Köpf, Jürgen Lerner, and Dorothea Wagner. Visone software for visual social network analysis. In *Graph Drawing*, pages 463–464. Springer, 2001.
- [8] Stephen P Borgatti and Martin G Everett. Models of core-periphery structures. *Social networks*, 21(4):375–395, 2000.

-
- [9] Stefan Bornholdt and Heinz Georg Schuster. *Handbook of Graphs and Networks*. Wiley Online Library, 2002.
- [10] Fan RK Chung and Linyuan Lu. *Complex graphs and networks*, volume 107. American mathematical society Providence, 2006.
- [11] Gennaro Cordasco and Luisa Gargano. Community detection via semi-synchronous label propagation algorithms. In *Business Applications of Social Network Analysis (BASNA), 2010 IEEE International Workshop on*, pages 1–8. IEEE, 2010.
- [12] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory social network analysis with Pajek*, volume 27. Cambridge University Press, 2011.
- [13] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.
- [14] Katherine Faust and Stanley Wasserman. Blockmodels: Interpretation and evaluation. *Social networks*, 14(1):5–61, 1992.
- [15] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.
- [16] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.
- [17] Sébastien Heymann and Bénédicte Le Grand. Visual analysis of complex networks for business intelligence with gephi. In *2013 17th International Conference on Information Visualisation*, pages 307–312. IEEE, 2013.
- [18] Petter Holme. Core-periphery organization of complex networks. *Physical Review E*, 72(4):046111, 2005.

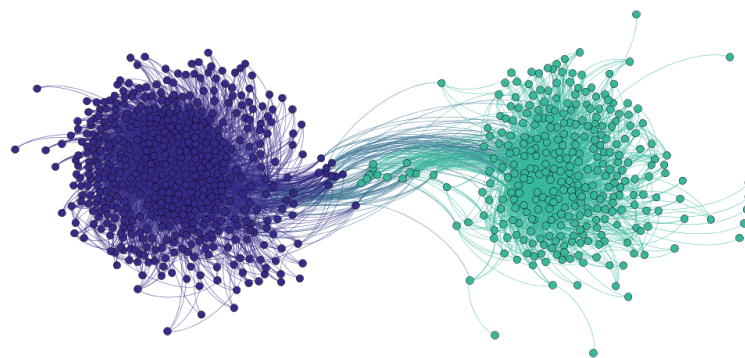
-
- [19] Xin Liu and Tsuyoshi Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.
- [20] Andrej Mrvar and Vladimir Batagelj. Analysis and visualization of large networks with program package pajek. *Complex Adaptive Systems Modeling*, 4(1):1–8, 2016.
- [21] Mark EJ Newman. *Networks: An Introduction*. Oxford University Press, 2010.
- [22] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.
- [23] M Puck Rombach, Mason A Porter, James H Fowler, and Peter J Mucha. Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, 74(1):167–190, 2014.
- [24] B. Yan and J. Luo. Multicore-periphery structure in networks. *ArXiv e-prints*, May 2016.
- [25] Xiao Zhang, Travis Martin, and Mark EJ Newman. Identification of core-periphery structure in networks. *Physical Review E*, 91(3):032803, 2015.
- [26] Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):046115, 2007.

Dodatek A

Grafi in tabele

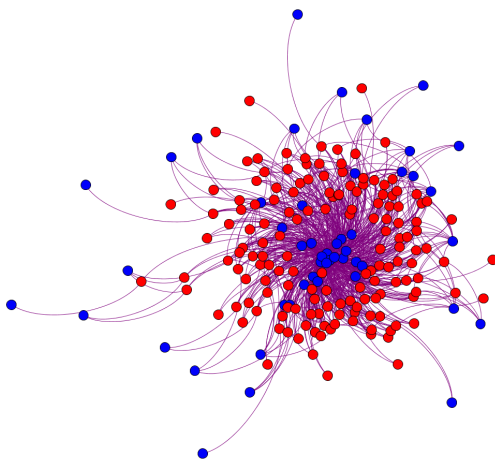


(a) Modra vozlišča predstavljajo predmete, rdeča pa študente.

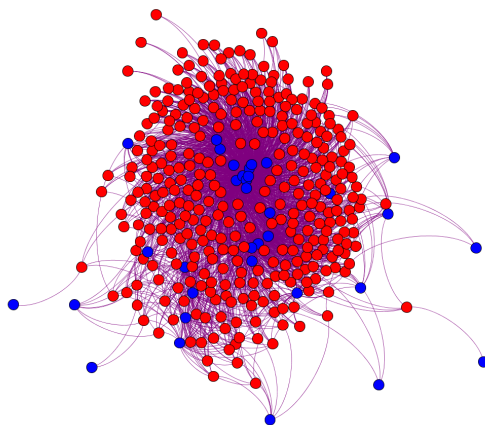


(b) Optimizacija modularnosti na omrežju brez obveznih predmetov.

Slika A.1: Dvoddelno omrežje vseh študijskih programov.



(a) Dvodnelno omrežje 2. letnika magistrskega študijskega programa z obveznimi predmeti.



(b) Dvodnelno omrežje 2. letnika visokošolskega študijskega programa brez obveznih predmetov.

Slika A.2: Modra vozlišča predstavljajo predmete, rdeča pa študente.

Osnovni podatki

	n	m	$\langle k \rangle$
Vsi študiji	820	6288	15,2
MAG (vsi letniki)	326	1977	12,1
MAG (1. letnik)	263	1231	9,3
MAG (2. letnik)	193	745	7,6
VSŠ (vsi letniki)	516	4311	16,6
VSŠ (2. letnik)	374	2618	13,9
VSŠ (3. letnik)	375	1690	8,9

Tabela A.1: Dvodavno omrežje brez obveznih predmetov.

	n	m	$\langle k \rangle$
Vsi študiji	92	1701	36,9
MAG (vsi letniki)	48	664	27,7
MAG (1. letnik)	47	446	18,9
MAG (2. letnik)	40	312	15,6
VSŠ (vsi letniki)	44	778	35,4
VSŠ (2. letnik)	32	358	22,4
VSŠ (3. letnik)	44	494	22,5

Tabela A.2: Enodelno omrežje brez obveznih predmetov.

Moduli in laboratoriji predmetov

Šifra	Ime predmeta	Moduli	Laboratorij
63226	Tehnologija upravljanja podatkov	INF	LAB15
63249	Elektronsko poslovanje	INF	LAB7
63250	Organizacija in management	INF	LAB6
63251	Poslovna inteligenca	RAC	LAB16
63252	Razvoj informacijskih sistemov	INF	LAB8
63253	Planiranje in upravljanje informatike	INF	LAB6
63254	Postopki razvoja programske opreme	RAC, INF	LAB9
63255	Spletno programiranje	RAC	LAB15
63256	Tehnologija programske opreme	RAC	LAB11
63258	Komunikacijski protokoli	RAC	LAB4
63259	Brezžična in mobilna omrežja	RAC	LAB5
63260	Digitalno načrtovanje	RAC	LAB3
63261	Porazdeljeni sistemi	RAC	LAB3
63262	Zanesljivost in zmogljivost računalniških sistemov	RAC	LAB5
63264	Sistemska programska oprema	RAC	LAB10
63265	Prevajalniki	RAC	LAB10
63266	Inteligentni sistemi	RAC	LAB15
63267	Umetno zaznavanje	RAC	LAB13
63268	Razvoj inteligentnih sistemov	RAC	LAB13
63269	Računalniška grafika in tehnologija iger	RAC	LAB1
63270	Multimedijski sistemi	RAC	LAB13
63271	Osnove oblikovanja	DRUGO	LAB12
63506	Matematika II	OBV	LAB17
63507	Programiranje	OBV	LAB15
63508	Algoritmi	OBV	LAB15
63509	Računalniški sistemi	OBV	LAB3
63510	Umetna inteligenca	UI	LAB14
63511	Brezžična senzorska omrežja	SO, OIV	LAB5
63512	Nekonvencionalne platforme in metode procesiranja	SO, RM	LAB5
63514	Obdelava biomedicinskih signalov in slik	UI, MM, SO	LAB2
63515	Sodobne metode razvoja programske opreme	PO	LAB11
63516	Digitalno procesiranje signalov	MM, SO	LAB20
63517	Izračunljivost in računska zahtevnost	ALG	LAB10
63518	E-izobraževanje	MM	LAB1
63519	Strojno ucenje	UI, RM	LAB15
63520	Uvod v bioinformatiko	UI, RM	LAB16
63521	Informacijska varnost in zasebnost	OIV, PO, ISSU	LAB7
63522	Numerična matematika	ALG, RM	LAB17
63523	Računalniška zvočna produkcija	MM, SO	LAB7
63525	Odkrivanje znanj iz podatkov	UI	LAB16
63526	Obvladovanje informatike	ISSU	LAB6
63527	Interaktivnost in oblikovanje informacij	MM	LAB12
63528	Kriptografija in računalniška varnost	OIV, ALG	LAB18

Šifra	Ime predmeta	Moduli	Laboratorij
63530	Računalniška forenzika	OIV	LAB19
63531	Sodobni pristopi in arhitekture pri razvoju informacijskih sistemov	ISSU	DRUGO
63532	Diskretna matematika	ALG, RM	LAB8
63533	Management proizvodnih in storitvenih procesov	PO	LAB9
63534	Obštudijska strokovna dejavnost I	SPL	LAB15
63535	Obštudijska strokovna dejavnost II	SPL	LAB15
63536	Izbrana poglavja iz računalništva in informatike	SPL	LAB15
63542	Računska topologija	ALG, RM	LAB17
63545	Aktualno raziskovalno področje I	STR	LAB13
63546	Aktualno raziskovalno področje II	STR	LAB13
63547	Poučevanje algoritmičnega razmišljanja	ALG	LAB16
63548	Magistrsko delo	OBV	DRUGO
63700	Diplomsko delo	OBV	DRUGO
63701	Uvod v računalništvo	OBV	LAB13
63702	Programiranje 1	OBV	LAB16
63703	Računalniška arhitektura	OBV	LAB20
63704	Matematika	OBV	LAB17
63705	Diskretne strukture	OBV	LAB17
63706	Programiranje 2	OBV	LAB10
63707	Podatkovne baze	OBV	LAB15
63708	Računalniške komunikacije	OBV	LAB4
63709	Operacijski sistemi	OBV	LAB12
63710	Osnove verjetnosti in statistike	OBV	LAB18
63711	Algoritmi in podatkovne strukture 1	OBV	LAB10
63712	Elektronsko in mobilno poslovanje	INF	LAB6
63713	Podatkovne baze 2	INF	LAB6
63714	Informacijski sistemi	INF	LAB6
63715	Grafično oblikovanje	DRUGO	LAB12
63716	Komunikacijski protokoli in omrežna varnost	RAC	LAB19
63717	Organizacija računalnikov	RAC	DRUGO
63718	Digitalna vezja	RAC	LAB3
63719	Računalniška grafika	RAC	LAB5
63720	Umetna inteligenca	RAC	LAB15
63721	Uporabniški vmesniki	RAC, INF	LAB2
63722	Prevajalniki in navidezni stroji	RAC	LAB10
63723	Algoritmi in podatkovne strukture 2	OBV	LAB19
63724	Testiranje in kakovost	RAC	LAB11
63725	Razvoj informacijskih sistemov	INF	LAB6
63726	Produkcija multimedijskih gradiv	RAC	LAB12
63727	Spletne tehnologije	RAC	LAB7
63728	Vhodno-izhodne naprave	RAC	LAB20
63729	Načrtovanje digitalnih naprav	RAC	LAB5
63732	Tehnologija programske opreme	RAC	LAB1
63733	Strateško planiranje informatike	INF	LAB8
63734	Multimedijske tehnologije	RAC	LAB1

Šifra	Ime predmeta	Moduli	Laboratorij
63735	Vzporedni in porazdeljeni sistemi in algoritmi	RAC	LAB3
63736	Sistemska programska oprema	RAC	LAB3
63737	Procesna avtomatika	RAC	LAB3
63738	Vgrajeni sistemi	RAC	LAB3
63739	Robotika in računalniško zaznavanje	RAC	LAB13
63740	Tehnologija iger in navidezna resničnost	RAC	LAB12
63741	Odločitveni sistemi	RAC	LAB14
63742	Numerične metode	RAC	LAB17
63743	Delovna praksa	OBV	DRUGO
63744	Digitalno procesiranje signalov	RAC	LAB2
63745	Angleški jezik nivo A	DRUGO	DRUGO
63746	Angleški jezik nivo B	DRUGO	DRUGO
63747	Angleški jezik nivo C	DRUGO	DRUGO
63748	Ekonomika in organizacija informatike	INF	DRUGO
63749	Izbrana poglavja iz računalništva in informatike	RAC, INF	DRUGO
63750	Športna vzgoja	DRUGO	DRUGO
63752	Računalništvo v praksi I	RAC, INF	DRUGO
63753	Računalništvo v praksi II	RAC, INF	DRUGO
63754	Izvedbe algoritmov (TP)	RAC	LAB3
63755	Projektne praktikum	OBV	LAB12
63764	Računalniška orodja, jeziki in okolja (TP A)	RAC	DRUGO
63765	Podatkovno rudarjenje	RAC	LAB16
63766	Tehnične veščine 2	RAC, INF	DRUGO
63767	Tehnične veščine	RAC, INF	DRUGO
63768	Planiranje in upravljanje informatike	INF	LAB6
63769	Programski jezik C	RAC	LAB10

Tabela A.3: Za vsak predmet so podani moduli, katerim je predmet določen in laboratorij, kateremu pripada nosilec predmeta.

1. letnik magistrskega študijskega programa

Šifra	Ime predmeta	Izbranost	Skupnost
63506	Matematika II	212	1
63507	Programiranje	212	1
63508	Algoritmi	212	1
63509	Računalniški sistemi	212	1
63530	Računalniška forenzika	104	1
63515	Sodobne metode razvoja programske opreme	91	1
63521	Informacijska varnost in zasebnost	77	2
63518	E-izobraževanje	75	1
63533	Management proizvodnih in storitvenih procesov	66	2
63527	Interaktivnost in oblikovanje informacij	58	2
63520	Uvod v bioinformatiko	52	1
63525	Odkrivanje znanj iz podatkov	51	1
63545	Aktualno raziskovalno področje I	43	2
63547	Poučevanje algoritmičnega razmišljanja	40	2
63546	Aktualno raziskovalno področje II	34	2
63514	Obdelava biomedicinskih signalov in slik	32	2
63534	Obštudijska strokovna dejavnost I	32	2
63510	Umetna inteligenca	31	2
63535	Obštudijska strokovna dejavnost II	30	2
63512	Nekonvencionalne platforme in metode procesiranja	29	2
63542	Računska topologija	28	2
63526	Obvladovanje informatike	27	2
63516	Digitalno procesiranje signalov	26	2
63519	Strojno učenje	26	2
63528	Kriptografija in računalniška varnost	23	2
63517	Izračunljivost in računska zahtevnost	22	2
63522	Numerična matematika	22	2
63271	Osnove oblikovanja	20	2
63532	Diskretna matematika	19	2
63536	Izbrana poglavja iz računalništva in informatike	19	2
63523	Računalniška zvočna produkcija	17	2
63511	Brezžična senzorska omrežja	16	2
63250	Organizacija in management	16	2
63226	Tehnologija upravljanja podatkov	15	3
63261	Porazdeljeni sistemi	9	3
63255	Spletno programiranje	8	3
63254	Postopki razvoja programske opreme	7	3
63267	Umetno zaznavanje	6	3
63269	Računalniška grafika in tehnologija iger	6	3
63252	Razvoj informacijskih sistemov	6	3
63259	Brezžična in mobilna omrežja	6	3
63265	Prevajalniki	5	3
63266	Inteligentni sistemi	5	3
63268	Razvoj inteligentnih sistemov	5	3

Šifra	Ime predmeta	Izbranost	Skupnost
63260	Digitalno načrtovanje	4	3
63270	Multimedijski sistemi	4	3
63251	Poslovna inteligenca	4	3
63262	Zanesljivost in zmogljivost računalniških sistemov	3	3
63249	Elektronsko poslovanje	3	3
63531	Sodobni pristopi in arhitekture pri razvoju informacijskih sistemov	3	3
63258	Komunikacijski protokoli	3	3

Tabela A.4: Razporeditev predmetov v skupnosti, dobljene s stohastičnim bločnim modeliranjem na enodelnem omrežju 1. letnika magistrskega študijskega programa z obveznimi predmeti.

2. letnik magistrskega študijskega programa

Šifra	Ime predmeta	Izbranost	Skupnost
63548	Magistrsko delo	155	1
63514	Obdelava biomedicinskih signalov in slik	54	1
63521	Informacijska varnost in zasebnost	48	1
63525	Odkrivanje znanj iz podatkov	45	1
63547	Poučevanje algoritmičnega razmišljanja	42	1
63527	Interaktivnost in oblikovanje informacij	40	1
63520	Uvod v bioinformatiko	39	1
63515	Sodobne metode razvoja programske opreme	38	2
63512	Nekonvencionalne platforme in metode procesiranja	33	2
63545	Aktualno raziskovalno področje I	32	2
63533	Management proizvodnih in storitvenih procesov	31	1
63523	Računalniška zvočna produkcija	27	1
63546	Aktualno raziskovalno področje II	25	2
63526	Obvladovanje informatike	23	1
63530	Računalniška forenzika	23	1
63518	E-izobraževanje	22	1
63510	Umetna inteligenca	20	1
63511	Brezžična senzorska omrežja	16	2
63519	Strojno učenje	15	2
63542	Računska topologija	14	1
63517	Izračunljivost in računska zahtevnost	14	2
63528	Kriptografija in računalniška varnost	14	2
63536	Izbrana poglavja iz računalništva in informatike	14	2
63271	Osnove oblikovanja	12	1
63532	Diskretna matematika	10	1
63226	Tehnologija upravljanja podatkov	9	1
63522	Numerična matematika	9	1
63534	Obštudijska strokovna dejavnost I	8	1
63249	Elektronsko poslovanje	7	2
63251	Poslovna inteligenca	7	1
63535	Obštudijska strokovna dejavnost II	7	1
63516	Digitalno procesiranje signalov	6	2
63250	Organizacija in management	5	2
63255	Spletno programiranje	5	1
63258	Komunikacijski protokoli	5	2
63253	Planiranje in upravljanje informatike	4	1
63261	Porazdeljeni sistemi	3	2
63269	Računalniška grafika in tehnologija iger	3	1
63507	Programiranje	3	1
63252	Razvoj informacijskih sistemov	3	1
63254	Postopki razvoja programske opreme	3	1
63259	Brezžična in mobilna omrežja	3	1
63262	Zanesljivost in zmogljivost računalniških sistemov	2	2
63506	Matematika II	2	1

Šifra	Ime predmeta	Izbranost	Skupnost
63508	Algoritmi	2	1
63270	Multimedijski sistemi	2	1
63265	Prevajalniki	1	1
63268	Razvoj inteligentnih sistemov	1	1
63260	Digitalno načrtovanje	1	1
63531	Sodobni pristopi in arhitekture pri razvoju informacijskih sistemov	1	1

Tabela A.5: Razporeditev predmetov v skupnosti, dobljene z optimizacijo modularnosti na enodelnem omrežju 2. letnika magistrskega študijskega programa z obveznimi predmeti.

2. letnik visokošolskega študijskega programa

Šifra	Ime predmeta	Izbranost	Skupnost
63716	Komunikacijski protokoli in omrežna varnost	233	1
63727	Spletne tehnologije	221	1
63721	Uporabniški vmesniki	194	1
63715	Grafično oblikovanje	180	1
63720	Umetna inteligenca	168	2
63719	Računalniška grafika	164	2
63712	Elektronsko in mobilno poslovanje	163	1
63714	Informacijski sistemi	162	1
63726	Produkcija multimedijskih gradiv	151	1
63725	Razvoj informacijskih sistemov	138	1
63744	Digitalno procesiranje signalov	114	1
63724	Testiranje in kakovost	114	1
63765	Podatkovno rudarjenje	106	2
63713	Podatkovne baze 2	90	1
63728	Vhodno-izhodne naprave	64	2
63717	Organizacija računalnikov	60	2
63722	Prevajalniki in navidezni stroji	46	2
63718	Digitalna vezja	43	2
63764	Računalniška orodja, jeziki in okolja (TP A)	33	2
63729	Načrtovanje digitalnih naprav	33	2
63750	Športna vzgoja	27	1
63769	Programski jezik C	22	2
63745	Angleški jezik nivo A	17	1
63749	Izbrana poglavja iz računalništva in informatike	16	2
63746	Angleški jezik nivo B	14	1
63747	Angleški jezik nivo C	10	1
63752	Računalništvo v praksi I	10	2
63754	Izvedbe algoritmov (TP)	8	2
63767	Tehnične veščine	7	2
63753	Računalništvo v praksi II	3	2
63732	Tehnologija programske opreme	3	1
63766	Tehnične veščine 2	2	2
63733	Strateško planiranje informatike	1	2
63734	Multimedijske tehnologije	1	2

Tabela A.6: Razporeditev predmetov v skupnosti, dobljene z optimizacijo modularnosti na dvodelnem omrežju 2. letnika visokošolskega študijskega programa brez obveznih predmetov.

3. letnik visokošolskega študijskega programa

Šifra	Ime predmeta	Izbranost	Skupina
63732	Tehnologija programske opreme	170	1
63712	Elektronsko in mobilno poslovanje	96	1
63713	Podatkovne baze 2	91	1
63740	Tehnologija iger in navidezna resničnost	90	1
63733	Strateško planiranje informatike	84	1
63734	Multimedijske tehnologije	82	1
63736	Sistemska programska oprema	80	1
63735	Vzporedni in porazdeljeni sistemi in algoritmi	78	1
63739	Robotika in računalniško zaznavanje	62	1
63750	Športna vzgoja	56	1
63741	Odločitveni sistemi	55	1
63714	Informacijski sistemi	53	1
63721	Uporabniški vmesniki	48	1
63749	Izbrana poglavja iz računalništva in informatike	45	1
63715	Grafično oblikovanje	42	1
63737	Procesna avtomatika	40	2
63738	Vgrajeni sistemi	37	1
63719	Računalniška grafika	34	2
63768	Planiranje in upravljanje informatike	32	2
63720	Umetna inteligenca	31	1
63764	Računalniška orodja, jeziki in okolja (TP A)	31	2
63724	Testiranje in kakovost	31	1
63746	Angleški jezik nivo B	31	1
63717	Organizacija računalnikov	25	2
63718	Digitalna vezja	25	1
63745	Angleški jezik nivo A	24	1
63744	Digitalno procesiranje signalov	22	1
63727	Spletne tehnologije	20	2
63742	Numerične metode	19	2
63725	Razvoj informacijskih sistemov	19	2
63726	Produkcija multimedijskih gradiv	19	2
63765	Podatkovno rudarjenje	18	2
63752	Računalništvo v praksi I	18	2
63716	Komunikacijski protokoli in omrežna varnost	17	1
63722	Prevaljalniki in navidezni stroji	13	3
63728	Vhodno-izhodne naprave	13	3
63767	Tehnične veščine	7	3
63729	Načrtovanje digitalnih naprav	7	3

Šifra	Ime predmeta	Izbranost	Skupina
63766	Tehnične veščine 2	5	3
63753	Računalništvo v praksi II	5	3
63747	Angleški jezik nivo C	4	3
63769	Programski jezik C	4	3
63748	Ekonomika in organizacija informatike	4	3
63754	Izvedbe algoritmov (TP)	3	3

Tabela A.7: Razporeditev predmetov v skupine, dobljene z bločnim modeliranjem na dvodelnem omrežju 3. letnika visokošolskega študijskega programa brez obveznih predmetov.