Analiza omrežij

1. Uvod

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, april 2024

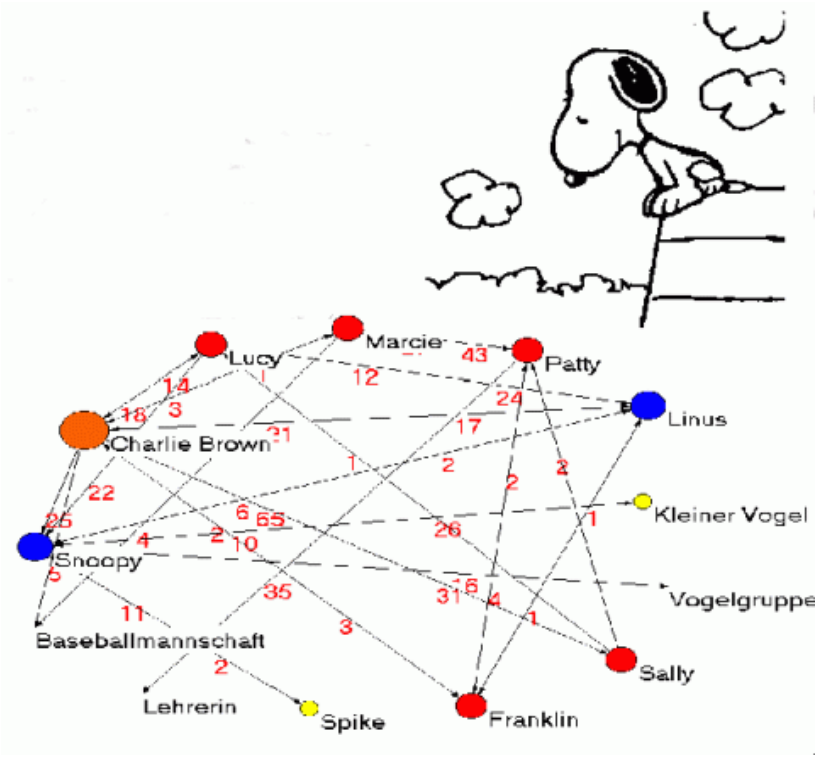
Kazalo

1 Uvod
2 Primeri
3 Razvoj
4 Viri

prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

22. april 2024 ob 01 : 13/ marec 2013

Analiza omrežij



Alexandra Schuler/ Marion Laging-Glaser:
Analyse von Snoopy Comics

Vse pogosteje se v analizi podatkov upošteva tudi odnose (relacije) med enotami. S tem se ukvarja analiza omrežij.

Cilj predmeta je seznaniti študente s problemi, metodami in orodji za analizo relacijskih podatkov ter jih usposobiti za samostojno uporabo leteh.

Wiki: http://vladowiki.fmf.uni-lj.si/doku.php?id=pajek:ev:pd:p24

Obveznosti študenta

Opravljene *domače naloge* – projekti.

1. projekt: priprava omrežij in osnove dela s programom `Pajek`.

2. projekt: uporaba metod analize omrežij na izbranem omrežju.

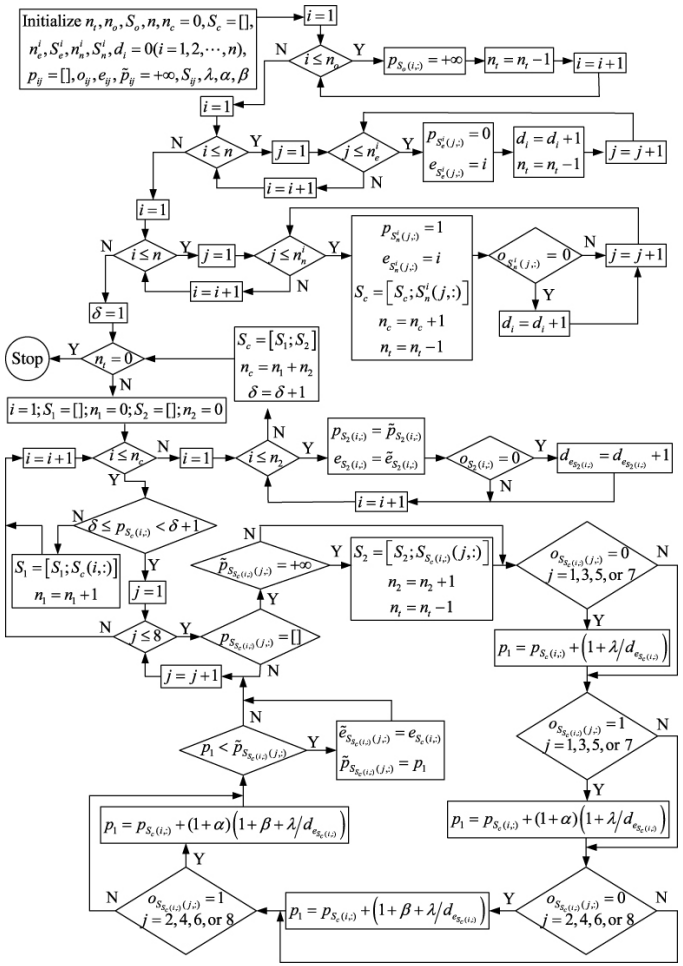
3. (zaključni) projekt: samostojna omrežna analiza izbranega večjega podatkovja.

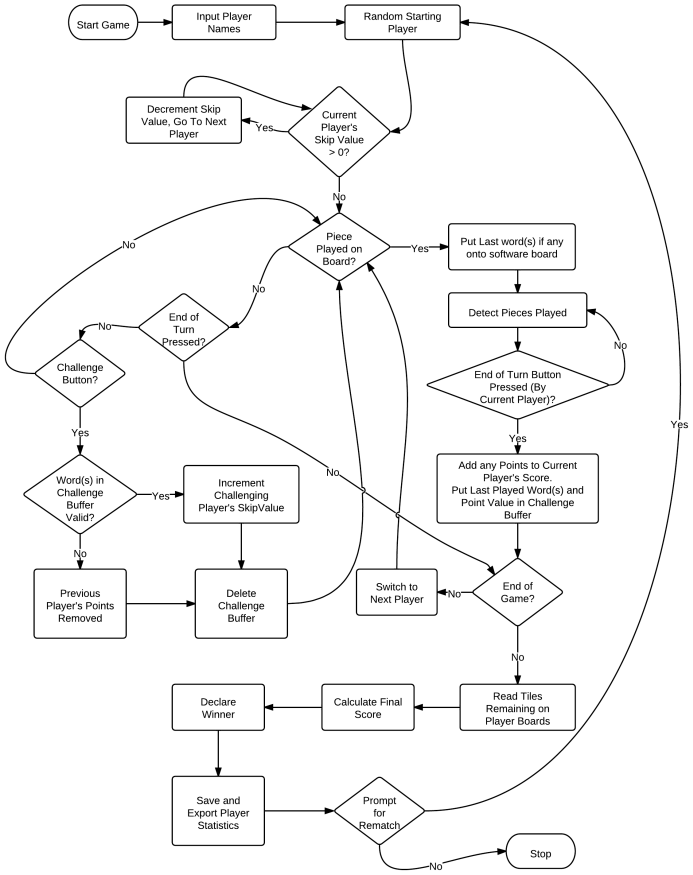
Rimsko cestno omrežje

Tabula Peutingeriana

Diagrami poteka

McCabe

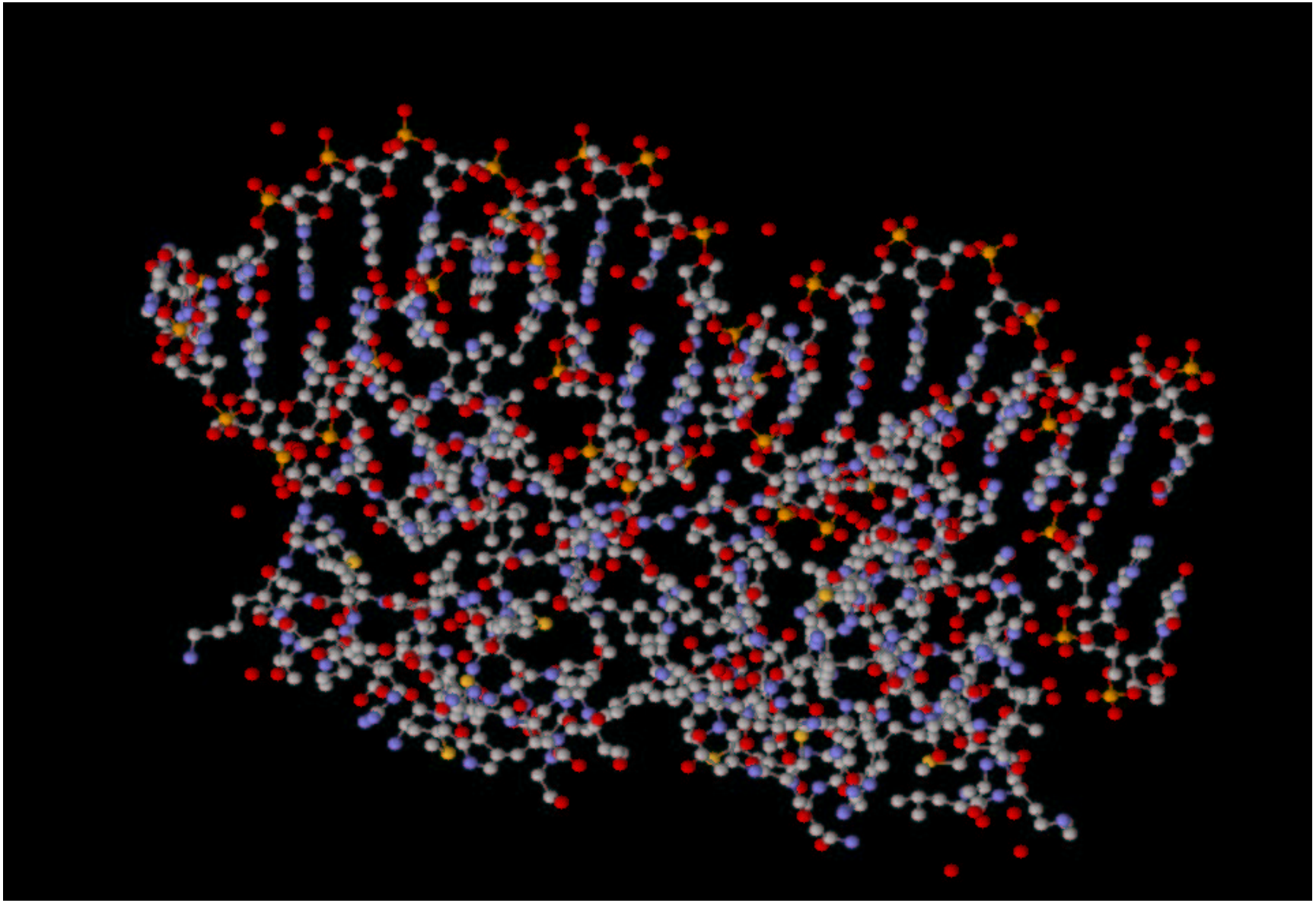




IOP

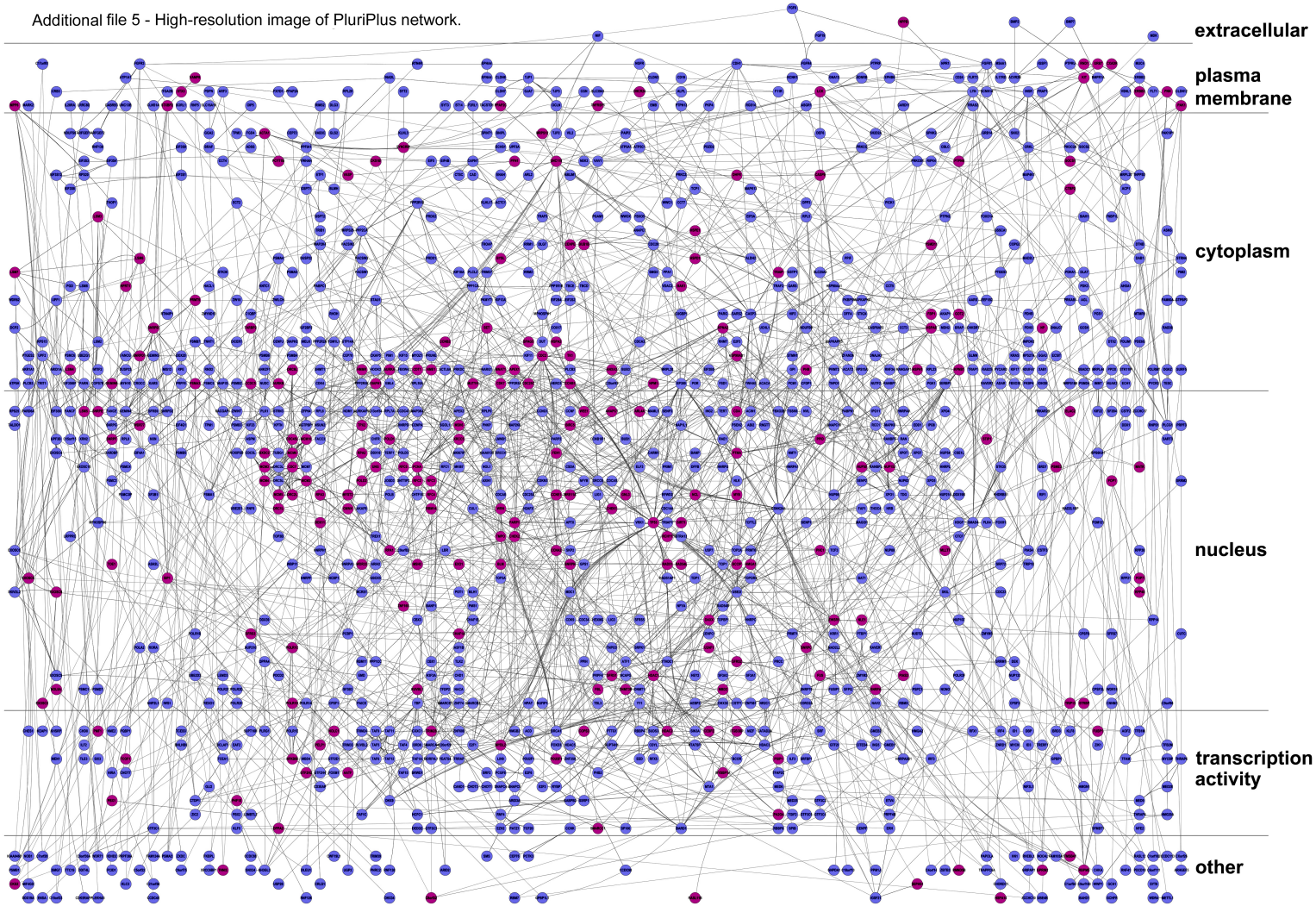
Velike organske molekule

3CRO, PDB



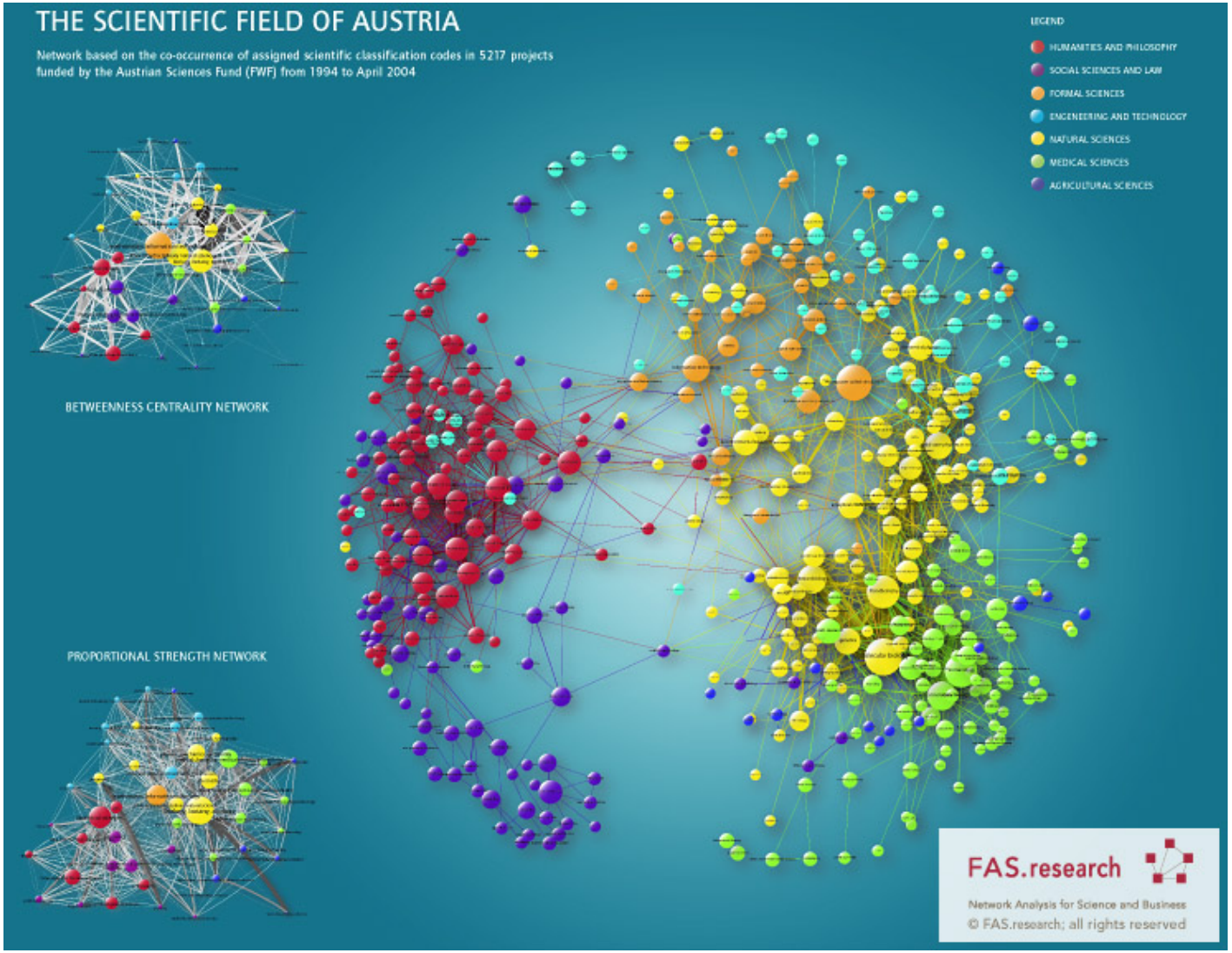
Beljakovinsko omrežje

PluriPlus



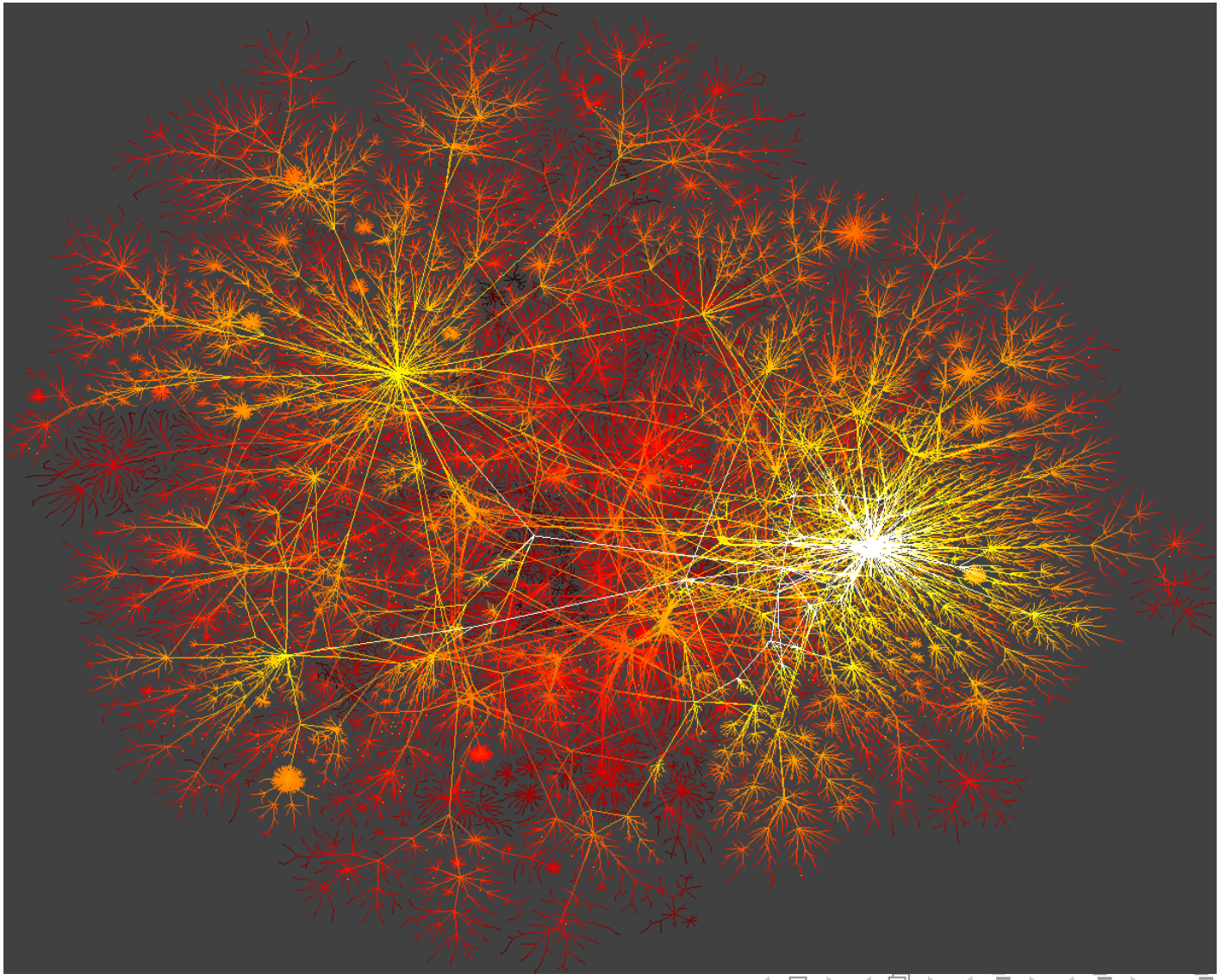
Avstrijski raziskovalni projekti

FAS

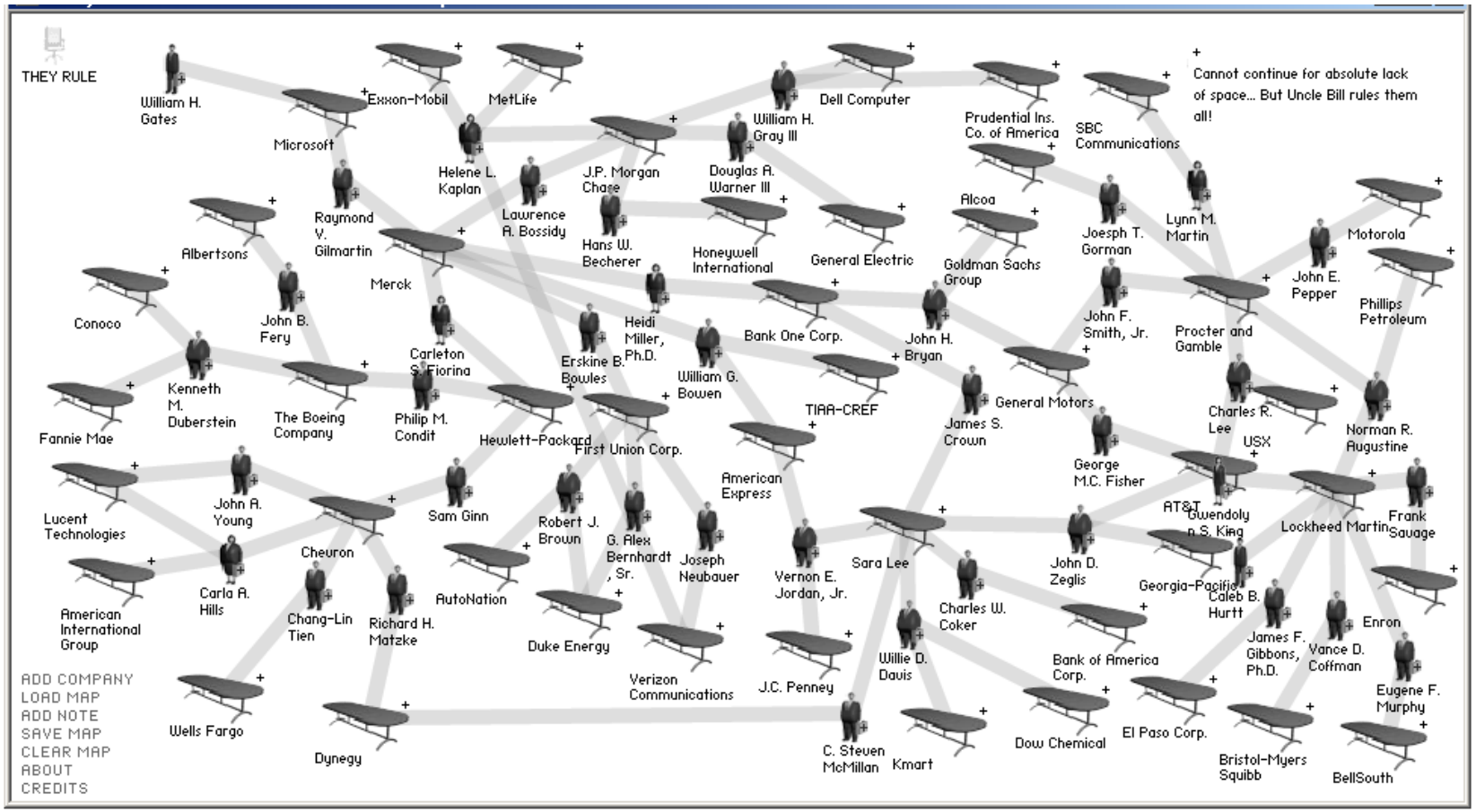


Internet

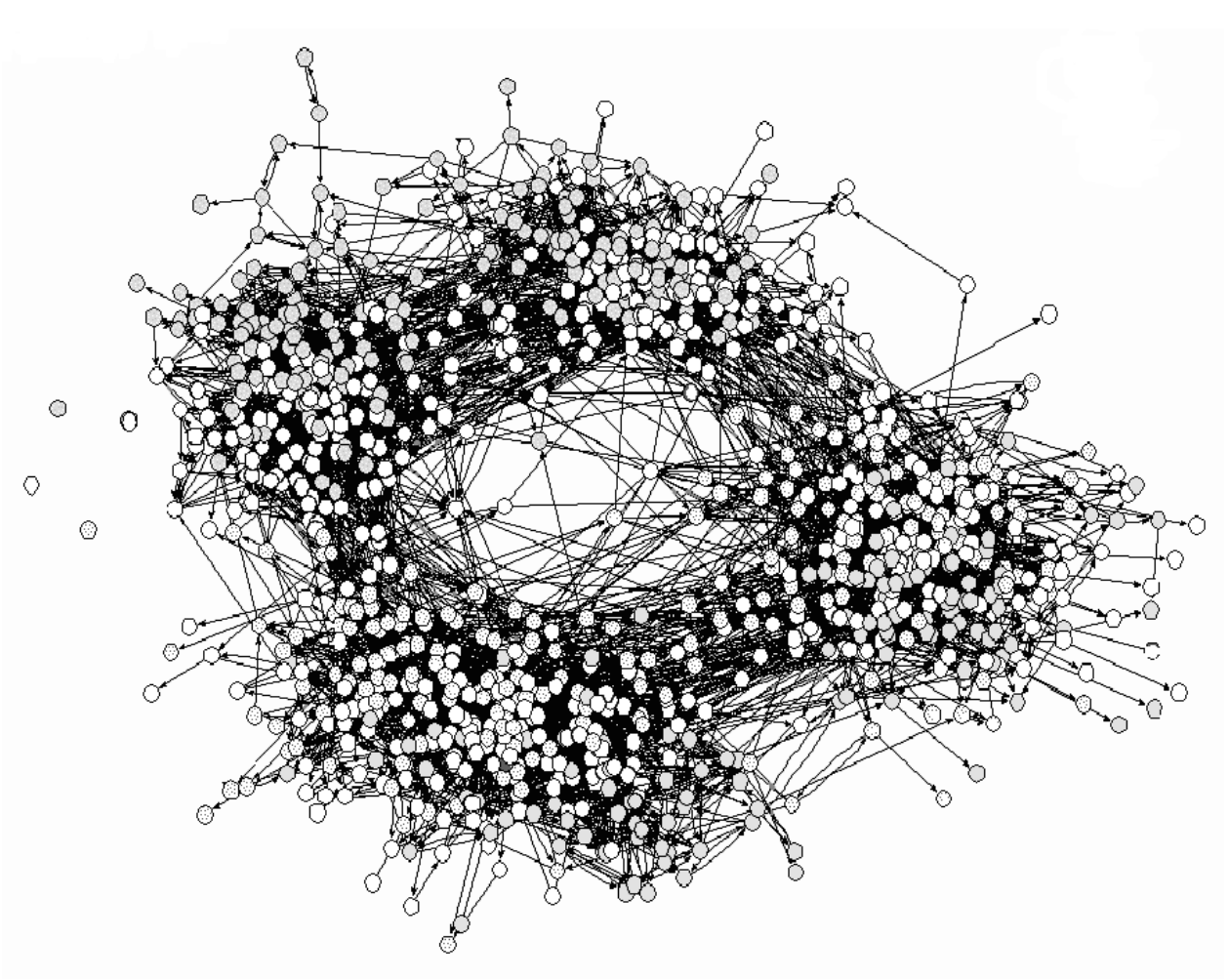
Cheswick, Opte



They Rule


THEY RULE
Cannot continue for absolute lack of space... But Uncle Bill rules them all!
William H. Gates
Microsoft
Exxon-Mobil
MetLife
Dell Computer
William H. Gray III
Prudential Ins. Co. of America
SBC Communications
Helene L. Kaplan
J.P. Morgan Chase
Douglas A. Warner III
Raymond V. Gilmartin
Lawrence A. Bossidy
Hans W. Becherer
Honeywell International
General Electric
Alcoa
Goldman Sachs Group
Joesph T. Gorman
Lynn M. Martin
Motorola
John E. Pepper
Phillips Petroleum
Albertsons
Merck
Conoco
John B. Fery
Heidi Miller, Ph.D.
Bank One Corp.
John H. Bryan
John F. Smith, Jr.
Procter and Gamble
Carleton S. Fiorina
Erskine B. Bowles
William G. Bowen
Kenneth M. Duberstein
The Boeing Company
Philip M. Condit
Hewlett-Packard
First Union Corp.
TIAA-CREF
James S. Crown
General Motors
Charles R. Lee
USX
Norman R. Augustine
Fannie Mae
American Express
George M.C. Fisher
Lucent Technologies
John A. Young
Sam Ginn
Robert J. Brown
G. Alex Bernhardt, Sr.
Joseph Neubauer
Vernon E. Jordan, Jr.
Sara Lee
John D. Zeglis
AT&T
Gwendolyn S. King
Lockheed Martin
Frank Savage
Chevron
AutoNation
Charles W. Coker
Georgia-Pacific
Caleb B. Hurtt
American International Group
Carla A. Hills
Chang-Lin Tien
Richard H. Matzke
Duke Energy
Verizon Communications
J.C. Penney
Willie D. Davis
Bank of America Corp.
James F. Gibbons, Ph.D.
Vance D. Coffman
Enron
Eugene F. Murphy
ADD COMPANY
LOAD MAP
ADD NOTE
SAVE MAP
CLEAR MAP
ABOUT
CREDITS
Wells Fargo
Dynegy
C. Steven McMillan
Kmart
Dow Chemical
El Paso Corp.
Bristol-Myers Squibb
BellSouth

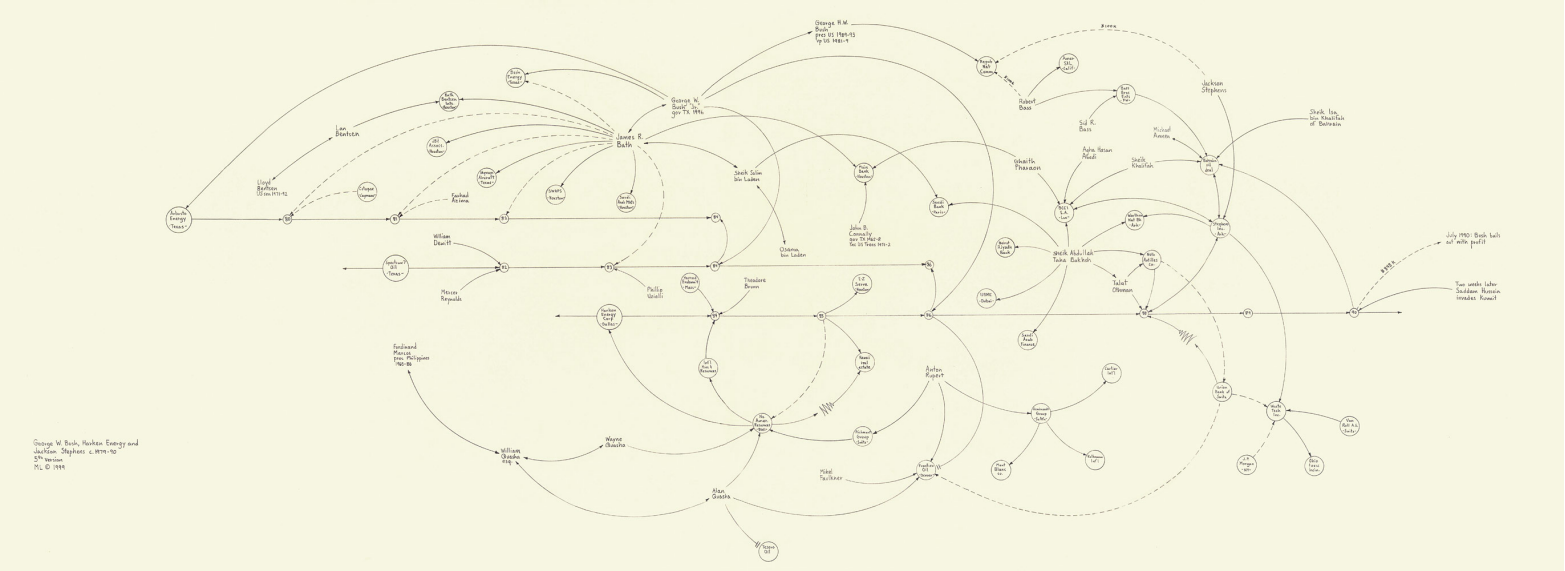
Podeželska šola – prijateljstvo



James Moody (2001) AJS Vol 107, 3,679–716

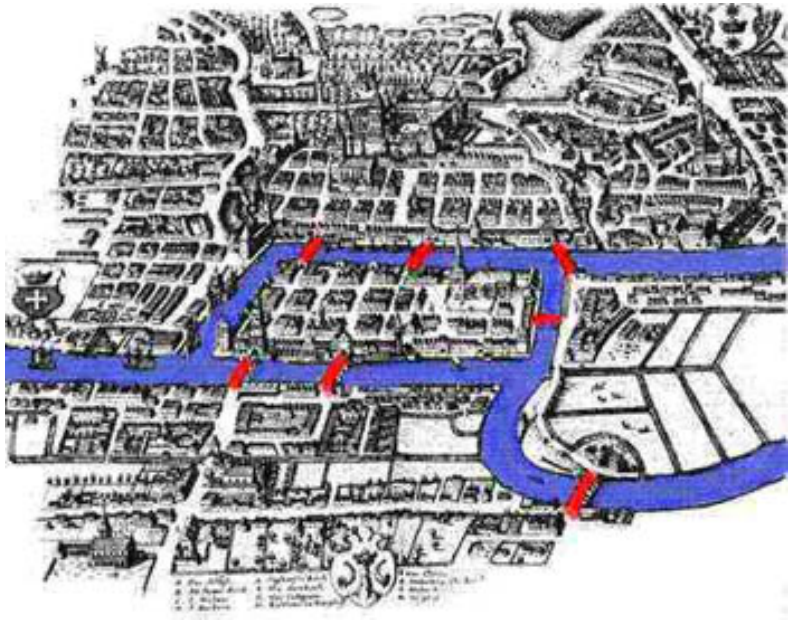
Mark Lombardi

Google



Še več primerov na Visual Complexity

Teorija grafov



- Euler (1736)
- Hamilton – obhodi vozlišč
- Kirchoff – električna vezja
- Kekule – molekule
- Ford, Fulkerson, Kruskal, Dijksta – optimizacija, OR
- Berge (1958, 1969), Harary (1965, 1969) – prve knjige
- transport (Wardrop)
- diagrami poteka (McCabe)
- kemija – napovedovanje (QSAR)

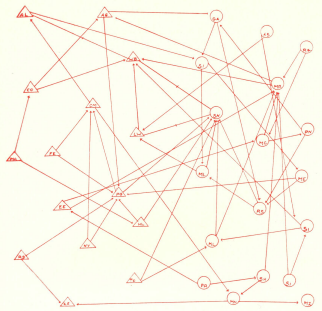
Začetniki analize družbenih omrežij

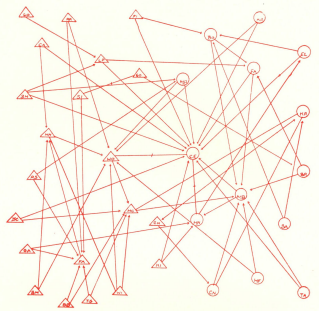
Moreno

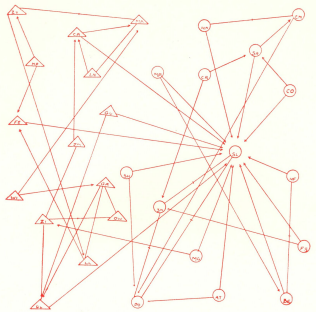
- Moreno (1934) – sociometrija
- Lewin (1936)
- Warner in Lunt (1941)
- Heider (1946)
- Bavelas (1948) – središčnosti
- Homans (1950)
- Cartwright in Harary (1965)
- Nadel (1957) – družbena mesta in vloge
- Mitchell (1969)

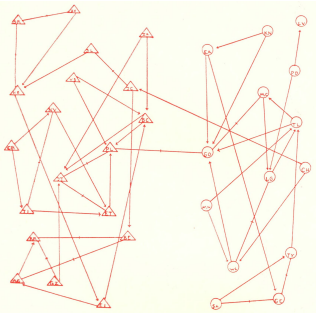
Freeman L.C. (2004) The Development of Social Network Analysis

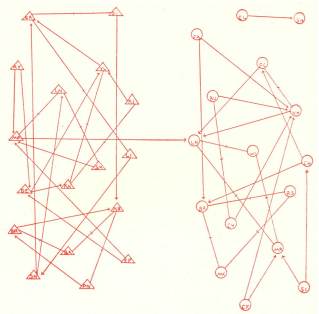
Moreno: Who shall survive?

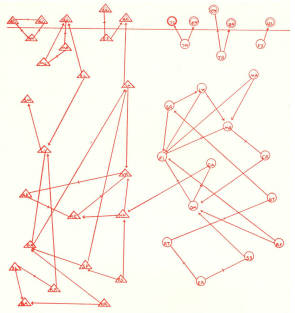
K: 

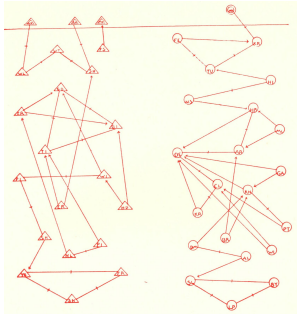
1: 

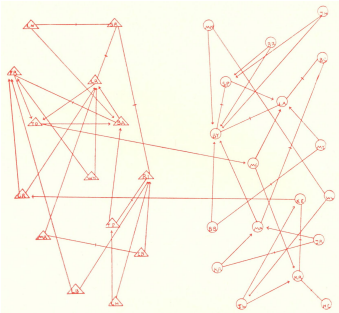
2: 

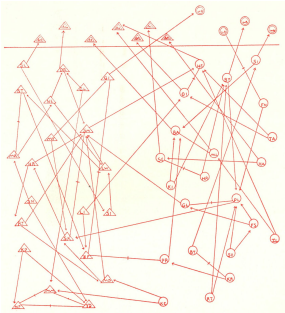
3: 

4: 

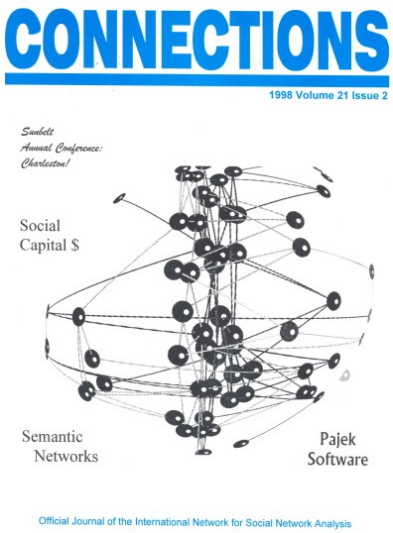
5: 

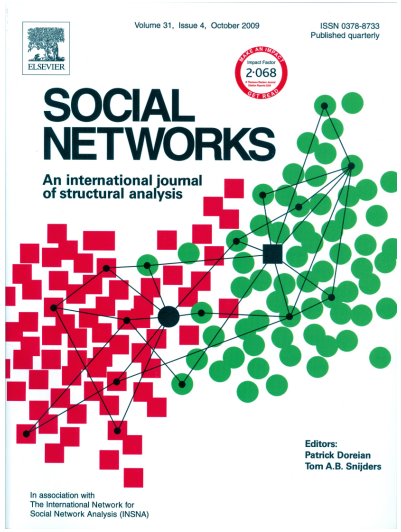
6: 

7: 

8: 

INSNA





- ustanovljeno združenje International Association of Social Network Analysis – INSNA, 1978
- obzornik: Connections, 1978
- revija: Social Networks, 1979
- srečanja: SUNBELT, 1981; Portorož 2004
- poštni seznam: SOCNET
- spletna revija: Journal of Social Structure, 2000
- revija: Social Network Analysis and Mining, 2011
- revija: Network Science, 2013
- European Conference of Social Networks (EUSN), Barcelona 2014 –

Zadnja desetletja

- prikazi omrežij (graph drawing) (1992)
- biologija – genom, beljakovinska omrežja
- fizikalni pristop – brezlestvična (scale free) omrežja (1998)
- ekonomska omrežja
- računalništvo / umetna inteligenca – link analysis
- družabna omrežja
- velika podatkovja (big data)
- biološka omrežja
- splošna analiza omrežij (network science)

Izbrane knjige o analizi omrežij

- J. P Scott: *Social Network Analysis*. SAGE Publications, 4ed, 2017. A.
- A. Degenne, M. Forsé: *Introducing Social Networks*. SAGE Publications, 1999. A.
- S. Wasserman, K. Faust: *Social Network Analysis: Methods and Applications*. CUP, 1994. A.
- W. de Nooy, A. Mrvar, V. Batagelj: *Exploratory Social Network Analysis with Pajek*, CUP, 2018, 3. izdaja. A. ESNA page.
- P. Doreian, V. Batagelj, A. Ferligoj: *Generalized Blockmodeling*, CUP, 2004. A.
- P.J. Carrington, J. Scott, S. Wasserman (Eds.): *Models and Methods in Social Network Analysis*. CUP, 2005. A.
- U. Brandes, T. Erlebach (Eds.): *Network Analysis: Methodological Foundations*. LNCS, Springer, Berlin 2005. A.
- E.D. Kolaczyk: *Statistical Analysis of Network Data: Methods and Models*. Springer, Berlin 2009.
- M.O. Jackson: *Social and Economic Networks*. Princeton UP, 2008.
- D. Easley, J. Kleinberg: *Networks, Crowds, and Markets*. CUP, 2010.
- M. Newman: *Networks*. OUP, 2ed, Oxford 2018.
- V. Batagelj, P. Doreian, A. Ferligoj, N. Kejžar: Understanding Large Temporal Networks and Spatial Networks. Wiley, 2014.
- A.-L. Barabási: *Network Science*. WWW 2015.
- S.P. Borgatti, M.G. Everett, J.C. Johnson: *Analyzing Social Networks*, 2ed. Sage 2018.
- S.P. Borgatti, M.G. Everett, J.C. Johnson, F. Agneessens: *Analyzing Social Networks Using R*. Sage 2022.

Programi za analizo omrežij

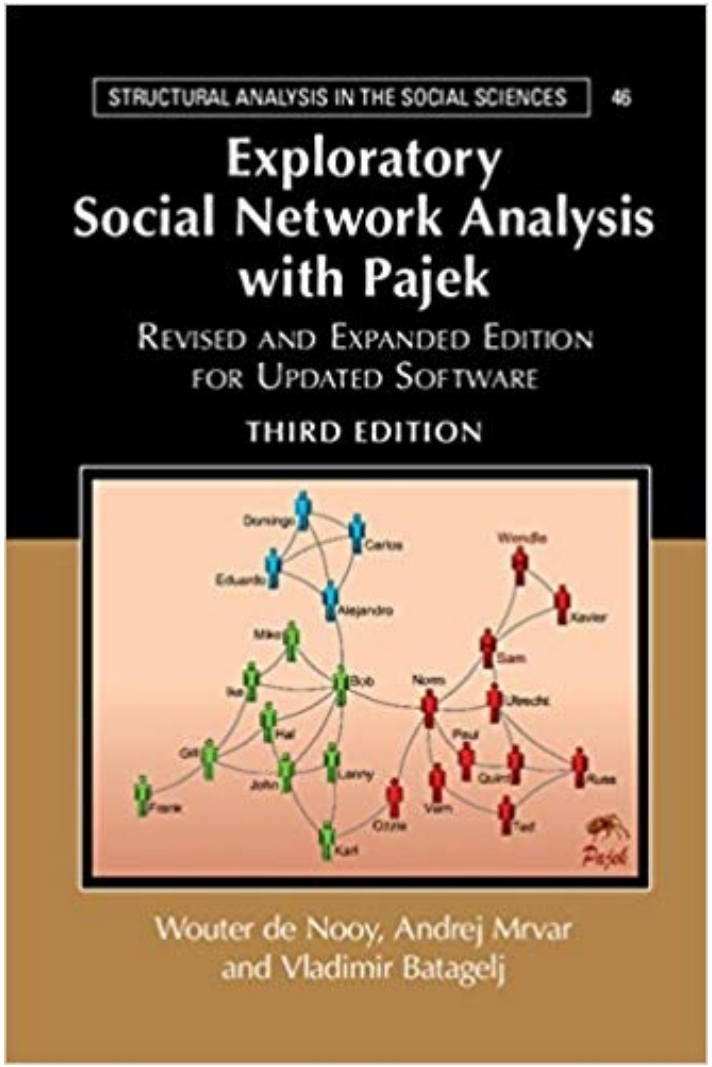
UCINET, NetDraw	Pajek	graph-tool
Visone	Cytoscape	Gephi
Negopy	InFlow	GUESS
NetworkX	prefuse	JUNG
SNAP	SNA, igraph / R	StOCNET, Siena
BGL/Python	Netminer	

Poglejte še seznam na Wikipedia ter preberite pregled Alen Orbanić in Monika Cerinšek Tools for Networks.

- Pajek: http://mrvar.fdv.uni-lj.si/pajek/
- igraph R package: http://igraph.org/r/
- Inkscape: https://inkscape.org/en/
- Ghostscript, Ghostview and GSview: http://pages.cs.wisc.edu/~ghost/

Primere podatkov o omrežjih najdete v Pajekovem seznamu.

ESNA Pajek



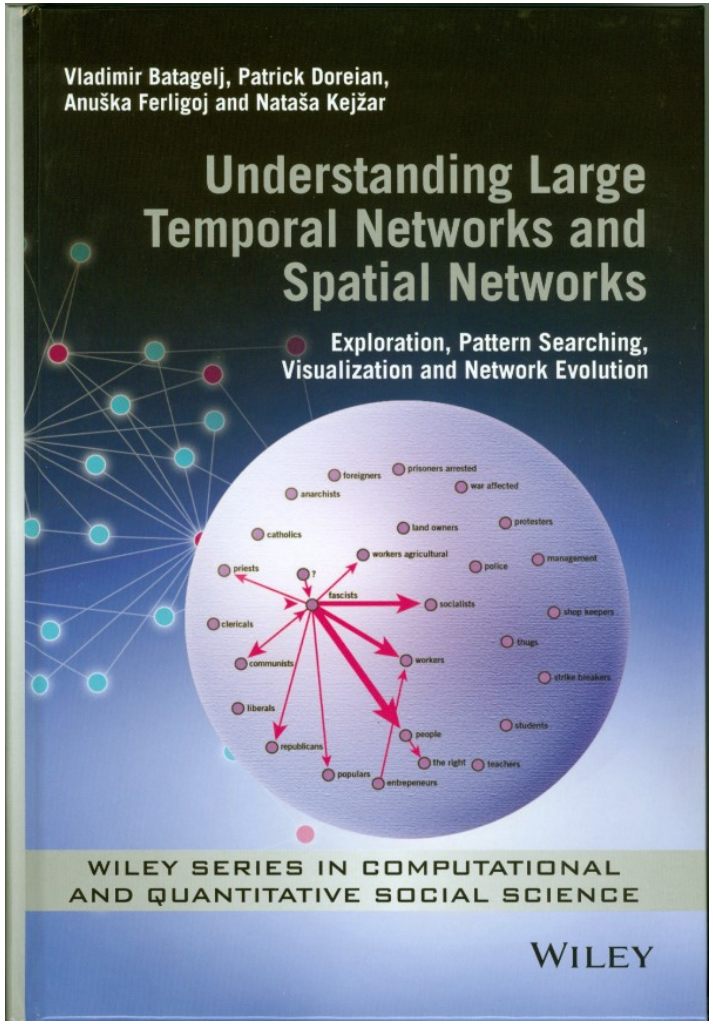
Pri delu z dejanskimi omrežji bomo v glavnem uporabljali domači program za analizo in prikaz velikih omrežij Pajek, ki je prosto dostopen na:

http://mrvar.fdv.uni-lj.si/pajek/

Uvod v analizo omrežij in program Pajek je mogoče najti v knjigi ESNA (de Nooy, Mrvar, Batagelj, CUP 2005; Revised and Expanded Edition for Updated Software, 2018).

Knjiga ESNA je izšla tudi v japonščini (Tokyo Denki University Press, 2010) in kitajščini (Beijing World Publishing, 2012).

Understanding . . .



Večino snovi je mogoče najti v drugem in tretjem poglavju knjige:

Vladimir Batagelj, Patrick Doreian, Anuška Ferligoj and Nataša Kejžar: Understanding Large Temporal Networks and Spatial Networks: Exploration, Pattern Searching, Visualization and Network Evolution. Wiley Series in Computational and Quantitative Social Science. Wiley, 2014.

Osnovni pojmi

Pajek

Opisi omrežij

Prikazi omrežij

Vrste omrežij

Pajek, Excel in R

Analiza omrežij

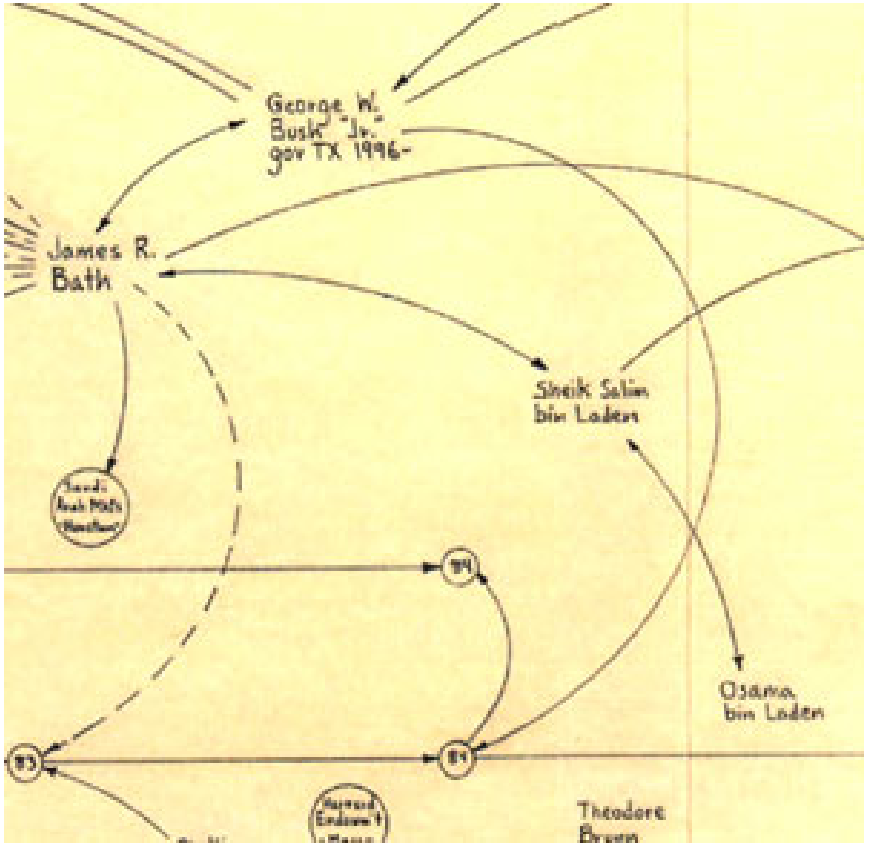
2. Vrste in opis omrežij

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, april 2024

Kazalo

1 Osnovni pojmi
2 Pajek
3 Opisi omrežij
4 Prikazi omrežij
5 Vrste omrežij
6 Pajek, Excel in R



prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

22. april 2024 ob 01 : 11/ marec 2013

Omrežja osnovni pojmi

Osnovni sestavini *omrežja* sta množica *vozlišč*, ki predstavljajo izbrane *enote*, in množica *povezav*, ki predstavljajo *odnose* (relacije) med enotami. Vozlišča in povezave določajo *graf*.

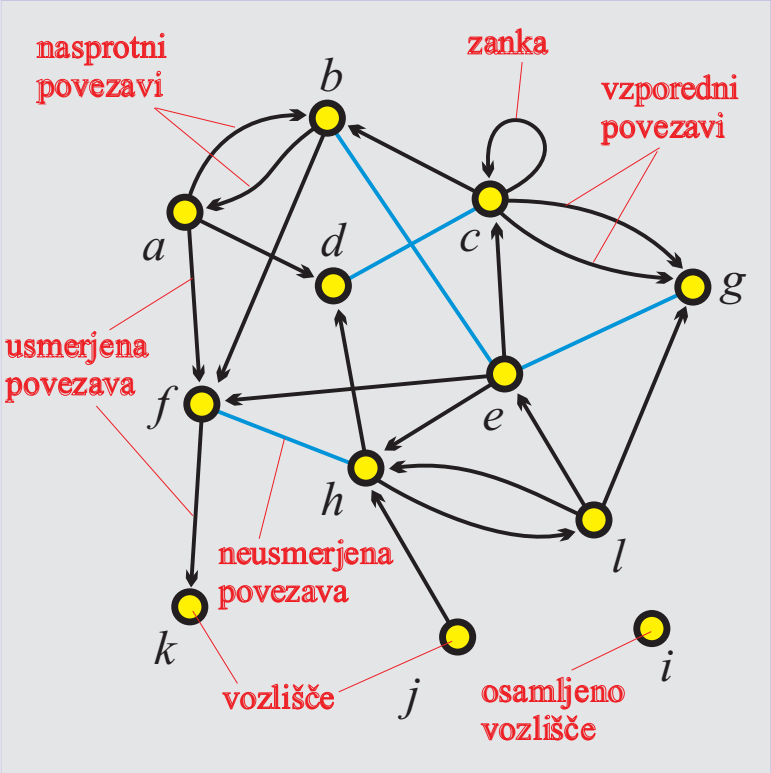
Pri povezavi je lahko smer pomembna – *usmerjena* povezava, ali pa ni *neusmerjena* povezava.

O vozliščih in povezavah lahko poznamo dodatne podatke – njihove *lastnosti*. Npr. oznaka, vrsta, vrednost, ...

$$\text{Omrežje} = \text{Graf} + \text{Podatki}$$

Ti podatki so lahko izmerjeni ali izračunani.

Graf



enota – *vozlišče*, točka

relacija – *povezava*

usmerjena povezava (a, d)

a je njen *začetek*

d pa njen *konec*.

neusmerjena povezava $(c : d)$

c in d sta njeni *krajišči*.

Omrežja / Formalno

Omrežje $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ je določeno z:

- *grafom* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, kjer je $\mathcal{V}$ množica vozlišč, $\mathcal{A}$ je množica usmerjenih povezav, in $\mathcal{E}$ je množica neusmerjenih povezav. Z $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ označimo množico vseh povezav. $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ *lastnosti* / vozliščne vrednosti: $p : \mathcal{V} \to A$
- $\mathcal{W}$ *uteži* / povezavne vrednosti: $w : \mathcal{L} \to B$

Velikost omrežja

Velikost omrežja/grafa je določena z dvema številoma: številom vozlišč $n = |\mathcal{V}|$ in številom povezav $m = |\mathcal{L}|$.

V *enostavnem neusmerjenem* grafu (ni vzporednih povezav in zank) je $m \leq \frac{1}{2}n(n-1)$; v *enostavnem usmerjenem* grafu (ni vzporednih povezav) pa $m \leq n^2$.
Razmerje $\gamma = \frac{m}{m_{max}}$ je *gostota* grafa.

Omrežja na nekaj deset vozliščih so *majhna* – lahko jih narišemo in za njihovo analizo uporabimo veliko algoritmov (**UCINET**, **NetMiner**). Tudi *srednje* velika omrežja (nekaj sto vozlišč) še lahko narišemo (preglednost !?), nekateri postopki že odpovejo.

Do začetka 90. let je bila večina omrežij majhnih – raziskovalci so jih zbrali z anketami, opazovanjem, iz arhivskih zapisov, ... Razvoj IT je omogočil, da je bilo mogoče ustvariti omrežja iz podatkov že zbranih na računalnikih. *Velika* omrežja so postala dejstvo. Velika omrežja ne moremo naenkrat podrobno prikazati v celoti. Za njihovo analizo in prikaz potrebujem posebne, nove pristope. Za to je bil razvit program `Pajek`.

Velika omrežja

Veliko omrežje – več tisoč ali celo milijonov vozlišč. V celoti ga lahko shranimo v pomnilnik računalnika – sicer *ogromno* omrežje. 64-bitni računalniki!

Jure Leskovec: SNAP – Stanford Large Network Dataset Collection

Social networks

Name	Type	Nodes	Edges	Description
ego-Facebook	Undirected	4,039	88,234	Social circles from Facebook (anonymized)
ego-Gplus	Directed	107,614	13,673,453	Social circles from Google+
ego-Twitter	Directed	81,306	1,768,149	Social circles from Twitter
soc-Epinions1	Directed	75,879	508,837	Who-trusts-whom network of Epinions.com
soc-LiveJournal1	Directed	4,847,571	68,993,773	LiveJournal online social network
soc-Pokec	Directed	1,632,803	30,622,564	Pokec online social network
soc-Slashdot0811	Directed	77,360	905,468	Slashdot social network from November 2008
soc-Slashdot0922	Directed	82,168	948,464	Slashdot social network from February 2009
wiki-Vote	Directed	7,115	103,689	Wikipedia who-votes-on-whom network

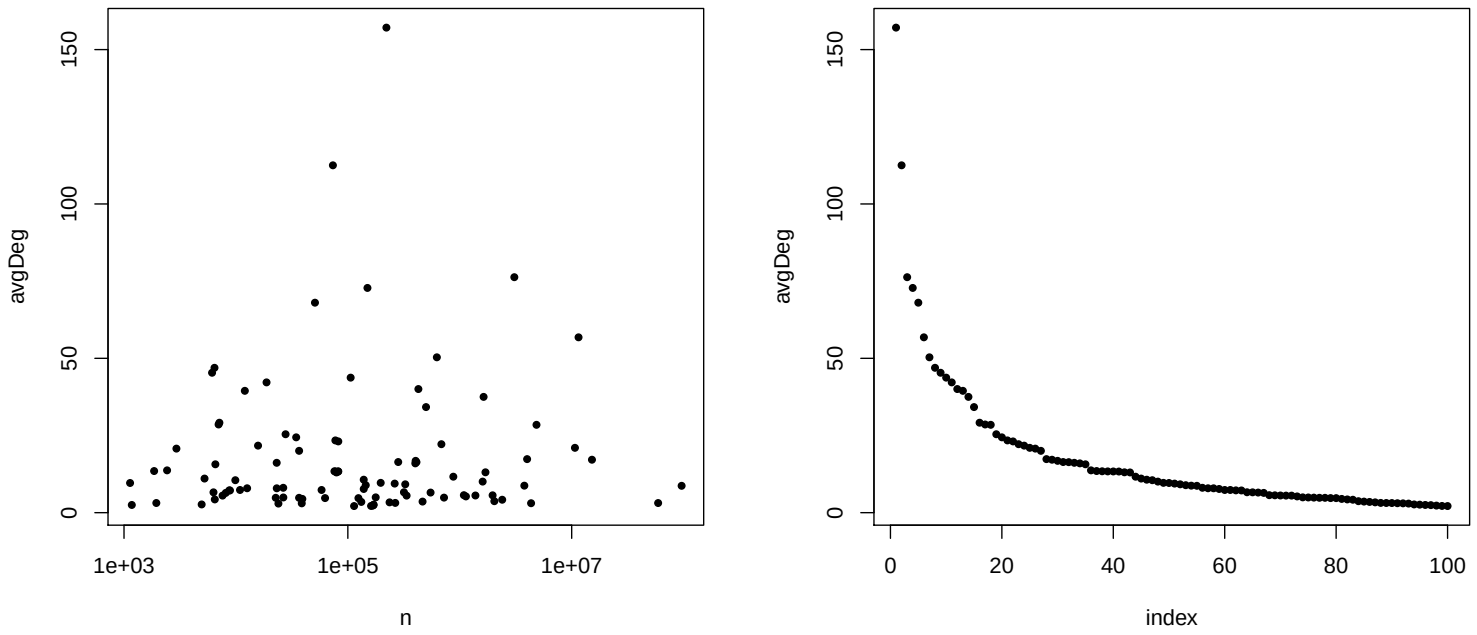
Networks with ground-truth communities

Name	Type	Nodes	Edges	Communities	Description
com-LiveJournal	Undirected, Communities	3,997,962	34,681,189	287,512	LiveJournal online social network
com-Friendster	Undirected, Communities	65,608,366	1,806,067,135	957,154	Friendster online social network
com-Orkut	Undirected, Communities	3,072,441	117,185,083	6,288,363	Orkut online social network

Pajek datasets. Konect.

Dunbarovo število

Povprečne stopnje omrežij SNAP in Konect:



Stopnja $\deg(v)$ je enaka številu povezav, ki imajo vozlišče v za krajišče. Povprečna stopnja $\bar{d} = \frac{1}{n}\sum_{v \in V} \deg(v) = \frac{2m}{n}$. Večina dejanskih omrežij je *redkih* – število vozlišč n in število povezav m sta istega velikostnega reda.
To opažanje je znano kot Dunbarovo število. Izhaja iz naslednjega razmisleka: recimo, da vsako vozlišče porabi za vzdrževanje posamezne povezave določeno "energijo" in da ima omejeno celotno energijo. Tedaj je tudi število povezav omejeno. V človeških združbah je Dunbarovo število običajno med 100 in 150.

Zahtevnost algoritmov

Poglejmo časovne zahtevnosti nekaj značilnih algoritmov:

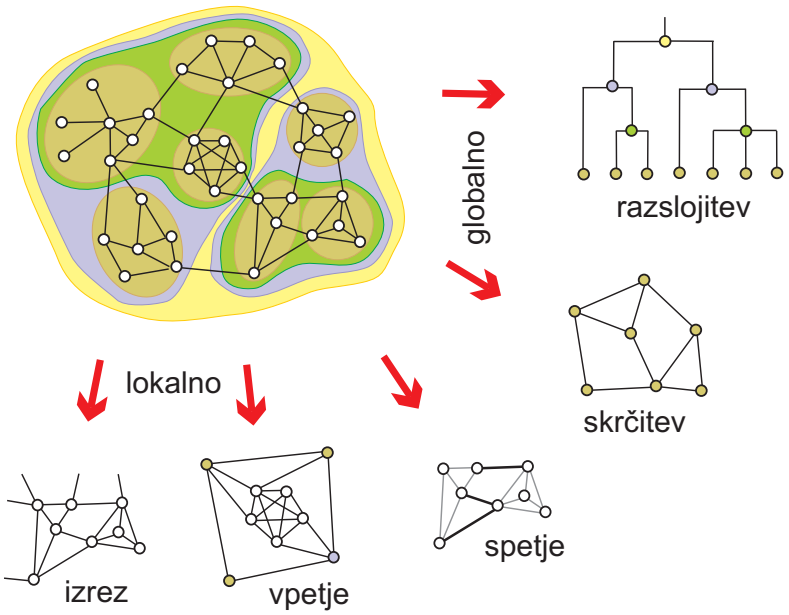
	T(n)	1.000	10.000	100.000	1.000.000	10.000.000
LinAlg	O(n)	0.00 s	0.015 s	0.17 s	2.22 s	22.2 s
LogAlg	O($n \log n$)	0.00 s	0.06 s	0.98 s	14.4 s	2.8 m
SqrtAlg	O($n\sqrt{n}$)	0.01 s	0.32 s	10.0 s	5.27 m	2.78 h
SqrAlg	O(n^2)	0.07 s	7.50 s	12.5 m	20.8 h	86.8 d
CubAlg	O(n^3)	0.10 s	1.67 m	1.16 d	3.17 L	3.17 kL

Za interaktivno uporabo na velikih podatkih so že kvadratični algoritmi, O(n^2), prezahtevni.

Pri algoritmih eksponentne zahtevnosti, npr. O(2^n), časovne zahteve naraščajo izjemno hitro. Recimo, da za $n = 20$ potrebujemo 1 s, tedaj za $n = 30$ potrebujemo 2^{10} s = 1024 s = 17 m; za $n = 40$ dobimo 1048576 s = 12.14 d; za $n = 50$ že 1073741824 s = 34.05 L; ... za $n = 80$ pa z 36.56 10^9 L presežemo domnevno starost Zemlje.

Pajek in velika omrežja

Glavni cilji v zasnovi programa Pajek so:

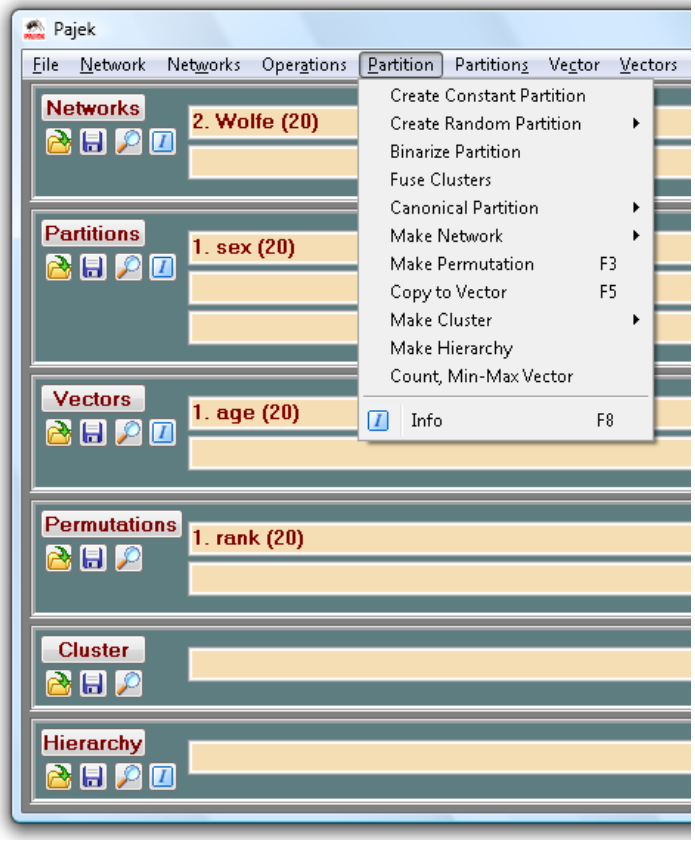


- podpora abstrakciji z (rekurzivnim) *razčlenjevanjem* velikega omrežja v več manjših, ki jih lahko v nadaljevanju posamično obdelamo z zahtevnejšimi postopki;
- ponuditi uporabniku izbor orodij za *prikaz* omrežij;
- izbrati/razviti nabor učinkovitih *podkvadratičnih* algoritmov za analizo velikih omrežij.

S Pajekom lahko: *določimo* zanimive skupine (komponente, soseščine 'pomembnih' vozlišč, sredice, ...) v omrežju, *izrežemo* del omrežja, ki pripada izbranim skupinam in ga ločeno *prikažemo*; lahko s 'povzetkom vpetosti' v celotno omrežje (lokalni pogled), *skrčimo* vozlišča, ki pripadajo isti skupini, in prikažemo odnose med skupinami (globalni pogled).

Vrste podatkov v Pajek u

Pri svojem delu se Pajek naslanja na 6 vrst podatkov:



- *omrežje*/network (graf),
- *razbitje*/partition (imenske ali urejenostne lastnosti vozlišč),
- *vektor*/vector (številske lastnosti vozlišč),
- *skupina*/cluster (podmnožica vozlišč),
- *urejenost*/permutation (urejenost vozlišč, urejenostna lastnost), in
- *hierarhija*/hierarchy (drevo nad vozlišči).

Novembra 2004 je bila vgrajena tudi podpora *večrelacijskim omrežjem* in *razbitjem povezav*.

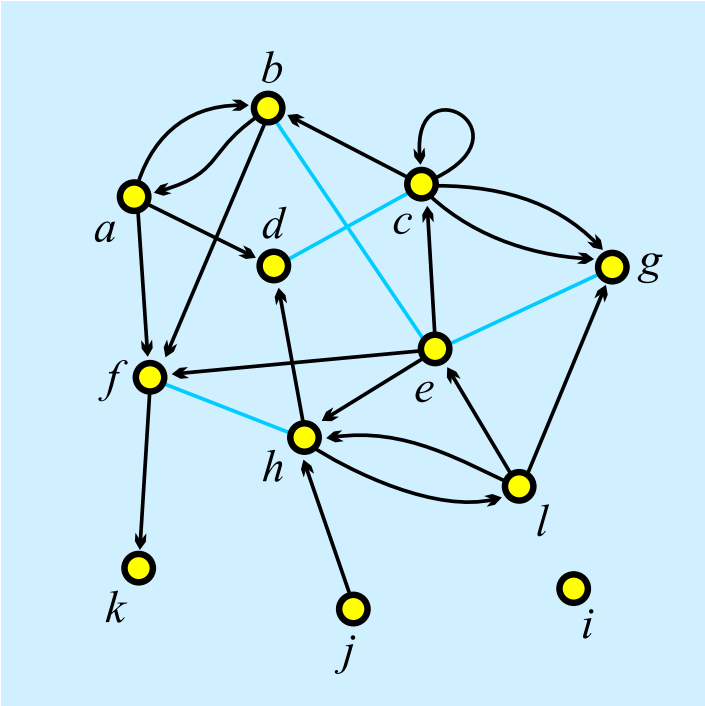
... Vrste podatkov v Pajek u

Moč programa Pajek temelji na velikem številu operacij, ki omogočajo različne pretvorbe med temi vrstami podatkov. Na njih je osnovana tudi zgradba uporabniškega vmesnika programa Pajek. Pajekovo *glavno okno* je nekakšno 'računalo' s seznamskimi 'registri' za vsako vrsto podatkov. Operacije se opravijo nad tekočimi (izbranimi) podatki v teh seznamih in vanje tudi vrnejo rezultate.

Operacije najdemo v izbirah glavnega okna. Razvrščene so glede na vrste podatkov, ki jih zahtevajo.

Pogosto uporabljana zaporedja operacij lahko združimo v *makro*je. To omogoča prilagoditve programa Pajek različnim skupinam uporabnikov (družboslovje, analiza besedil, rodoslovje, kemija, biologija, transport, računalništvo, matematika, ...) in posebnim obdelavam. Pajek podpira tudi *ponavljanje operacij* na zaporedjih omrežij.

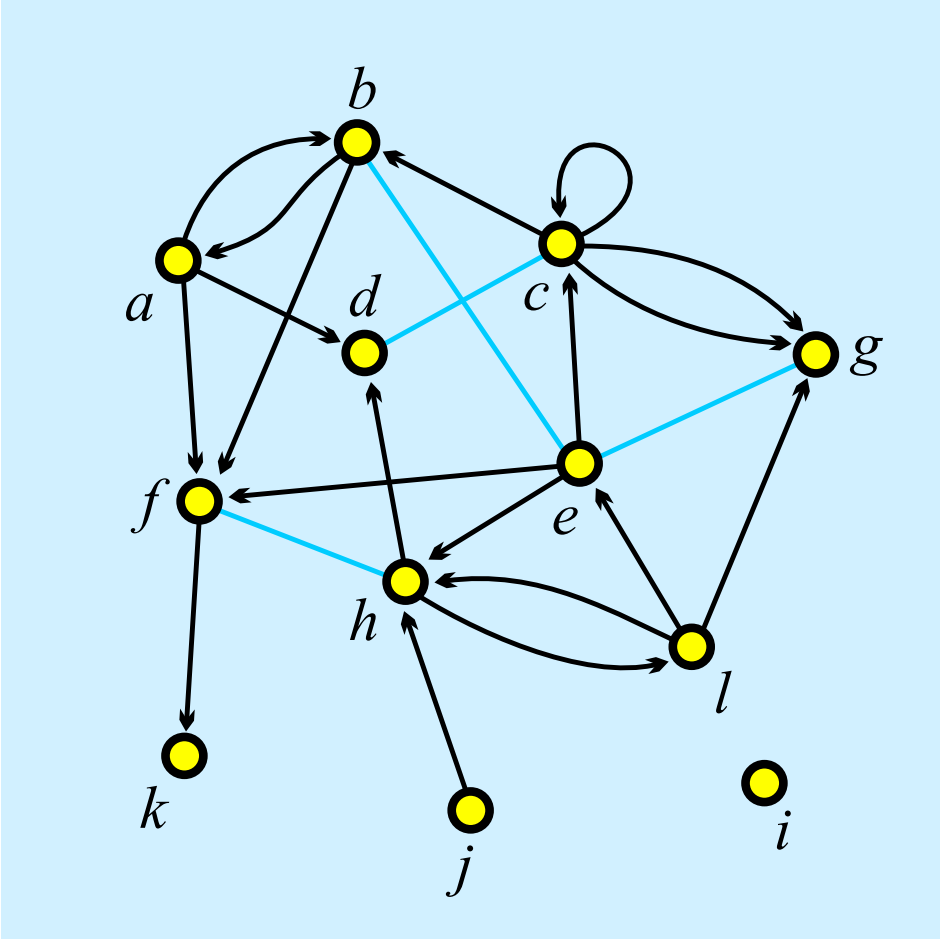
Opis grafa – množice / NET



$$
\begin{aligned}
\mathcal{V} &= \{a, b, c, d, e, f, g, h, i, j, k, l\} \\
\mathcal{A} &= \{(a, b), (a, d), (a, f), (b, a), \\
&\quad (b, f), (c, b), (c, c), (c, g), \\
&\quad (c, g), (e, c), (e, f), (e, h), \\
&\quad (f, k), (h, d), (h, l), (j, h), \\
&\quad (l, e), (l, g), (l, h)\} \\
\mathcal{E} &= \{(b\colon e), (c\colon d), (e\colon g), (f\colon h)\} \\
\mathcal{G} &= (\mathcal{V}, \mathcal{A}, \mathcal{E}) \\
\mathcal{L} &= \mathcal{A} \cup \mathcal{E}
\end{aligned}
$$

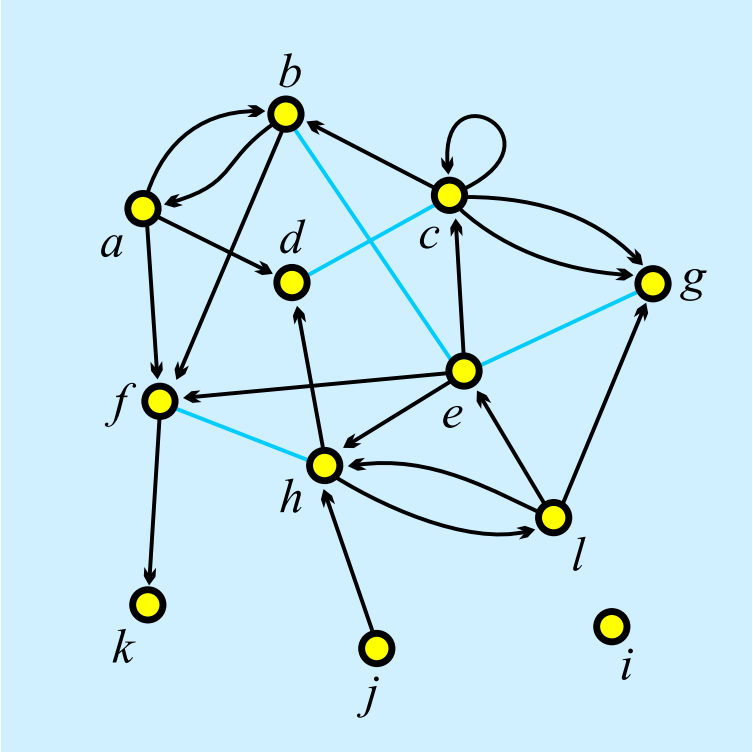
$\mathcal{A} = \emptyset$ – *neusmerjen* graf; $\mathcal{E} = \emptyset$ – *usmerjen* graf.
Pajek: GraphSet; TinaSet;
WWW: GraphSet / net; TinaSet / net, slika picture.

Graf – Množice / NET




```
*Vertices  12
  1 "a"     0.1020  0.3226
  2 "b"     0.2860  0.0876
  3 "c"     0.5322  0.2304
  4 "d"     0.3259  0.3917
  5 "e"     0.5543  0.4770
  6 "f"     0.1552  0.6406
  7 "g"     0.8293  0.3249
  8 "h"     0.4479  0.6866
  9 "i"     0.8204  0.8203
 10 "j"     0.4789  0.9055
 11 "k"     0.1175  0.9032
 12 "l"     0.7095  0.6475
*Arcs
  1  2
  2  1
  1  4
  1  6
  2  6
  3  2
  3  3
  3  7
  3  7
  5  3
  5  6
  5  8
  6 11
  8  4
 10  8
 12  5
 12  7
  8 12
 12  8
*Edges
  2  5
  3  4
  5  7
  6  8
```

Graf – Sosedi / NET



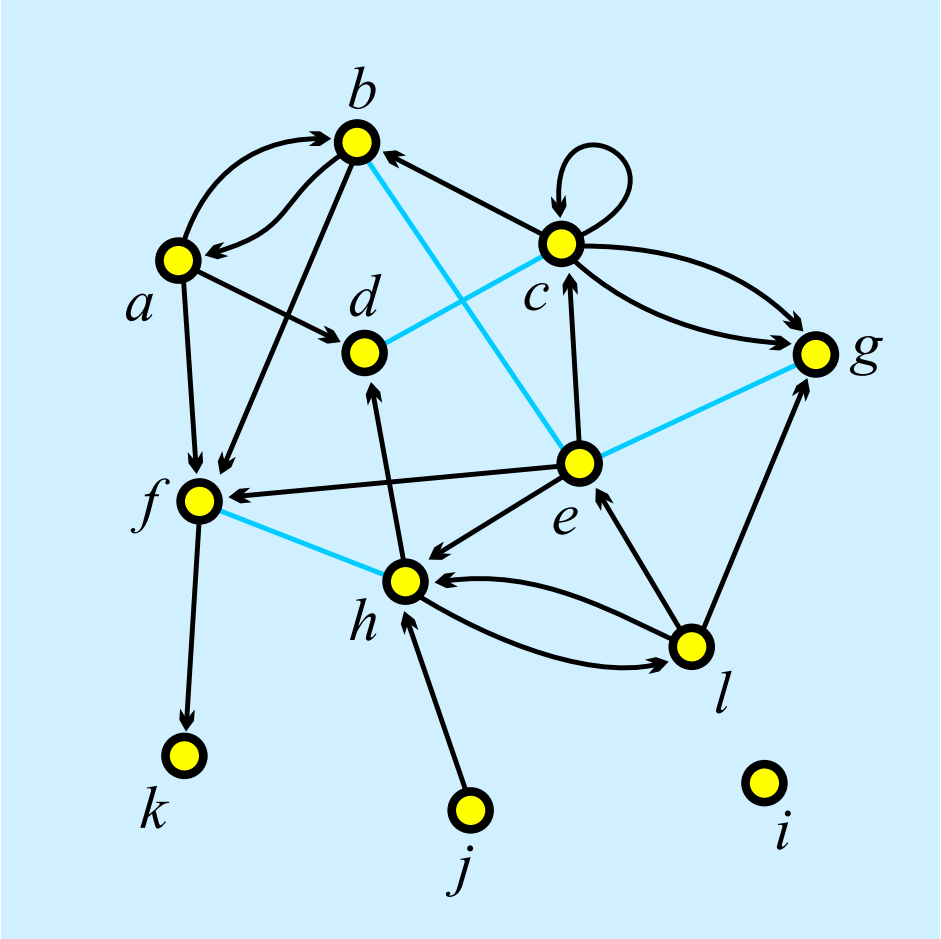
$N_{\mathcal{A}}(a) = \{b, d, f\}$
$N_{\mathcal{A}}(b) = \{a, f\}$
$N_{\mathcal{A}}(c) = \{b, c, g, g\}$
$N_{\mathcal{A}}(e) = \{c, f, h\}$
$N_{\mathcal{A}}(f) = \{k\}$
$N_{\mathcal{A}}(h) = \{d, l\}$
$N_{\mathcal{A}}(j) = \{h\}$
$N_{\mathcal{A}}(l) = \{e, g, h\}$

$N_{\mathcal{E}}(e) = \{b, g\}$
$N_{\mathcal{E}}(c) = \{d\}$
$N_{\mathcal{E}}(f) = \{h\}$

Pajek: GraphList; TinaList;

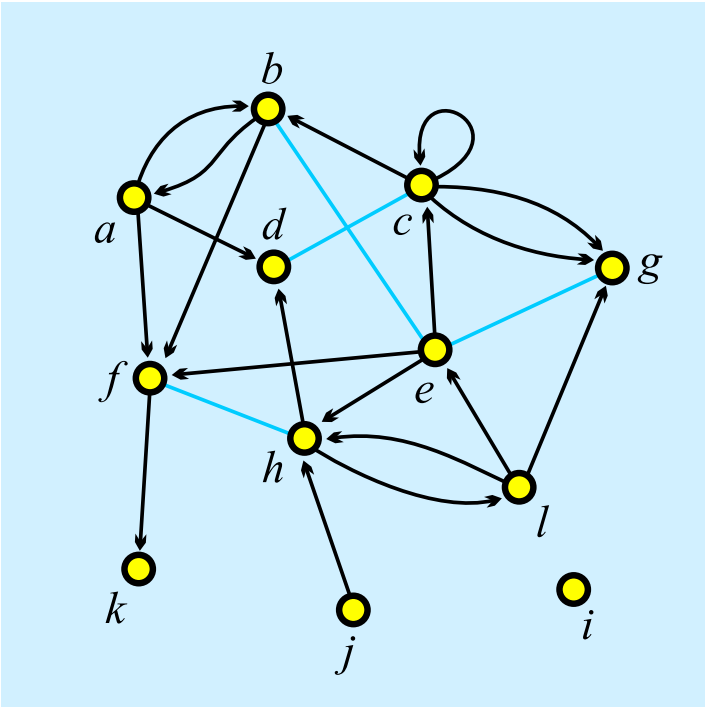
WWW: GraphList / net; TinaList / net.

Graph – Sosedi / NET




```
*Vertices  12
  1 "a"   0.1020  0.3226
  2 "b"   0.2860  0.0876
  3 "c"   0.5322  0.2304
  4 "d"   0.3259  0.3917
  5 "e"   0.5543  0.4770
  6 "f"   0.1552  0.6406
  7 "g"   0.8293  0.3249
  8 "h"   0.4479  0.6866
  9 "i"   0.8204  0.8203
 10 "j"   0.4789  0.9055
 11 "k"   0.1175  0.9032
 12 "l"   0.7095  0.6475
*Arcslist
  1  2  4  6
  2  1  6
  3  2  3  7  7
  5  3  6  8
  6 11
  8  4 12
 10  8
 12  5  7  8
*Edgeslist
  2  5
  3  4
  5  7
  6  8
```

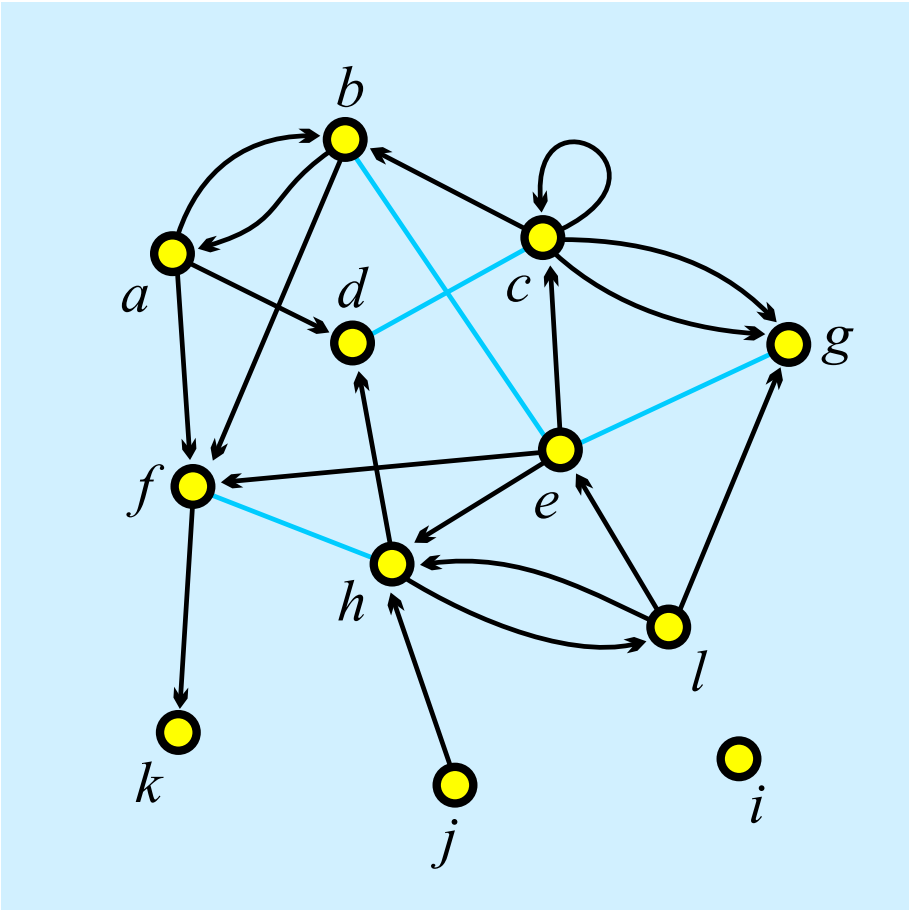
Graf – Matrika / NET



	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	0	1	0	1	0	0	0	0	0	0
b	1	0	0	0	1	1	0	0	0	0	0	0
c	0	1	1	1	0	0	2	0	0	0	0	0
d	0	0	1	0	0	0	0	0	0	0	0	0
e	0	1	1	0	0	1	1	1	0	0	0	0
f	0	0	0	0	0	0	0	1	0	0	1	0
g	0	0	0	0	1	0	0	0	0	0	0	0
h	0	0	0	1	0	1	0	0	0	0	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	1	0	0	0	0
k	0	0	0	0	0	0	0	0	0	0	0	0
l	0	0	0	0	1	0	1	1	0	0	0	0

Pajek: GraphMat; TinaMat, slika picture;
WWW: GraphMat / net; TinaMat / net, paj.
Graf G je *enostaven* ntk. vse vrednosti v matriki so 0 ali 1.

Graph – Matrika / NET




```
*Vertices  12
  1 "a"    0.1020  0.3226
  2 "b"    0.2860  0.0876
  3 "c"    0.5322  0.2304
  4 "d"    0.3259  0.3917
  5 "e"    0.5543  0.4770
  6 "f"    0.1552  0.6406
  7 "g"    0.8293  0.3249
  8 "h"    0.4479  0.6866
  9 "i"    0.8204  0.8203
 10 "j"    0.4789  0.9055
 11 "k"    0.1175  0.9032
 12 "l"    0.7095  0.6475
*Matrix
 0 1 0 1 0 1 0 0 0 0 0 0
 1 0 0 0 1 1 0 0 0 0 0 0
 0 1 1 1 0 0 2 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0
 0 1 1 0 0 1 1 1 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0
 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 1 0 1 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 1 1 0 0 0 0
```

Lastnosti vozlišč / CLU, VEC, PER

Vse tri vrste datotek imajo enako zgradbo:

`*vertices` n	n število vozlišč
v_1	vozlišče 1 ima vrednost v_1
...	
v_n	

CLUstering – razbitje vozlišč – *imenska* ali *urejenostna* lastnost vozlišč
$v_i \in \mathbb{N}$: vozlišče i pripada skupini v_i;
VECtor – *številska* lastnost vozlišč
$v_i \in \mathbb{R}$: lastnost ima vrednost v_i na vozlišču i;
PERmutation – *urejenost* vozlišč
$v_i \in \mathbb{N}$: vozlišče i je na v_i-tem mestu.

Ko zbiramo podatke o omrežjih, ne pozabimo zajeti tudi čimveč lastnosti in uteži.

Primer: Wolfe Monkey Data

inter.net

```
*Vertices  20
  1 "m01"
  2 "m02"
  3 "m03"
  4 "m04"
  5 "m05"
  6 "f06"
  7 "f07"
  8 "f08"
  9 "f09"
 10 "f10"
 11 "f11"
 12 "f12"
 13 "f13"
 14 "f14"
 15 "f15"
 16 "f16"
 17 "f17"
 18 "f18"
 19 "f19"
 20 "f20"
*Edges
  1   2   2
  1   3  10
  1   4   4
```

inter.net

```
  1   6   5
  1   7   9
  1   8   7
  1   9   4
  1  10   3
  1  11   3
  1  12   7
  1  13   3
  1  14   2
  1  15   5
  1  16   1
  1  17   4
  1  18   1
  2   3   5
  2   4   1
  2   5   3
  2   6   1
  2   7   4
  2   8   2
  2   9   6
  2  10   2
  2  11   5
  2  12   4
  2  13   3
  2  14   2

...
```

sex.clu

```
*vertices 20
1
1
1
1
1
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
```

age.vec

```
*vertices 20
15
10
10
 8
 7
15
 5
11
 8
 9
16
10
14
 5
 7
11
 7
 5
15
 4
```

rank.per

```
*vertices 20
 1
 2
 3
 4
 5
10
11
 6
12
 9
 7
 8
18
19
20
13
14
15
16
17
```

Pozor: 0 ni dovoljena številka vozlišča.

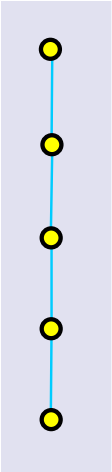
Pajekova projektna datoteka / PAJ

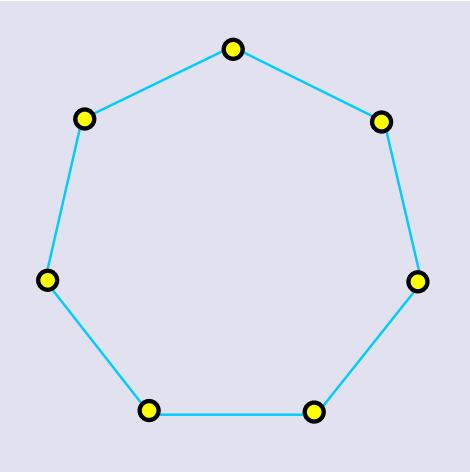
Vse vrste podatkov o omrežju lahko združimo v eno samo datoteko – Pajekovo *projektno* datoteko *file*`.paj`.
Najenostavneje to naredimo tako:

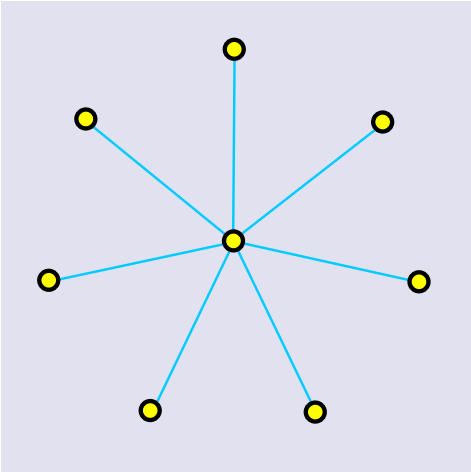
- preberite v program `Pajek` vse podatkovne datoteke,
- izračunajte še morebitne dodatne podatke,
- odstranite (`dispose`) morebitne odvečne podatke,
- shranite vse skupaj na projektno datoteko z `File/Project file/Save`.

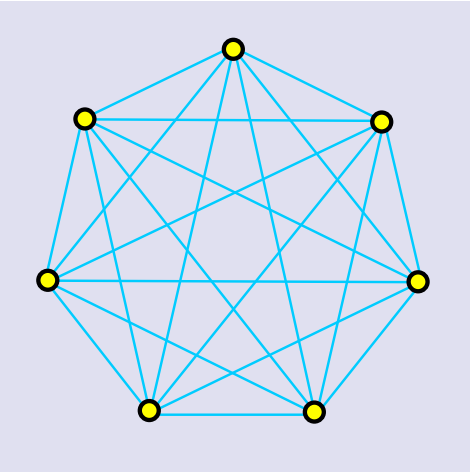
Naslednjič lahko obnovite stanje z eno samo zahtevo `File/Project file/Read`.
Wolfe-vo omrežje kot `Pajekova` projektna datoteka (PDF/paj).

Posebni grafi – pot, cikel, zvezda, polni









Na slikah so zaporedoma prikazani grafi: *pot* P_5, *cikel* C_7, *zvezda* S_8 in *polni graf* K_7.

Prikazi omrežij

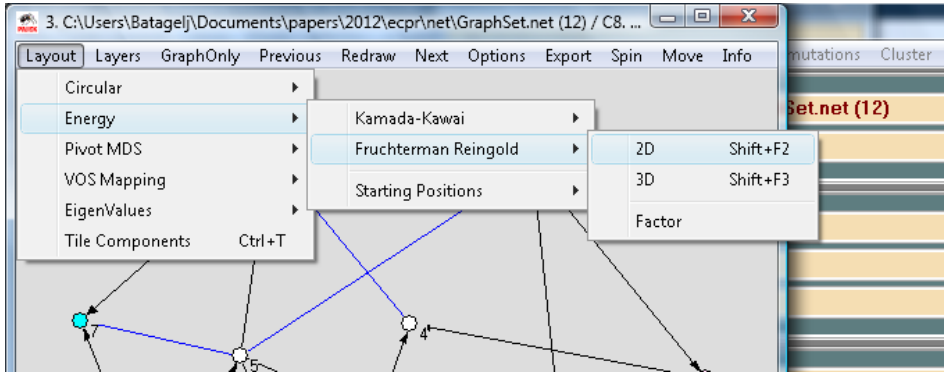
Ne prevelika omrežja

`Options/Read-Write [Large Network (Vertices)]`

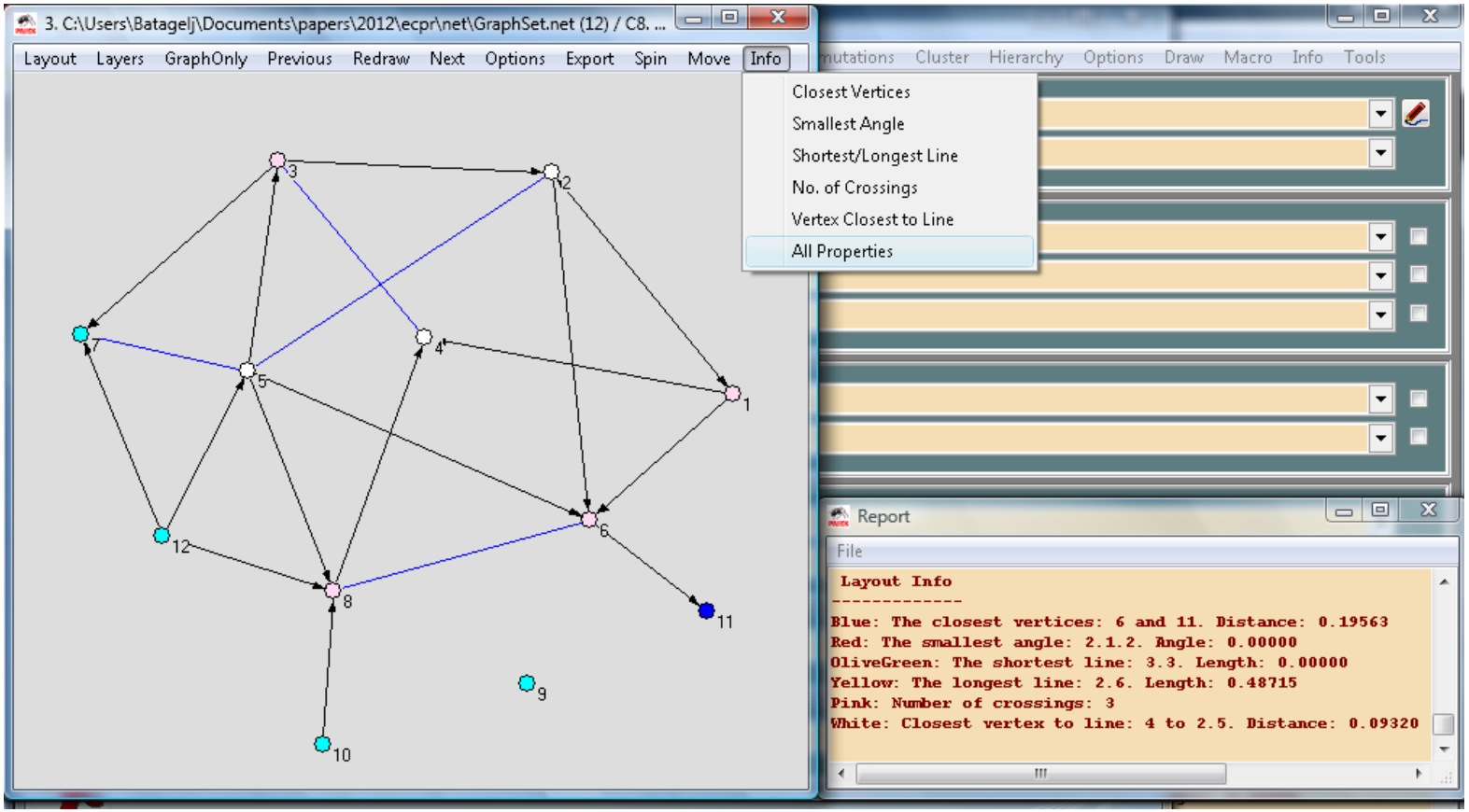
lahko narišemo v *prikaznem oknu* z ukazi iz izbire `Draw`, npr. `Draw/Network`.

Če omrežje nima določenih koordinat vozlišč, ga `Pajek` nariše *krožno*. Sliko lahko spreminjamo s premikanjem vozlišč z miško.

Za večja omrežja uporabimo postopke za prikaz omrežij, ki jih najdemo v izbiri `[Draw] Layout`.



Lepe risbe


3. C:\Users\Batagelj\Documents\papers\2012\ecpr\net\GraphSet.net (12) / C8. ...
Layout Layers GraphOnly Previous Redraw Next Options Export Spin Move Info
Closest Vertices
Smallest Angle
Shortest/Longest Line
No. of Crossings
Vertex Closest to Line
All Properties
Cluster Hierarchy Options Draw Macro Info Tools
Report
File
Layout Info
Blue: The closest vertices: 6 and 11. Distance: 0.19563
Red: The smallest angle: 2.1.2. Angle: 0.00000
OliveGreen: The shortest line: 3.3. Length: 0.00000
Yellow: The longest line: 2.6. Length: 0.48715
Pink: Number of crossings: 3
White: Closest vertex to line: 4 to 2.5. Distance: 0.09320

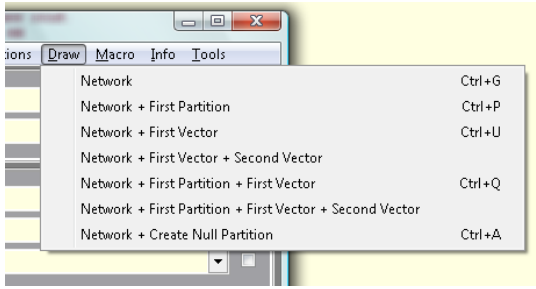
Prikazi lastnosti

Za omrežje $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ so lastnosti vozlišč $\mathcal{P}$ in povezav $\mathcal{W}$ lahko merjene v različnih lestvicah. Lahko so *vnešene* z ostalimi podatki o omrežju, ali pa *izračunane* pri analizi omrežja.

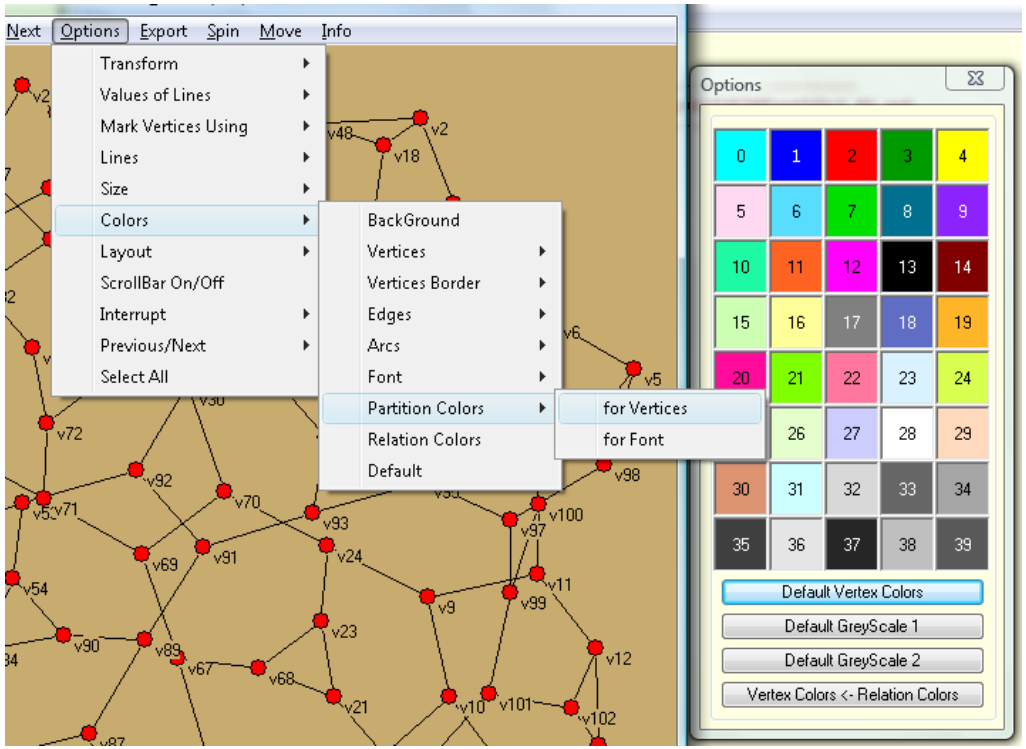
V programu `Pajek` podamo posamezno številsko lastnost vozlišč kot *vektor*, imensko lastnost pa ali kot *razbitje* ali kot *oznako* vozlišč. Na sliki lahko številsko lastnost prikažemo kot *velikost*(i) vozlišča ali njegovo *koordinato* ali velikost imena/oznake; imensko lastnost pa kot *barvo* ali *obliko* lika, ali kot *oznako* vozlišča, ali kot barvo oznake.

Vrednosti povezav so številske. Na sliki jih prikažemo z izpisom *vrednosti*, *debelino* črte ali *sivino*. Imenske vrednosti lahko dodamo v opisu omrežja kot oznake, barvo ali vzorec (glejte priročnik programa `Pajek`, razdelek **4.3**).

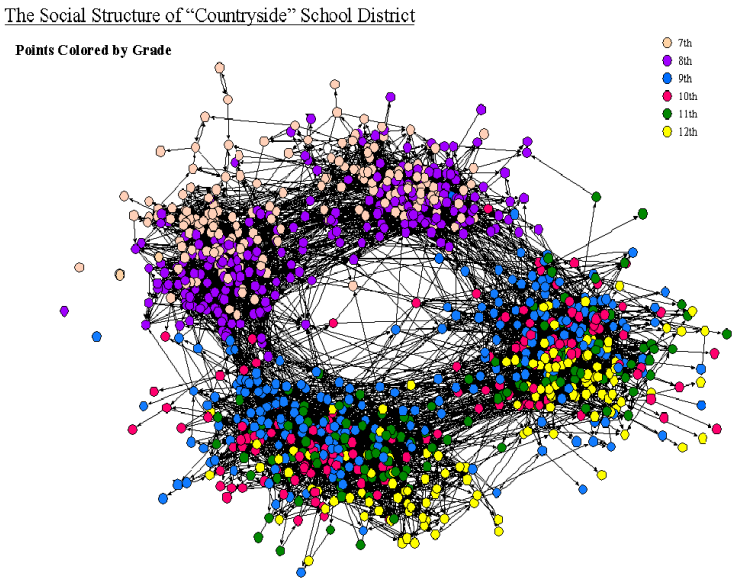
Nekaj ukazov v programu Pajek

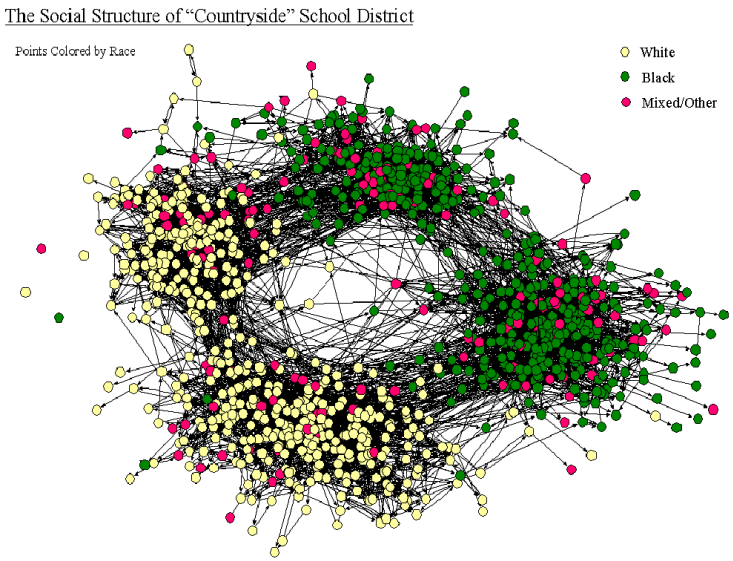



```
Operations/Network+Vector/Transform/Put Coordi
Network/Create Vector/Get Coordinate
[Draw] Options
[Draw] Layout/Energy/Kamada-Kawai/Free
[Draw] Export/2D/EPS-PS
```



Prikaz lastnosti – šolarji (Moody)





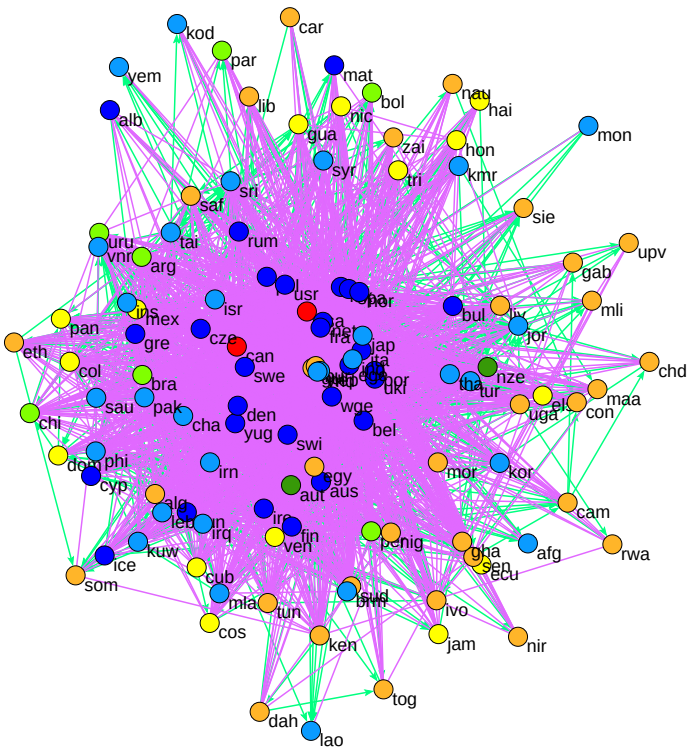
Primer: Snyder in Kick-ovo omrežje svetovne trgovine

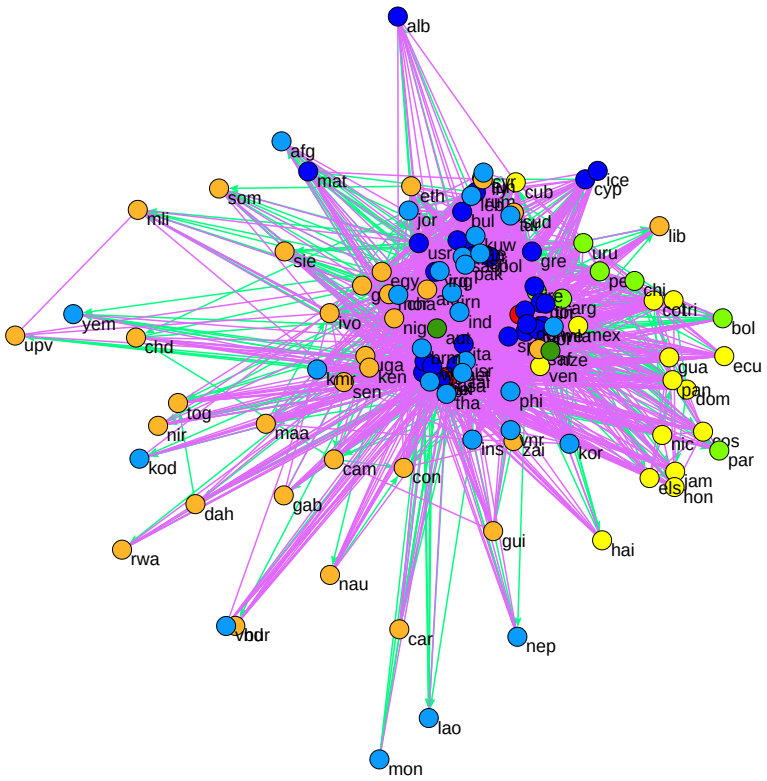
Omrežje je dostopno kot `Pajek`-ova projektna datoteka `SaKtrade.paj`

Omrežje opisuje trgovanje med državami (118 vozlišč, 515 usmerjenih in 2116 neusmerjenih povezav). Omrežje je podrobneje opisano v članku: Snyder, David and Edward Kick (1979). *The World System and World Trade: An Empirical Exploration of Conceptual Conflicts*, Sociological Quaterly, 20,1, 23-36.

Za omrežje je podano tudi razbitje na (pod)kontinente: 1 - Evropa, 2 - Severna Amerika, 3 - Srednja Amerika, 4 - Južna Amerika, 5 - Azija, 6 - Afrika, 7 - Oceanija.
Novejše podatke je mogoče dobiti na NBER / Feenstra; in primer analize na Science / Hidalgo et. al.

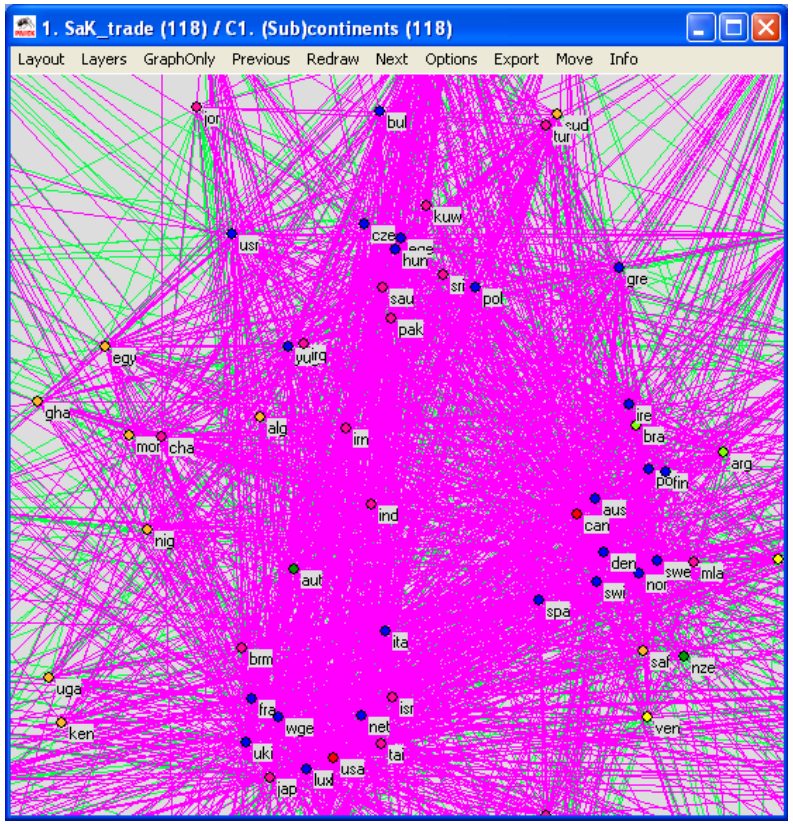
Draw / Partition





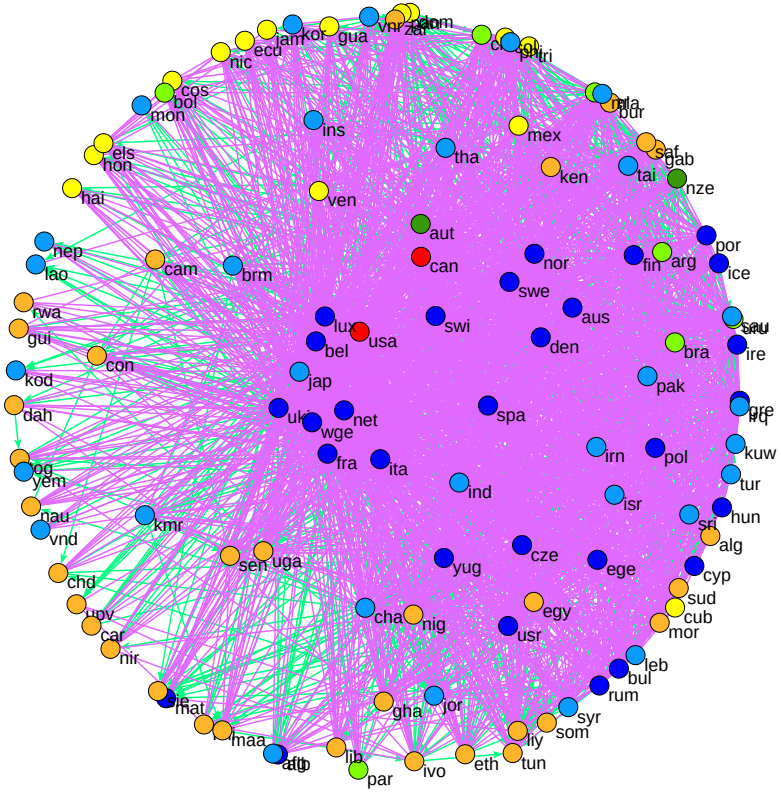
Draw/Network + First Partition
Layout/Energy/Kamada-Kawai/Free
Layout/Energy/Fruchterman Reingold/2D

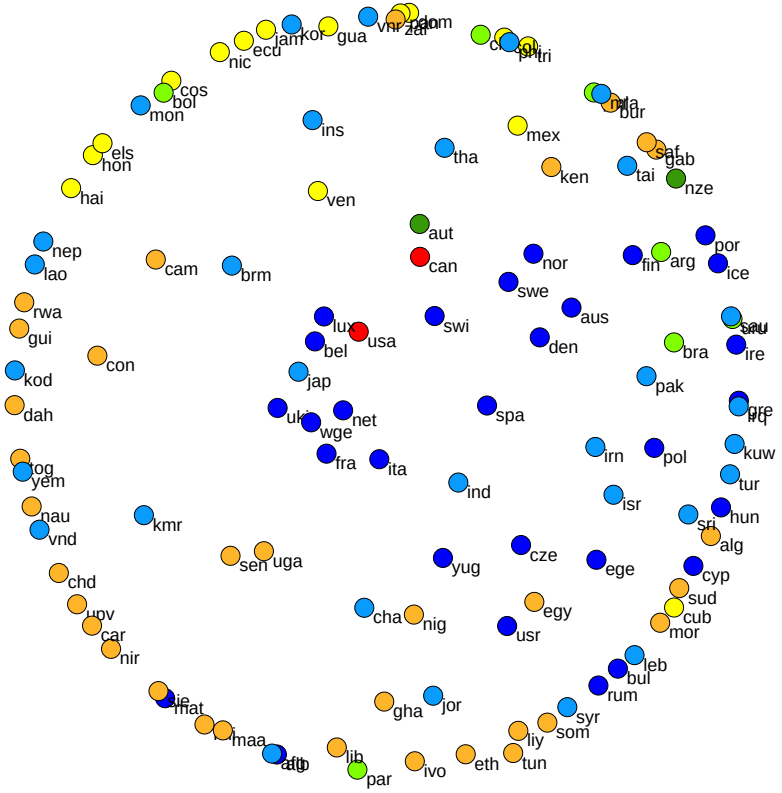
Povečave



Z desnim gumbom na miški izberite območje povečave. Običajni prikaz obnovite z `Redraw`.

Fruchterman Reingold / factor = 9



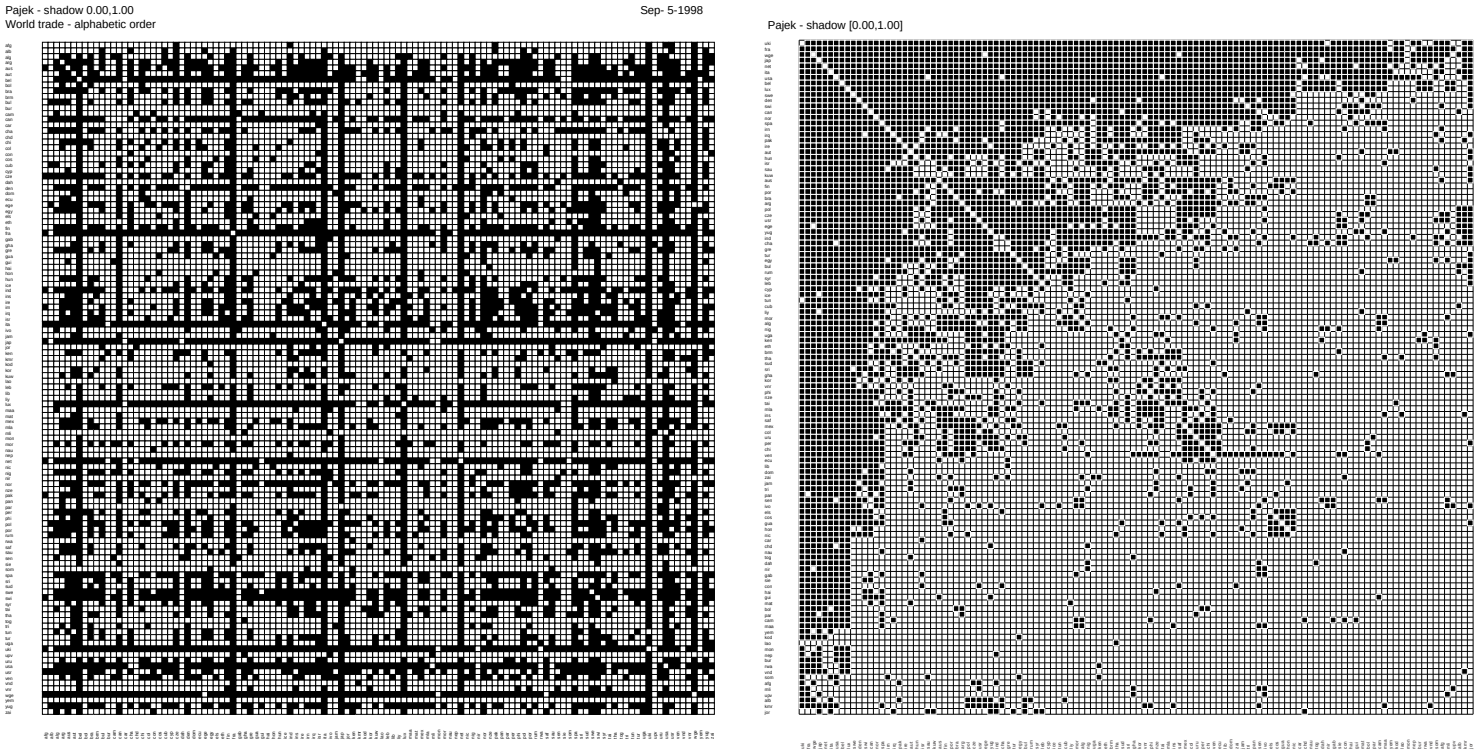


`Layout/Energy/Fruchterman Reingold/3D`

3D picture / King

Matrični prikaz

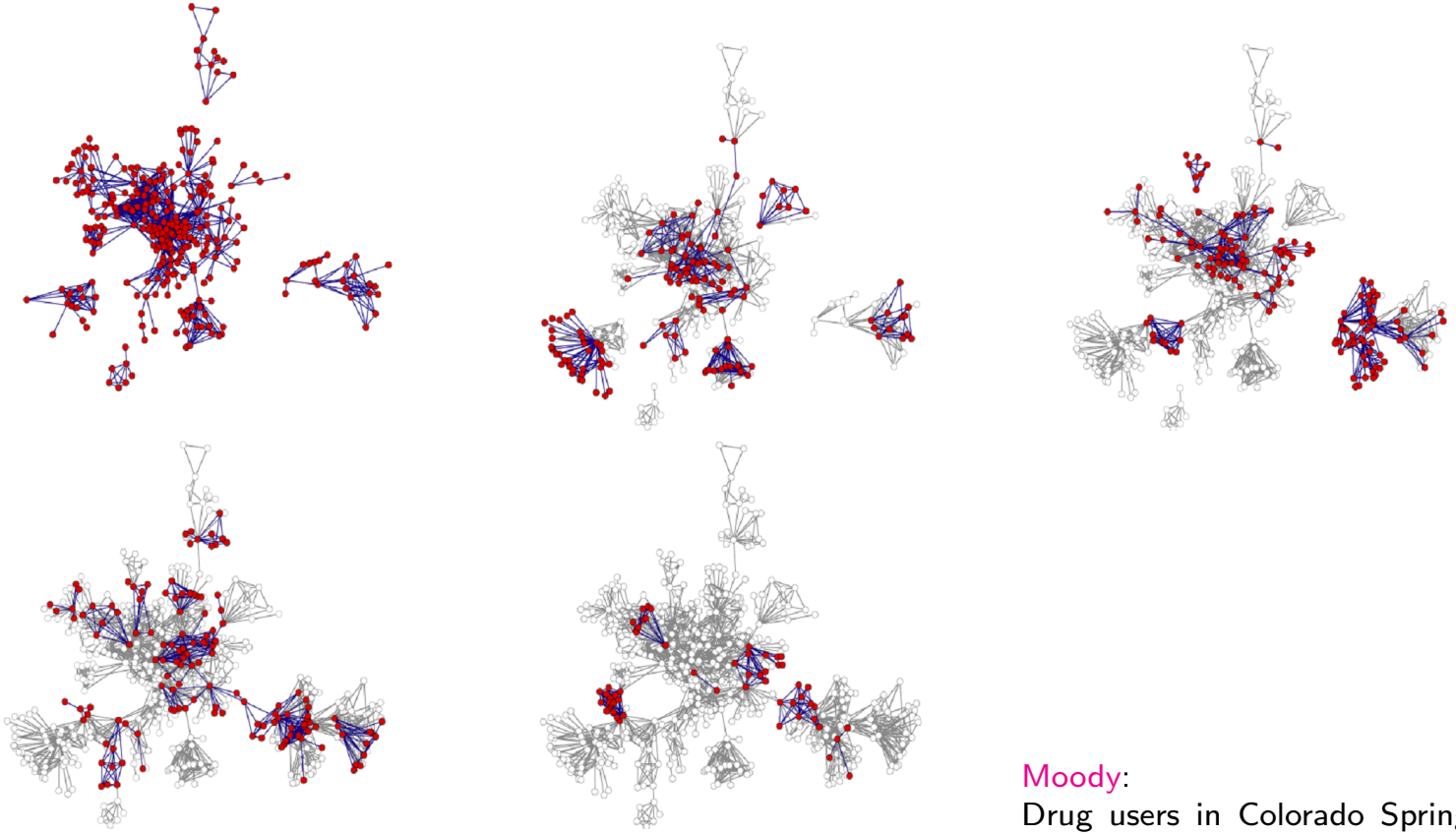
Snyder & Kick's World trade network / $n = 118$, $m = 514$



Abecedni vrstni red (levo) in preureditev (desno)

Časovna omrežja

V časovnem omrežju se prisotnost vozlišča/povezave v omrežju spreminja skozi čas. Pajek ponuja dva načina opisa časovnih omrežij *prisotnostni* in *dogodkovni*.



Moody:
Drug users in Colorado Springs, 5 let

Časovna omrežja

Časovno omrežje

$$\mathcal{N}_T = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W}, T)$$

dobimo, če navadnemu omrežju dodamo *čas* T . T je množica *časovnih točk* ali *trenutkov* $t \in T$.

V časovnem omrežju vozlišča $v \in \mathcal{V}$ in povezave $l \in \mathcal{L}$ niso nujno vseskozi dejavna/prisotna. Če je povezava $l(u, v)$ dejavna v trenutku t, morata biti dejavni v t tudi njeni krajišči u in v.

Omrežje sestavljeno iz vozlišč in povezav dejavnih v trenutku $t \in T$ bomo označevali z $\mathcal{N}(t)$ in mu rekli *časovna rezina* v trenutku t. V programu `Pajek` dobimo zaporedje časovnih rezin z zahtevo
`Network/Temporal Network/Generate in time`

Časovna omrežja – prisotnostni opis

```
*Vertices 3
1 "a" [5-10,12-14]
2 "b" [1-3,7]
3 "e" [4-*]
*Edges
1 2 1 [7]
1 3 1 [6-8]
```

Podatke o obdobjih prisotnosti/ dejavnosti posameznega vozlišča/ povezave navedemo na koncu pripadajoče vrstice v oglatih oklepajih [in]. Obdobja so oštevilčena od 1 naprej. Pri naštevanju jih ločimo z vejico ,. Zaporedna obdobja od začetka z do konca k lahko krajše zapišemo z-k. Znak $*$ pomeni neskončno. Vozlišče a je prisotno v obdobjih 5 do 10 in 12 do 14. Povezava $(1:3)$ je prisotna v obdobjih 6 do 8.

Povezava je prisotna, če sta prisotni obe njeni krajišči.

Time.net.

Opis časovnih omrežij uporabljen v programu Pajek je omejen – spreminjajo se lahko tudi vrednosti lastnosti in uteži. V razvoju je poseben program Nets / TQ, ki bo omogočal opis in analizo splošnejših časovnih omrežij. 1, 2

Časovna omrežja – dogodkovni opis

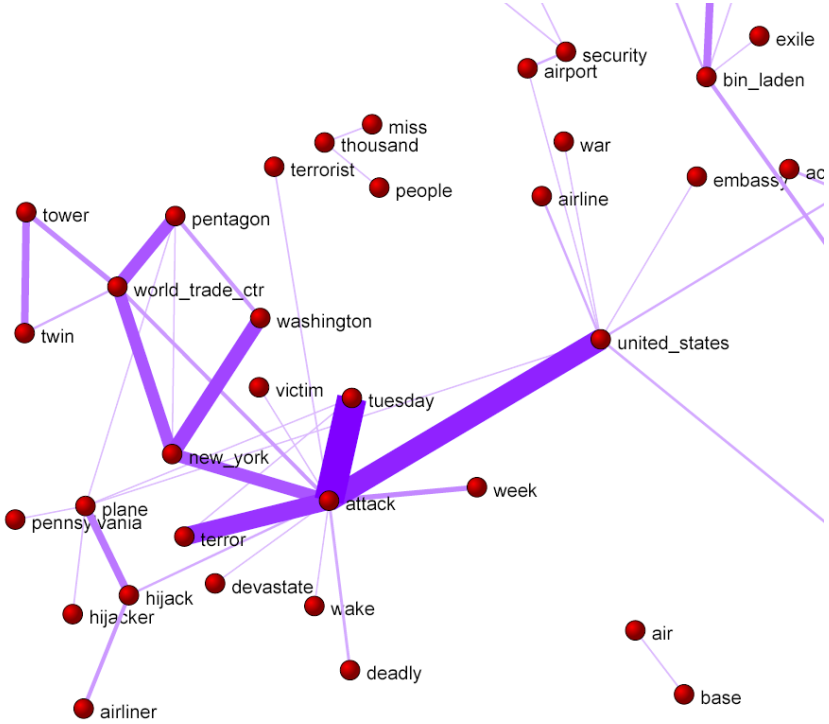
Event	Explanation
TI *t*	initial events – following events happen when time point *t* starts
TE *t*	end events – following events happen when time point *t* is finished
AV *vns*	add vertex *v* with label *n* and properties *s*
HV *v*	hide vertex *v*
SV *v*	show vertex *v*
DV *v*	delete vertex *v*
AA *uvs*	add arc *(u,v)* with properties *s*
HA *uv*	hide arc *(u,v)*
SA *uv*	show arc *(u,v)*
DA *uv*	delete arc *(u,v)*
AE *uvs*	add edge *(u:v)* with properties *s*
HE *uv*	hide edge *(u:v)*
SE *uv*	show edge *(u:v)*
DE *uv*	delete edge *(u:v)*
CV *vs*	change property of vertex *v* to *s*
CA *uvs*	change property of arc *(u,v)* to *s*
CE *uvs*	change property of edge *(u:v)* to *s*
CT *uv*	change (un)directedness of line *(u,v)*
CD *uv*	change direction of arc *(u,v)*
PE *uvs*	replace pair of arcs *(u,v)* and *(v,u)* by single edge *(u:v)* with properties *s*
AP *uvs*	add pair of arcs *(u,v)* and *(v,u)* with properties *s*
DP *uv*	delete pair of arcs *(u,v)* and *(v,u)*
EP *uvs*	replace edge *(u:v)* by pair of arcs *(u,v)* and *(v,u)* with properties *s*

s je lahko prazen.
Pri vzporednih povezavah *:k* označuje *k*-to povezavo – `HE:3 14 37` skrije tretjo neusmerjeno povezavo med vozliščema 14 in 37.

```
*Vertices 3
*Events
TI 1
AV 2 "b"
TE 3
HV 2
TI 4
AV 3 "e"
TI 5
AV 1 "a"
TI 6
AE 1 3 1
TI 7
SV 2
AE 1 2 1
TE 7
DE 1 2
DV 2
TE 8
DE 1 3
TE 10
HV 1
TI 12
SV 1
TE 14
DV 1
```

`Time.tim`. `Friends.tim`.

Časovna omrežja / 11. september



Slike v SVG: **66 dni**.

V časovnih omrežjih posamezna vozlišča in povezave niso nujno ves čas dejavne/prisotne. Steve Corman s sodelavci z Arizona State University je s Centering Resonance Analysis (**CRA**) predelal dnevne Reutersove novice (66 dni) o 11. septembru v časovno omrežje sopojavljanja besed.

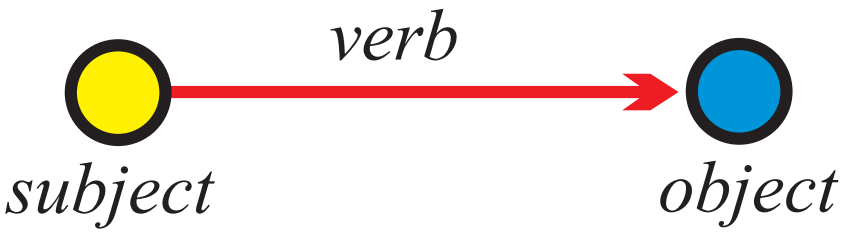
Večkratna omrežja

V *večkratnem* ali *večrelacijskem* omrežju

$$\mathcal{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$$

je množica povezav $\mathcal{L}$ razbita na podmnožice (*relacije*) $\mathcal{L}_i$.

Večrelicijska omrežja omogočajo prevedbo besedil na osnovi enostavnih stavkov S-V-O (*Subject-Verb-Object*) in njih izboljšav.



Subjects ∪ *Objects* določajo vozlišča, *Verbs* pa relacije.

Primeri: Roberto Franzosi; *KEDS*, *Tabari*, KEDS / Gulf.
Glej še trojice RDF na pomenskem spletu in SPARQL.

... Večkratna omrežja

V Pajeku jih lahko zapišemo na dva načina:

- geslom, ki napovedujejo opis povezav, (`*arcs`, `*edges`, `*arcslist`, `*edgeslist`, `*matrix`) dodamo številko relacije in lahko tudi njeno ime. Npr.

  ```
  *arcslist :3 "posojanje gradiv"
  ```

  Vse geslu podrejene povezave pripadajo navedeni relaciji. (Sampson, SampsonL)

- Med povezavami, podrejenimi gesloma `*arcs` ali `*edges`, lahko posamezno povezavo pripišemo izbrani relaciji, tako da njen opis začnemo s številko relacije

  ```
  3:  47 14 5
  ```

  Povezava s krajiščema 47 in 14 ter utežjo 5 pripada relaciji 3.

Večrelacijsko časovno omrežje – KEDS/WEIS

```
% Recoded by WEISmonths, Sun Nov 28 21:57:00 2004
% from http://www.ku.edu/~keds/data.dir/balk.html
*vertices 325
1 "AFG" [1-*]
2 "AFR" [1-*]
3 "ALB" [1-*]
4 "ALBMED" [1-*]
5 "ALG" [1-*]
 ...
318 "YUGGOV" [1-*]
319 "YUGMAC" [1-*]
320 "YUGMED" [1-*]
321 "YUGMTN" [1-*]
322 "YUGSER" [1-*]
323 "ZAI" [1-*]
324 "ZAM" [1-*]
325 "ZIM" [1-*]
*arcs :0 "*** ABANDONED"
*arcs :10 "YIELD"
*arcs :11 "SURRENDER"
*arcs :12 "RETREAT"
 ...
*arcs :223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
*arcs
224: 314 153 1 [4]                890402  YUG     KSV     224  (RIOT)  RIOT-TORN
212: 314 83 1 [4]                 890404  YUG     ETHALB  212  (ARREST PERSON) ALB ETHNIC JAILED
224: 3 83 1 [4]                   890407  ALB     ETHALB  224  (RIOT)  RIOTS
123: 83 153 1 [4]                 890408  ETHALB  KSV     123  (INVESTIGATE)   PROBING
 ...
42: 105 63 1 [175]                030731  GER     CYP     042  (ENDORSE)       GAVE SUPPORT
212: 295 35 1 [175]               030731  UNWCT   BOSSER  212  (ARREST PERSON) SENTENCED TO PRIS
43: 306 87 1 [175]                030731  VAT     EUR     043  (RALLY) RALLIED
13: 295 35 1 [175]                030731  UNWCT   BOSSER  013  (RETRACT)       CLEARED
121: 295 22 1 [175]               030731  UNWCT   BAL     121  (CRITICIZE)     CHARGES
122: 246 295 1 [175]              030731  SER     UNWCT   122  (DENIGRATE)     TESTIFIED
121: 35 295 1 [175]               030731  BOSSER  UNWCT   121  (CRITICIZE)     ACCUSED
```

Kansas Event Data System *KEDS*

Dvovrstna omrežja

V *dvovrstnem* omrežju $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2), \mathcal{L}, \mathcal{P}, \mathcal{W})$ je množica vozlišč sestavljena iz dveh ločenih množic vozlišč $\mathcal{V}_1$ in $\mathcal{V}_2$, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, povezave iz množice $\mathcal{L}$ pa imajo eno krajišče v $\mathcal{V}_1$ drugo pa v $\mathcal{V}_2$. Običajno je znana tudi *utež* $w : \mathcal{L} \to \mathbb{R} \in \mathcal{W}$; če ni, privzamemo $w(u, v) = 1$ za vse povezave $(u, v) \in \mathcal{L}$.
Dvovrstno omrežje lahko opišemo tudi s pravokotno matriko $\mathbf{A} = [a_{uv}]_{\mathcal{V}_1 \times \mathcal{V}_2}$.

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

Primeri: (članki, avtorji), (ljudje, društva, leta članstva), (kupci, dobrine, količina), (poslanci, vprašanje, pozitivni glas), (ljudje, revije, branost).
Dvovrstno omrežje napovemo z `*vertices` n $n_{\mathcal{U}}$.
Avtorji in dela.

Južnakinje

Najbolj znan primer dvovrstnega omrežja so Davisove južnakinje.
`Davis.paj`. Freemanov pregled.

Names of Participants in Group I	Code Numbers and Dates of Social Events Reported in *Old City Herald*													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson	×	×	×	×	×	×	.	×	×	.	.	.	.	.
2. Miss Laura Mandeville	×	×	×	.	×	×	×	×	.	.	.	.	.	.
3. Miss Theresa Anderson	.	×	×	×	×	×	×	×	×	.	.	.	.	.
4. Miss Brenda Rogers	×	.	×	×	×	×	×	×	.	.	.	.	.	.
5. Miss Charlotte McDowd	.	.	×	×	×	.	×	.	.	.	.	.	.	.
6. Miss Frances Anderson	.	.	×	.	×	×	.	×	.	.	.	.	.	.
7. Miss Eleanor Nye	.	.	.	.	×	×	×	×	.	.	.	.	.	.
8. Miss Pearl Oglethorpe	.	.	.	.	.	×	.	×	×	.	.	.	.	.
9. Miss Ruth DeSand	.	.	.	.	×	.	×	×	×	.	.	.	.	.
10. Miss Verne Sanderson	.	.	.	.	.	.	×	×	×	.	.	×	.	.
11. Miss Myra Liddell	.	.	.	.	.	.	.	×	×	×	.	×	.	.
12. Miss Katherine Rogers	.	.	.	.	.	.	.	×	×	×	.	×	×	×
13. Mrs. Sylvia Avondale	.	.	.	.	.	.	×	×	×	×	.	×	×	×
14. Mrs. Nora Fayette	.	.	.	.	.	×	×	.	×	×	×	×	×	×
15. Mrs. Helen Lloyd	.	.	.	.	.	.	×	×	.	×	×	×	.	.
16. Mrs. Dorothy Murchison	.	.	.	.	.	.	.	×	×	.	.	.	.	.
17. Mrs. Olivia Carleton	.	.	.	.	.	.	.	.	×	.	×	.	.	.
18. Mrs. Flora Price	.	.	.	.	.	.	.	.	×	.	×	.	.	.

Večvrstna omrežja in povezana omrežja

V *večvrstnem* omrežju $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_k), \mathcal{L}, \mathcal{P}, \mathcal{W})$ je množica vozlišč sestavljena iz več ločenih množic *vrst* vozlišč $\mathcal{V}_i$, $i \in 1 : k$.

Tako omrežje lahko razstavimo v zbirko *povezanih* enovrstnih

$$\mathcal{N}_j = (\mathcal{V}_j, \mathcal{L}_j, \mathcal{P}_j, \mathcal{W}_j)$$

in dvovrstnih omrežij

$$\mathcal{N}_{i,j} = ((\mathcal{V}_i, \mathcal{V}_j), \mathcal{L}_{i,j}, \mathcal{P}_{i,j}, \mathcal{W}_{i,j}), \qquad i < j$$

Omrežja in razpredelnice

Kako opisati omrežje $\mathcal{N}$? Odgovor je, kot vemo, preprost – podati moramo množice $\mathcal{V}$, $\mathcal{L}$, $\mathcal{P}$, and $\mathcal{W}$.

Pogosto lahko to storimo tako, da omrežje $\mathcal{N}$ podamo s tabelama $(\mathcal{V}, \mathcal{P})$ in $(\mathcal{L}, \mathcal{W})$.

Kot primer opišimo omrežje, ki ga določajo naslednja dela: Generalized blockmodeling, Clustering with relational constraint, Partitioning signed social networks, The Strength of Weak Ties

Imamo več vrst vozlišč: osebe, članke, knjige, zbirke, revije, založnike; in različne odnose (relacije) med njimi: author_of, editor_of, contained_in, cites, published_by.

Tabeli lahko pripravimo in vzdržujemo v razpredelniškem programu (npr. Excel). Shranimo jih lahko v znakovni obliki CSV (Comma Separated Values).

bibNodes.csv

```
name;mode;country;sex;year;vol;num;fPage;lPage;x;y
"Batagelj, Vladimir";person;SI;m;;;;;;809.1;653.7
"Doreian, Patrick";person;US;m;;;;;;358.5;679.1
"Ferligoj, Anuška";person;SI;f;;;;;;619.5;680.7
"Granovetter, Mark";person;US;m;;;;;;145.6;660.5
"Moustaki, Irini";person;UK;f;;;;;;783.0;228.0
"Mrvar, Andrej";person;SI;m;;;;;;478.0;630.1
"Clustering with relational constraint";paper;;;1982;47;;413;426;684.1;3
"The Strength of Weak Ties";paper;;;1973;78;6;1360;1380;111.3;329.4
"Partitioning signed social networks";paper;;;2009;31;1;1;11;408.0;337.8
"Generalized Blockmodeling";book;;;2005;24;;1;385;533.0;445.9
"Psychometrika";journal;;;;;;;;741.8;086.1
"Social Networks";journal;;;;;;;;321.4;236.5
"The American Journal of Sociology";journal;;;;;;;;111.3;168.9
"Structural Analysis in the Social Sciences";series;;;;;;;;310.4;082.8
"Cambridge University Press";publisher;UK;;;;;;;534.3;238.2
"Springer";publisher;US;;;;;;;884.6;174.0
```

bibNodes.csv

V velikih omrežjih, v izogib praznim celicam, omrežje razcepimo na podomrežja – zbirka omrežij.

bibLinks.csv

```
from;relation;to
"Batagelj, Vladimir";authorOf;"Generalized Blockmodeling"
"Doreian, Patrick";authorOf;"Generalized Blockmodeling"
"Ferligoj, Anuška";authorOf;"Generalized Blockmodeling"
"Batagelj, Vladimir";authorOf;"Clustering with relational constraint"
"Ferligoj, Anuška";authorOf;"Clustering with relational constraint"
"Granovetter, Mark";authorOf;"The Strength of Weak Ties"
"Granovetter, Mark";editorOf;"Structural Analysis in the Social Sciences
"Doreian, Patrick";authorOf;"Partitioning signed social networks"
"Mrvar, Andrej";authorOf;"Partitioning signed social networks"
"Moustaki, Irini";editorOf;"Psychometrika"
"Doreian, Patrick";editorOf;"Social Networks"
"Generalized Blockmodeling";containedIn;"Structural Analysis in the Soci
"Clustering with relational constraint";containedIn;"Psychometrika"
"The Strength of Weak Ties";containedIn;"The American Journal of Sociolo
"Partitioning signed social networks";containedIn;"Social Networks"
"Partitioning signed social networks";cites;"Generalized Blockmodeling"
"Generalized Blockmodeling";cites;"Clustering with relational constraint
"Structural Analysis in the Social Sciences";publishedBy;"Cambridge Univ
"Psychometrika";publishedBy;"Springer"
```

bibLinks.csv

Faktorizacija in opis velikih omrežij

V analizi podatkov za varčevanje s pomnilnikom in večjo računsko učinkovitost pogosto zamenjamo vrednosti imenskih (kategoričnih) spremenljivk s (naravno) številskimi kodami. V R-ju temu rečemo *faktorizacija*.

Vse možne vrednosti imenske spremenljivke oštevilčimo (kodna tabela) in nato zamenjamo vrednosti s pripadajočimo indeksi v tabeli.

Ta pristop se uporablja tudi v večini programov za delo z velikimi omrežji. Žal je pogosto kodna tabela pozabljena ali ohranjena kot meta podatki.

Branje Pajkovega omrežja v R (wiki).

CSV2Pajek.R

```
# transforming CSV file to Pajek files
# by Vladimir Batagelj, June 2016
# setwd("C:/Users/batagelj/work/Python/graph/SVG/EUSN")
# colC <- c(rep("character",4),rep("numeric",7)); nas=c("","NA","NaN")
colC <- c(rep("character",4),rep("numeric",5)); nas=c("","NA","NaN")
nodes <- read.csv2("bibNodes.csv",encoding='UTF-8',colClasses=colC,na.strings=nas)
n <- nrow(nodes); M <- factor(nodes$mode); S <- factor(nodes$sex)
mod <- levels(M); sx <- levels(S); S <- as.numeric(S); S[is.na(S)] <- 0
links <- read.csv2("bibLinks.csv",encoding='UTF-8',colClasses="character")
F <- factor(links$from,levels=nodes$name,ordered=TRUE)
T <- factor(links$to,levels=nodes$name,ordered=TRUE)
R <- factor(links$relation); rel <- levels(R)
net <- file("bib.net","w"); cat('*vertices ',n,'\n',file=net)
clu <- file("bibMode.clu","w"); sex <- file("bibSex.clu","w")
cat('%',file=clu); cat('%',file=sex)
for(i in 1:length(mod)) cat(' ',i,mod[i],file=clu)
cat('\n*vertices ',n,'\n',file=clu)
for(i in 1:length(sx)) cat(' ',i,sx[i],file=sex)
cat('\n*vertices ',n,'\n',file=sex)
for(v in 1:n) {
  cat(v,' "',nodes$name[v],'"\n',sep='',file=net);
  cat(M[v],'\n',file=clu); cat(S[v],'\n',file=sex)
}
for(r in 1:length(rel)) cat('*arcs :',r,' "',rel[r],'"\n',sep='',file=net)
cat('*arcs\n',file=net)
for(a in 1:nrow(links))
  cat(R[a],': ',F[a],' ',T[a],' 1 l "',rel[R[a]],'"\n',sep='',file=net)
close(net); close(clu); close(sex)
```

CSV2Pajek.R

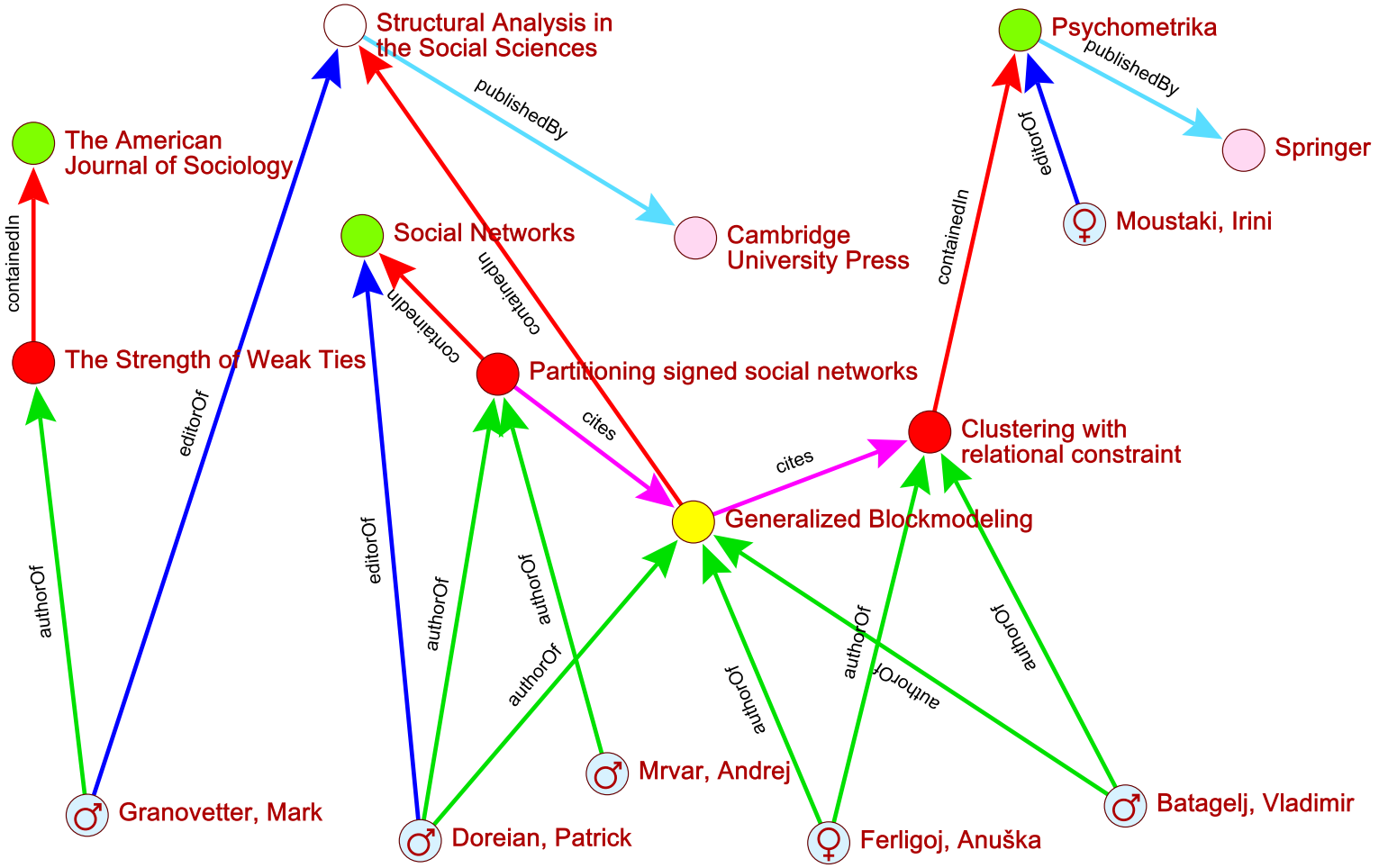
bib.net

```
*vertices  16
1 "Batagelj, Vladimir"
2 "Doreian, Patrick"
3 "Ferligoj, Anuška"
4 "Granovetter, Mark"
5 "Moustaki, Irini"
6 "Mrvar, Andrej"
7 "Clustering with relational constraint"
8 "The Strength of Weak Ties"
9 "Partitioning signed social networks"
10 "Generalized Blockmodeling"
11 "Psychometrika"
12 "Social Networks"
13 "The American Journal of Sociology"
14 "Structural Analysis in the Social Sciences"
15 "Cambridge University Press"
16 "Springer"
*arcs :1 "authorOf"
*arcs :2 "cites"
*arcs :3 "containedIn"
*arcs :4 "editorOf"
*arcs :5 "publishedBy"
```

```
*arcs
1: 1 10 1 l "authorOf"
1: 2 10 1 l "authorOf"
1: 3 10 1 l "authorOf"
1: 1 7 1 l "authorOf"
1: 3 7 1 l "authorOf"
1: 4 8 1 l "authorOf"
4: 4 14 1 l "editorOf"
1: 2 9 1 l "authorOf"
1: 6 9 1 l "authorOf"
4: 5 11 1 l "editorOf"
4: 2 12 1 l "editorOf"
3: 10 14 1 l "containedIn"
3: 7 11 1 l "containedIn"
3: 8 13 1 l "containedIn"
3: 9 12 1 l "containedIn"
2: 9 10 1 l "cites"
2: 10 7 1 l "cites"
5: 14 15 1 l "publishedBy"
5: 11 16 1 l "publishedBy"
```

bib.net, bibMode.clu, bibSex.clu; bib.paj, bib.ini.

Bibliografsko omrežje – slika / Pajek



Kako do omrežja?

Rodovniki

Molekule

GraphML

Analiza besedil

Odnosi med državami

Dvovrstna omrežja

Multivariatna omrežja

Internet

Slučajna omrežja

Viri

Analiza omrežij

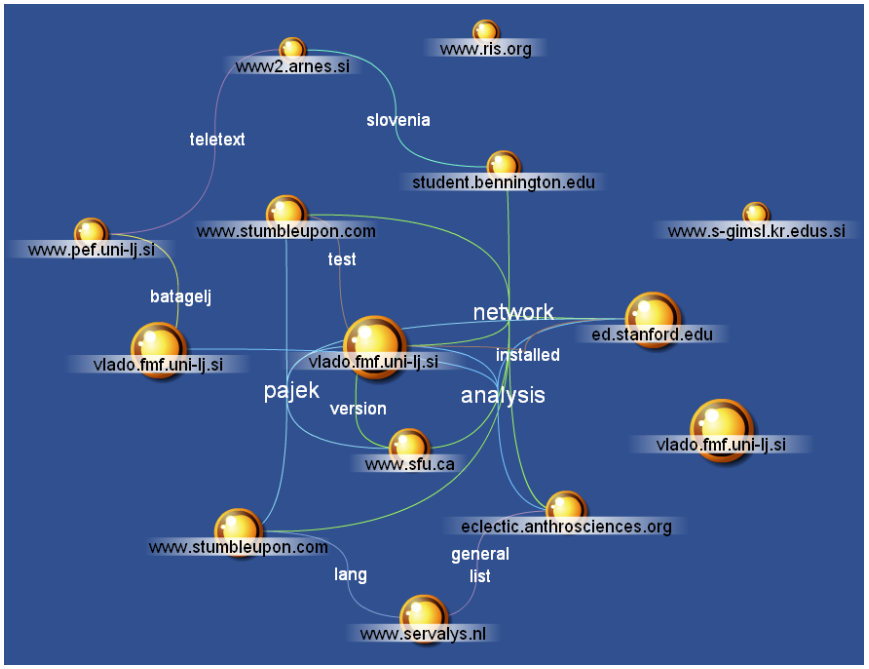
3. Omrežja vsepovsod

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, april 2024

Kazalo

1 Kako do omrežja?
2 Rodovniki
3 Molekule
4 GraphML
5 Analiza besedil
6 Odnosi med državami
7 Dvovrstna omrežja
8 Multivariatna omrežja
9 Internet
10 Slučajna omrežja
11 Viri



prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

22. april 2024 ob 01 : 15/ marec 2013

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

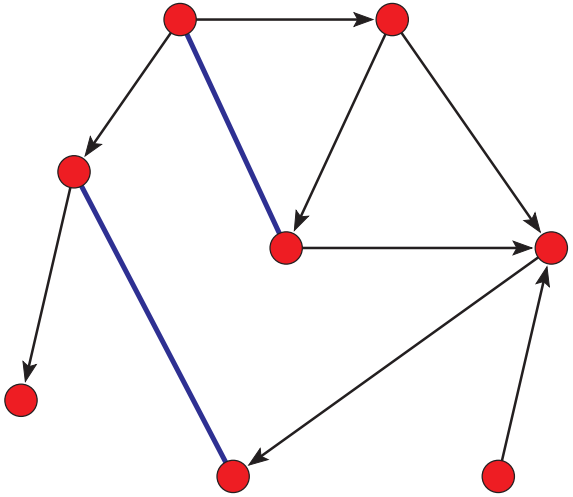
Kako do omrežja?

Pri zbiranju podatkov o omrežju $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ se moramo odločiti, kaj je množica enot (vozlišč) – *meje omrežja*, kdaj sta vozlišči povezani – *polnost omrežja* in katere lastnosti vozlišč/povezav bomo upoštevali.

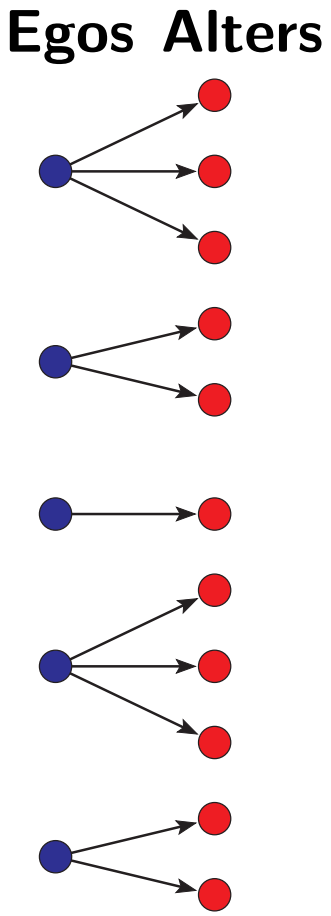
Ta vprašanja so še posebej pereča pri merjenju družbenih omrežij (vprašalniki, pogovori, opazovanja, arhivski zapiski, poskusi, ...). Nekatere 'enote' nočeje sodelovati. Nekateri postopki merjenja, na primer, omejujejo število sosedov ... Naslednja velika ovira pri izgradnji teh omrežij je spoštovanje zasebnosti. Pri tem si pri velikih množicah enot ne moremo privoščiti polnega opisa. Omrežje izmerimo samo za izbrane enote (in njihove sosede). Tako dobljena omrežja imenujemo *osebna omrežja*. O merjenju družbenih omrežij si lahko preberete v knjigi V. Hlebec in T. Kogovšek (2006).

V zadnjih 20 letih vsebolj pridobivamo omrežja iz podatkovij, ki so že dostopna na računalniku ((spletne) podatkovne baze, dnevniki uporabe, družabna omrežja, ...).

Cela in osebna omrežja



celo omrežje



osebna omrežja

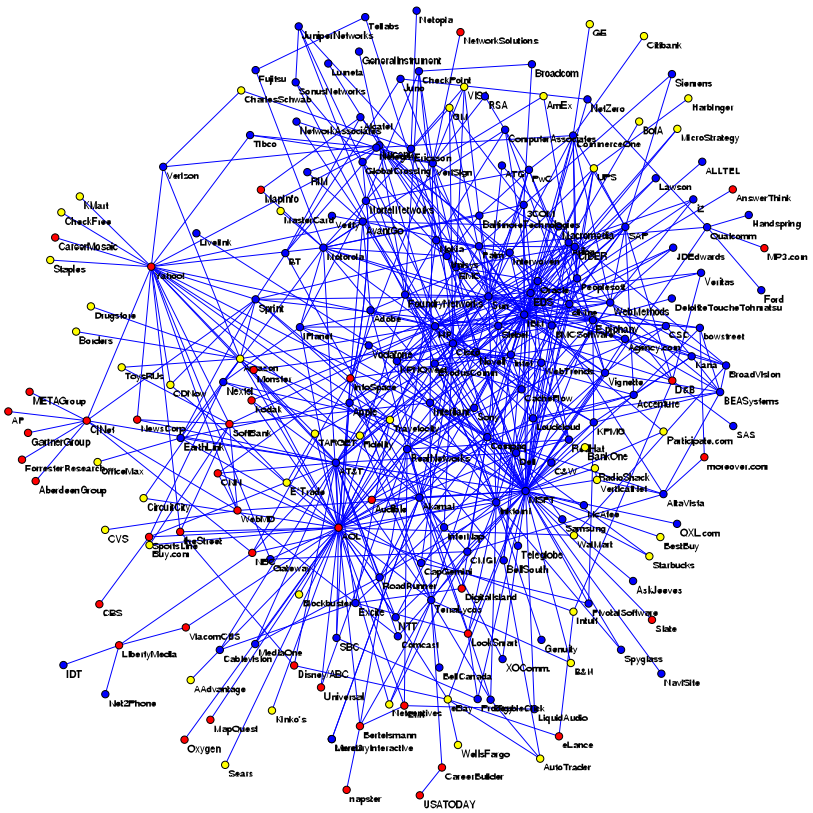
Uporaba že zbranih omrežij

Pajek omogoča branje več drugih oblik opisa omrežij: UCINETove datoteke DL, grafi projekta Vega, kemični opisi molekul MDLMOL, MAC in BS ter rodovniki v obliki GEDCOM.
Davis.DAT, C84N24.VGR, MDL, 1CRN.BS, DNA.BS, ADF073.MAC, Bouchard.GED.

Večkrat pa je potrebno dani opis omrežja le nekoliko popraviti (dodati določila *vertices, *arcs,...; zamenjati vozlišče 0 s pozitivnim številom) v znakovnem urejevalniku in ga tako predelati v zahtevano obliko.

Za nekoliko zahtevnejše predelave poglejte Jürgen Pfeffer: txt2pajek.

Krebsova Internetska podjetja



Omrežje sestavljajo izbrana Internetska podjetja v obdobju 1998 do 2001.
$n = 219$, $m = 631$.
rdeča – vsebina,
modra – podpora,
rumena – trgovina.
Podjetji sta povezani, če sta objavili skupni posel ali sodelovanje.

Spletni naslov: `http://www.orgnet.com/netindustry.html`. *Recode*, *InfoRapid*.

Rodovniki

Za opis rodovnikov se najpogostoje uporablja oblika zapisa GEDCOM (*GEDCOM standard 5.5*).
Veliko rodovnikov (datoteke `*.GED`) najdemo na spletu – na primer *Roper's GEDCOMs*, *Genealogy Forum GEDCOMs*, *KinSource* ali *Isle-of-Man GEDCOMs*. `Family.GED`.
Za pripravo in vzdrževanje rodovnikov je na voljo več programov: prosti *GIM* in tržni *Brothers Keeper* (obstaja tudi slovenska različica – *SRD*).
Iz rodovnikov zbranih v doktoratu: Mahnken, Irmgard. 1960. Dubrovački patricijat u XIV veku. Beograd, Naučno delo. je bilo ustvarjeno podatkovje *Ragusa*.

GEDCOM

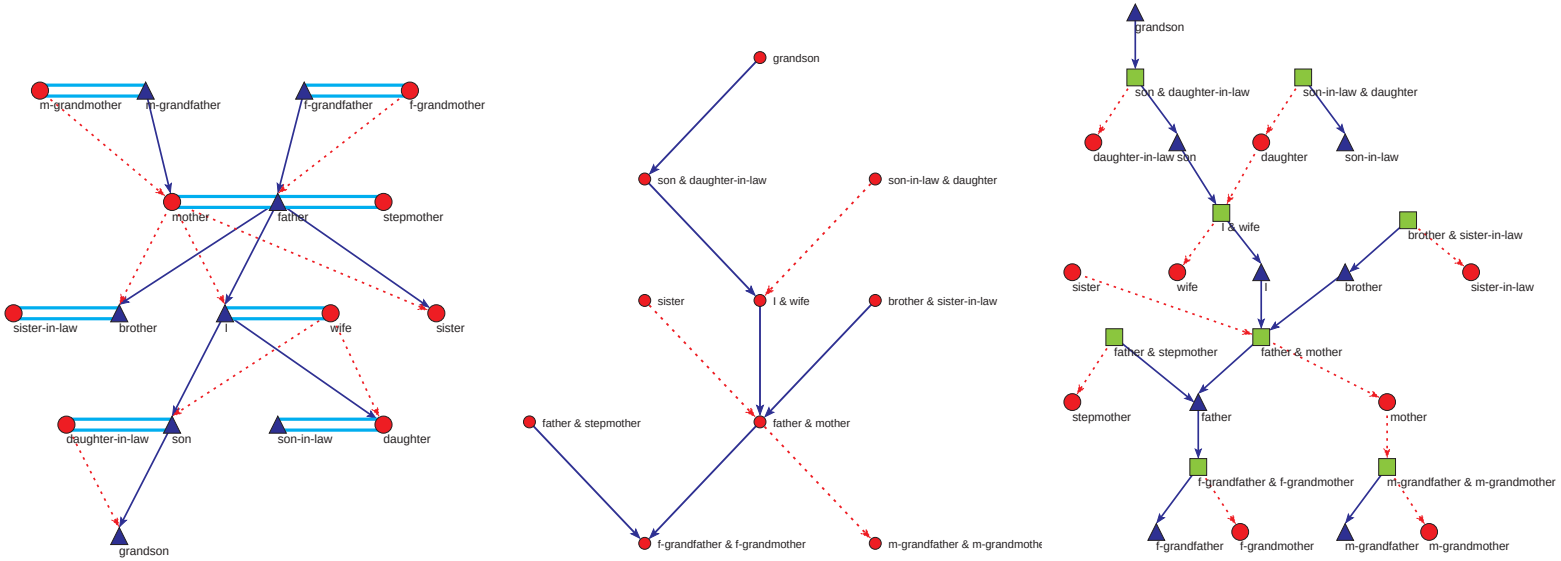
GEDCOM je dogovor o zapisu rodoslovnih podatkov, ki se uporablja za izmenjavo in združevanje podatkov iz različnih programov uporabljenih za pripravo podatkov.

```
0 HEAD
1 FILE ROYALS.GED
...
0 @I58@ INDI
1 NAME Charles Philip Arthur/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 14 NOV 1948
2 PLAC Buckingham Palace, London
1 CHR
2 DATE 15 DEC 1948
2 PLAC Buckingham Palace, Music Room
1 FAMS @F16@
1 FAMC @F14@
...
...
0 @I65@ INDI
1 NAME Diana Frances /Spencer/
1 TITL Lady
1 SEX F
1 BIRT
2 DATE 1 JUL 1961
2 PLAC Park House, Sandringham
1 CHR
2 PLAC Sandringham, Church
1 FAMS @F16@
1 FAMC @F78@
...
...
```

```
0 @I115@ INDI
1 NAME William Arthur Philip/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 21 JUN 1982
2 PLAC St.Mary's Hospital, Paddington
1 CHR
2 DATE 4 AUG 1982
2 PLAC Music Room, Buckingham Palace
1 FAMC @F16@
...
0 @I116@ INDI
1 NAME Henry Charles Albert/Windsor/
1 TITL Prince
1 SEX M
1 BIRT
2 DATE 15 SEP 1984
2 PLAC St.Mary's Hosp., Paddington
1 FAMC @F16@
...
0 @F16@ FAM
1 HUSB @I58@
1 WIFE @I65@
1 CHIL @I115@
1 CHIL @I116@
1 DIV N
1 MARR
2 DATE 29 JUL 1981
2 PLAC St.Paul's Cathedral, London
```

Omrežne predstavitve rodovnikov

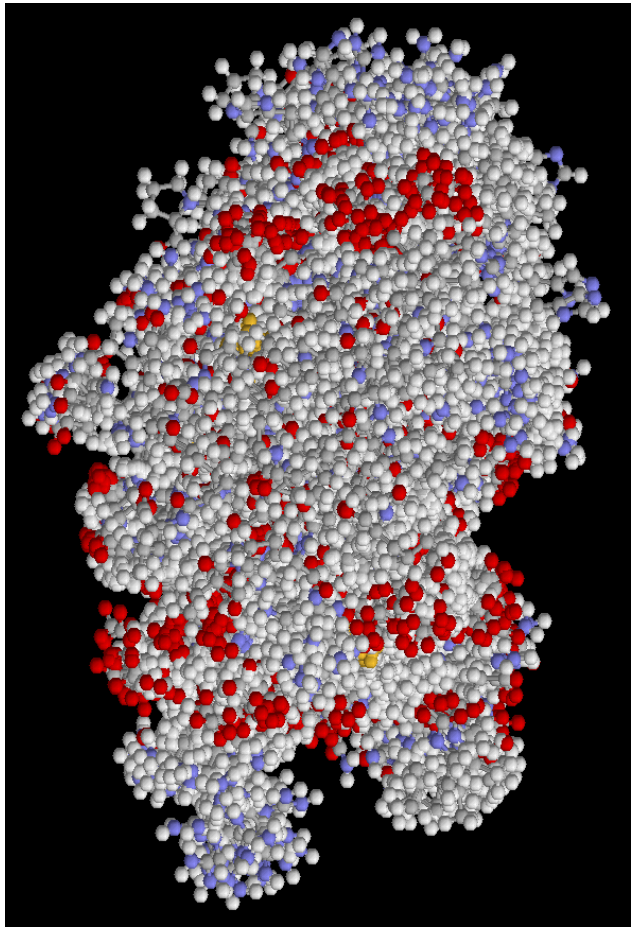
Običajna predstavitev rodovnika z grafom, *Orejev graf*, ima za vozlišča posameznike in združuje dve relaciji: *sta poročena* (modro-bela neusmerjena) in *ima otroka* (usmerjena), razbita na *je oče od* (modra) in *je mati od* (črtkana rdeča). V *parnem grafu* so vozlišča poročeni pari ali neporočeni posamezniki, in relaciji *je sin* (polna modra) in *je hči* (črtkana rdeča). Več o parnih grafih *D. White*.



Orejev graf, parni graf in dvodelni parni graf

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

Omrežja molekul



virus 1GDY: $n = 39865$, $m = 40358$

V zbirki Brookhaven Protein Data Bank lahko najdemo veliko velikih organskih molekul (npr. Simian / 1AZ5.pdb) opisanih v obliki PDB. Molekulo si lahko ogledamo s programom Rasmol (*RasMol*, *program*, *RasWin*) ali *Protein Explorer*. Molekulo v obliki PDB lahko predelamo v obliko BS, ki jo pozna `Pajek`, s programom *BabelWin* + *Babel16*.

Koristno bi bilo imeti podoben program za splošna omrežja.

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

GraphML

GraphML – zapis omrežja v obliki XML.
L'Institut de Linguistique et Phonétique Générales et Appliquées (ILPGA), Paris III; Traitement Automatique du Langage (TAL):
BaO4 : Des Textes Aux Graphes Plurital
LibXML, `xsltproc` download, XSLT, Xalan, Python, Sxslt.

```
xsltproc GraphML2Pajek.xsl graph.xml > graph.net
java -jar saxon8.jar graph.xml GraphML2Pajek.xsl > graph.net
java org.apache.xalan.xslt.Process -IN p.xml -XSL m.xsl -OUT p.tx
```

XSLT/Zvon

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Title: 1. D:\vlado\docs\Books\SKRIPTA\Nets\nets\graph.net (12) -->
<!-- Creator: Pajek: http://vlado.fmf.uni-lj.si/pub/networks/pajek/ -->
<!-- CreationDate: 11-03-2006, 17:25:13 -->
<graphml>
  <key id="a1" for="node" attr.name="Label" attr.type="string">
    <desc>Label of the node</desc> <default>NoLabel</default>
  </key>
  <key id="b1" for="edge" attr.name="Weight" attr.type="double">
    <desc>Weight (value) of the edge</desc> <default>1</default>
  </key>
  <graph id="G" edgedefault="directed" parse.nodes="12" parse.edges="23">
    <node id="v1"><data key="a1">a</data></node>
    <node id="v2"><data key="a1">b</data></node>
    <node id="v3"><data key="a1">c</data></node>
    <node id="v4"><data key="a1">d</data></node>
    <node id="v5"><data key="a1">e</data></node>
    <node id="v6"><data key="a1">f</data></node>
    <node id="v7"><data key="a1">g</data></node>
    <node id="v8"><data key="a1">h</data></node>
    <node id="v9"><data key="a1">i</data></node>
    <node id="v10"><data key="a1">j</data></node>
    <node id="v11"><data key="a1">k</data></node>
    <node id="v12"><data key="a1">l</data></node>
    <edge source="v1" target="v2"/> <edge source="v2" target="v1"/>
    <edge source="v1" target="v4"/> <edge source="v1" target="v6"/>
    <edge source="v2" target="v6"/> <edge source="v3" target="v2"/>
    <edge source="v3" target="v3"/> <edge source="v3" target="v7"/>
    <edge source="v3" target="v7"/> <edge source="v5" target="v3"/>
    <edge source="v5" target="v6"/> <edge source="v5" target="v8"/>
    <edge source="v6" target="v11"/> <edge source="v8" target="v4"/>
    <edge source="v10" target="v8"/> <edge source="v12" target="v5"/>
    <edge source="v12" target="v7"/> <edge source="v8" target="v12"/>
    <edge source="v12" target="v8"/>
    <edge directed="false" source="v2" target="v5"/>
    <edge directed="false" source="v3" target="v4"/>
    <edge directed="false" source="v5" target="v7"/>
    <edge directed="false" source="v6" target="v8"/>
  </graph>
</graphml>
```


```
*Vertices 12
1 "a"
2 "b"
3 "c"
4 "d"
5 "e"
6 "f"
7 "g"
8 "h"
9 "i"
10 "j"
11 "k"
12 "l"
*Edges
2 5
3 4
5 7
6 8
*Arcs
1 2
2 1
1 4
1 6
2 6
3 2
3 3
3 7
3 7
5 3
5 6
5 8
6 11
8 4
10 8
12 5
12 7
8 12
12 8
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="iso-8859-1"/>
  <xsl:template match="/">
    <xsl:text>*Vertices </xsl:text>
    <xsl:value-of select="count(graphml/graph/node)"/>
    <xsl:text>&#10;</xsl:text>
    <xsl:apply-templates select="graphml/graph/node"/>
    <xsl:text>*Edges&#10;</xsl:text>
    <xsl:apply-templates select="graphml/graph/edge" mode="edge"/>
    <xsl:text>*Arcs&#10;</xsl:text>
    <xsl:apply-templates select="graphml/graph/edge" mode="arc"/>
  </xsl:template>

  <xsl:template match="edge" mode="arc">
    <xsl:if test="not(./@directed='false')">
      <xsl:value-of select="substring(./@source,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="substring(./@target,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="./data"/>
      <xsl:text>&#10;</xsl:text>
    </xsl:if>
  </xsl:template>

  <xsl:template match="edge" mode="edge">
    <xsl:if test="./@directed='false'">
      <xsl:value-of select="substring(./@source,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="substring(./@target,2)"/>
      <xsl:text> </xsl:text>
      <xsl:value-of select="./data"/>
      <xsl:text>&#10;</xsl:text>
    </xsl:if>
  </xsl:template>

  <xsl:template match="node">
    <xsl:value-of select="substring(./@id,2)"/>
    <xsl:text> "</xsl:text>
    <xsl:value-of select="./data"/>
    <xsl:text>"&#10;</xsl:text>
  </xsl:template>

</xsl:stylesheet>
```

Pristopi k računalniško podprti analizi besedil

R. Popping: Computer-Assisted Text Analysis (2000) razlikuje tri glavne pristope k RPAB: *tematska* AB, *pomenska* AB, in *omrežna* AB.

Pojmi (besede, besedne zveze, izrazi, ...) upoštevani pri AB so zbrani v *slovarju*. Ta je lahko določen vnaprej ali pa se gradi sproti. Osnovni vprašanji pri tem sta *enakovrednost zapisov* – različni zapisi, ki predstavljajo isti pojem; in *dvoumnost zapisov* – isti zapis lahko predstavlja več pojmov. Poskusi rešitve: za imena: ResearcherID, ORCID, AMS; za besede: krnjenje (stemming) in geslenje (lemmatization).

Zato je *kodiranje* – pretvorba surovih podatkov v formalni *opis*, pogosto opravljeno pretežno ročno ali vsaj pod nadzorom uporabnika. Kot *enote* AB ponavadi vzamemo stavke, odstavke, novice, sporočila, ...

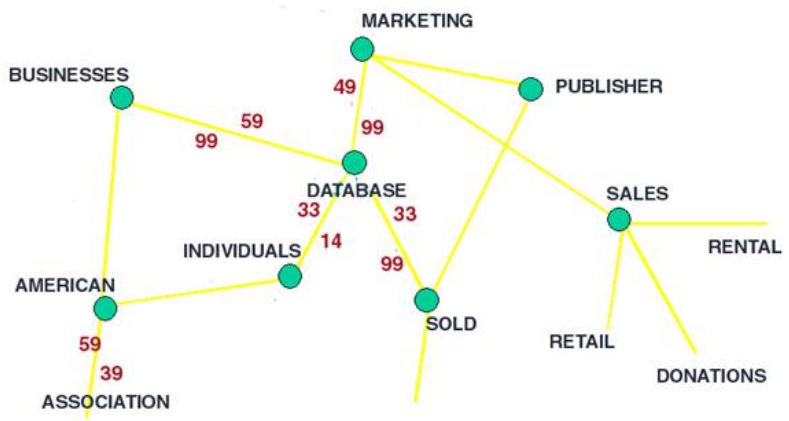
Dosedaj sta tematska in pomenska AB temeljili predvsem na statistični analizi kodiranih podatkov.

... pristopi k RPAB

Pri tematski AB so enote besedila kodirane s pravokotno matriko *Enote* $\times$ *Pojmi* : pojem p se pojavlja v enoti e – dvovrstno omrežje.
Primeri: M.M. Miller: VBPro, H. Klein: Text Analysis/ TextQuest.
Pri pomenski AB so enote (največkrat enostavni stavki) kodirane po shemi o-P-p (*osebek-Povedek-predmet*) ali njenih izpeljankah.

Primeri: Roberto Franzosi; *KEDS*, *Tabari*.
To kodiranje določa večrelacijsko omrežje na množici vozlišč *Osebki* $\cup$ *Predmeti* s povezavami iz relacij *Povedki*.
Glejte tudi RDF v pomenskem spletu (semantic web), SPARQL.

Omrežna RPAB



TextAnalyst's 'semantic network'

Tako smo že v omrežni AB.
Primeri:
Carley: Cognitive maps,
J.A. de Ridder: CETA,
Megaputer: TextAnalyst.

Glejte še: W. Evans: Computer Environments for Content Analysis, K.A. Neuendorf: The Content Analysis Guidebook / Online and H.D. White: Publications.
Obstajajo tudi drugi načini, kako ustvariti omrežja iz besedil.

AB – slovarska omrežja

book

A collection of leaves of paper, parchment, vellum, cloth, or other material (written, printed, or blank) fastened together along one edge, with or without a protective case or cover. Also refers to a literary work or one of its volumes. Compare with monograph.

To qualify for the special parcel post rate known in the United States as media rate, a publication must consist of 24 or more pages, at least 22 of which bear printing consisting primarily of reading material or scholarly bibliography, with advertising limited to book announcements. UNESCO defines a book as a nonperiodical literary publication consisting of 49 or more pages, covers excluded. The ANSI standard includes publications of less than 49 pages which have hard covers. ***See also***: art book, board book, children's book, coffee table book, gift book, licensed book, managed book, new book, packaged book, picture book, premium book, professional book, promotional book, rare book, reference book, religious book, and reprint book.

Also, a major division of a longer work (usually of fiction) which is further subdivided into chapters. Usually numbered, such a division may or may not have its own title. Also refers to one of the divisions of the Christian ***Bible***, the first being *Genesis*.

opis pojma **book** v ODLIS

V *slovarskem omrežju* so vozlišča v slovarju opisani pojmi; iz pojma *u* vodi povezava do pojma *v* ntk. pojem *v* nastopa v opisu pojma *u*.

Online Dictionary of Library and Information Science *ODLIS*, *Odlis.net* (2909 / 18419).

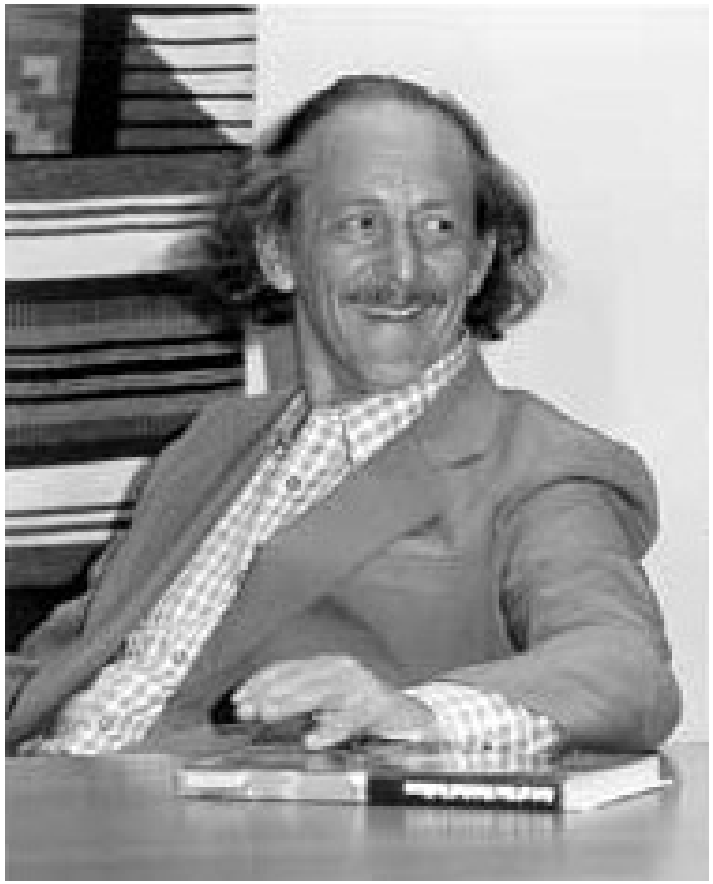
Free On-line Dictionary of Computing *FOLDOC*, *Foldoc2b.net* (133356 / 120238).

Artlex, *Wordnet*, *ConceptNet*, *OpenCyc*.

The Edinburgh Associative Thesaurus (*EAT*) / *net* je bil zbran s spraševanjem (študentov).

NASA Thesaurus. *Paper*.

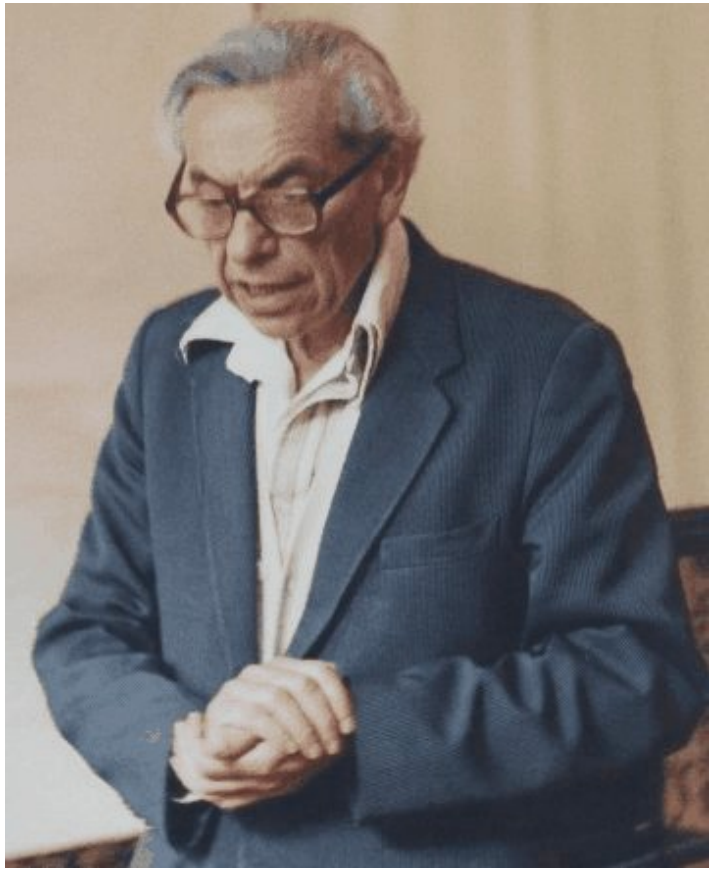
AB – Omrežja sklicevanj



V *omrežju sklicevanj* so vozlišča razna dela (članki, knjige, poročila, ...) iz izbranega področja; deli sta povezani z usmerjeno povezavo, če se prvo sklicuje na drugo. Omrežja sklicevanj so (skoraj) aciklična.
E. Garfield: HistCite / *Pajek*, *papers*.

Primer zelo velikega omrežja sklicevanj je US Patents / Nber,
$n = 3774768$, $m = 16522438$.
extended data

AB – Omrežja sodelovanj



V omrežjih sodelovanj so enote osebe ali ustanove. Enoti sta povezani, če sta sodelovali pri skupnem delu. Utež povezave je število skupnih del.

Najbolj poznano omrežje sodelovanja je *The Erdős Number Project*, *Erdos.net*. Določitev Erdősevega števila.

Bogat vir podatkov za izgradnjo omrežij sodelovanj so bibliografije v BibTEXu *Nelson H. F. Beebe's Bibliographies Page* – primer B. Jones-ova bibliografija računalniške geometrije *Computational geometry database* (2002), *FTP*, *Geom.net*.

Pri pripravi omrežja sodelovanj iz izvornih podatkov si lahko pomagamo z ustreznimi programi. Nato pa sledi mukotrpno čiščenje - ugotavljanje enot.

Glejte tudi: *The Internet Movie Database*; Trier DBLP: Digital Bibliography & Library Project; 1124. sredin seminar. Omrežja sklicevanj in sodelovanj je mogoče dobiti iz Web of Science z WoS2Pajek. Glejte še Bibexcel.

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

Odnosi med državami

Paul Hensel's International Relations Data Site,
International Conflict and Cooperation Data,
Correlates of War,
Kansas Event Data System *KEDS*,
KEDSi na Pajkovih datotekah,
Prekodirni programi v R-ju.

Pretvorba podatkov KEDS/WEIS v Pajekovo obliko


```
% Recoded by WEISmonths, Sun Nov 28 21:57:00 2004
% from http://www.ku.edu/~keds/data.dir/balk.html
*vertices 325
1 "AFG" [1-*]
2 "AFR" [1-*]
3 "ALB" [1-*]
4 "ALBMED" [1-*]
5 "ALG" [1-*]
 ...
318 "YUGGOV" [1-*]
319 "YUGMAC" [1-*]
320 "YUGMED" [1-*]
321 "YUGMTN" [1-*]
322 "YUGSER" [1-*]
323 "ZAI" [1-*]
324 "ZAM" [1-*]
325 "ZIM" [1-*]
*arcs :0 "*** ABANDONED"
*arcs :10 "YIELD"
*arcs :11 "SURRENDER"
*arcs :12 "RETREAT"
 ...
*arcs :223 "MIL ENGAGEMENT"
*arcs :224 "RIOT"
*arcs :225 "ASSASSINATE TORTURE"
*arcs
224: 314 153 1 [4]                890402  YUG     KSV     224  (RIOT)  RIOT-TORN
212: 314 83 1 [4]                 890404  YUG     ETHALB  212  (ARREST PERSON) ALB ETHNIC JAILED
224: 3 83 1 [4]                   890407  ALB     ETHALB  224  (RIOT)  RIOTS
123: 83 153 1 [4]                 890408  ETHALB  KSV     123  (INVESTIGATE)   PROBING
 ...
42: 105 63 1 [175]                030731  GER     CYP     042  (ENDORSE)       GAVE SUPPORT
212: 295 35 1 [175]               030731  UNWCT   BOSSER  212  (ARREST PERSON) SENTENCED TO PRIS
43: 306 87 1 [175]                030731  VAT     EUR     043  (RALLY) RALLIED
13: 295 35 1 [175]                030731  UNWCT   BOSSER  013  (RETRACT)       CLEARED
121: 295 22 1 [175]               030731  UNWCT   BAL     121  (CRITICIZE)     CHARGES
122: 246 295 1 [175]              030731  SER     UNWCT   122  (DENIGRATE)     TESTIFIED
121: 35 295 1 [175]               030731  BOSSER  UNWCT   121  (CRITICIZE)     ACCUSED
```

... Program v R-ju

Za pretvorbo podatkov KEDS/WEIS smo uporabili kratke programe v R-ju, kot je naslednji:

```
# WEISmonths
# recoding of WEIS files into Pajek's multirelational temporal files
# granularity is 1 month
# ------------------------------------------------------------------
# Vladimir Batagelj, 28. November 2004
# ------------------------------------------------------------------
# Usage:
#   WEISmonths(WEIS_file,Pajek_file)
# Examples:
#   WEISmonths('Balkan.dat','BalkanMonths.net')
# ------------------------------------------------------------------
# http://www.ku.edu/~keds/data.html
# ------------------------------------------------------------------

WEISmonths <- function(fdat,fnet){

  get.codes <- function(line){
    nlin <<- nlin + 1;
    z <- unlist(strsplit(line,"\t")); z <- z[z != ""]
    if (length(z)>4) {
      t <- as.numeric(z[1]); if (t < 500000) t <- t + 1000000
      if (t<t0) t0 <<- t; u <- z[2]; v <- z[3]; r <- z[4]
      if (is.na(as.numeric(r))) cat(nlin,'NA rel-code',r,'\n')
      h <- z[5]; h <- substr(h,2,nchar(h)-1)
      if (nchar(h) == 0) h <- '*** missing description'
      if (!exists(u,env=act,inherits=FALSE)){
        nver <<- nver + 1; assign(u,nver,env=act) }
      if (!exists(v,env=act,inherits=FALSE)){
        nver <<- nver + 1; assign(v,nver,env=act) }
      if (!exists(r,env=rel,inherits=FALSE)) assign(r,h,env=rel)
    }
  }
```

```
  recode <- function(line){
    nlin <<- nlin + 1;
    z <- unlist(strsplit(line,"\t")); z <- z[z != ""]
    if (length(z)>4) {
      t <- as.numeric(z[1]); if (t < 500000) t <- t + 1000000
      cat(as.numeric(z[4]),': ',get(z[2],env=act,inherits=FALSE),
        ' ',get(z[3],env=act,inherits=FALSE),' 1 [',
        12*(1900 + t %/% 10000) + (t %% 10000) %/% 100 - t0,
        ']\n',sep='',file=net)
    }
  }

  cat('WEISmonths: WEIS -> Pajek\n')
  ts <- strsplit(as.character(Sys.time())," ")[[1]][2]
  act <- new.env(TRUE,NULL); rel <- new.env(TRUE,NULL)
  dat <- file(fdat,"r"); net <- file(fnet,"w")
  lst <- file('WEIS.lst',"w"); dni <- 0
  nver <- 0; nlin <- 0; t0 <- 9999999
  lines <- readLines(dat); close(dat)
  sapply(lines,get.codes)
  a <- sort(ls(envir=act)); n <- length(a)
  cat(paste('% Recoded by WEISmonths,',date()),"\n",file=net)
  cat("% from http://www.ku.edu/~keds/data.html\n",file=net)
  cat("*vertices",n,"\n",file=net)
  for(i in 1:n){ assign(a[i],i,env=act);
    cat(i,' "',a[i],'" [1-*]\n',sep='',file=net) }
  b <- sort(ls(envir=rel)); m <- length(b)
  for(i in 1:m){ assign(a[i],i,env=act);
  cat("*arcs :",as.numeric(b[i]),' "',
  get(b[i],env=rel,inherits=FALSE),'"\n',sep='',file=net) }
  t0 <- 12*(1900 + t0 %/% 10000)
  slice <- 0
  cat("*arcs\n",file=net); nlin <- 0
  sapply(lines,recode)
  cat(' ',nlin,'lines processed\n'); close(net)
  te <- strsplit(as.character(Sys.time())," ")[[1]][2]
  cat('  start:',ts,'  finish:',te,'\n')
}

WEISmonths('Balkan.dat','BalkanMonthsR.net')
```

Opomba: Slovarjem (dictionary) se v R-ju reče *environment*.

Dvovrstna omrežja iz podatkovnih baz

Enostavno *podatkovno bazo* $\mathcal{B}$ sestavlja množica *zapisov* $\mathcal{B} = \{R_k : k \in \mathcal{K}\}$, kjer je $\mathcal{K}$ množica *ključev*. Posamezni zapis ima obliko $R_k = (k, q_1(k), q_2(k), \ldots, q_r(k))$ kjer je $q_i(k)$ opis/vrednost *lastnosti* $\mathbf{q}_i$ za enoto določeno s ključem k.
Naj opis $q(k)$ zavzema vrednosti v končni množici $\mathcal{Q}$. Tak opis lahko vselej zagotovimo z diskretizacijo množice $\mathcal{Q}$. Potem lahko za izbrano lastnost $\mathbf{q}$ ustvarimo pripadajoče *dvovrstno omrežje* $\mathcal{K} \times \mathbf{q} = (\mathcal{K}, \mathcal{Q}, \mathcal{L}, w)$, kjer je $(k, v) \in \mathcal{L}$ ntk $v \in q(k)$. $w(k, v)$ je *utež* na povezavi (k, v); če ni drugače določeno, postavimo $w(k, v) = 1$.
Lastnosti z eno samo vrednostjo lahko varčneje predstavimo z razbitjem.

Primeri:
(članki, avtorji, je bil soustvarjen od),
(članki, ključne besede, je opisan z),
(poslanci, zakoni, je glasoval za),
(ljudje, časopisi, bere),
(ljudje, združenja, je član, število let članstva),
(kupci, dobrine, je kupil, vrednost), itd.

Zapis z Web of Science

```
PT J
AU Dipple, H
   Evans, B
TI The Leicestershire Huntington's disease support group: a social network
   analysis
SO HEALTH & SOCIAL CARE IN THE COMMUNITY
LA English
DT Article
C1 Rehabil Serv, Troon Way Business Ctr, Leicester LE4 9HA, Leics, England.
RP Dipple, H, Rehabil Serv, Troon Way Business Ctr, Sandringham
   Suite,Humberstone Lane, Leicester LE4 9HA, Leics, England.
CR BORGATTI SP, 1992, UCINET 4 VERSION 1 0
   FOLSTEIN S, 1989, HUNTINGTONS DIS DISO
   SCOTT J, 1991, SOCIAL NETWORK ANAL
NR 3
TC 3
PU BLACKWELL SCIENCE LTD
PI OXFORD
PA P O BOX 88, OSNEY MEAD, OXFORD OX2 ONE, OXON, ENGLAND
SN 0966-0410
J9 HEALTH SOC CARE COMMUNITY
JI Health Soc. Care Community
PD JUL
PY 1998
VL 6
IS 4
BP 286
EP 289
PG 4
SC Public, Environmental & Occupational Health; Social Work
GA 105UP
UT ISI:000075092200008
ER
```

Program WoS2Pajek

Za pretvorbo datotek z WoS v nabor Pajkovih omrežij je bil v Pythonu napisan program WoS2Pajek, ki ustvari naslednje datoteke:

- omrežje sklicevanj: dela $\times$ dela;
- omrežje avtorstev (dvovrstno): dela $\times$ avtorji; za dela brez polnih opisov je znan le prvi avtor;
- omrežje ključnih besed (dvovrstno): dela $\times$ ključne besede; samo za dela s polnimi opisi;
- omrežje revij (dvovrstno): dela $\times$ revije – polje J9;
- razbitje del glede na leto objave;
- razbitje del – polni opis (1) / samo ime ISI (0);
- vektor s številom strani za posamezno delo – PG ali EP − BP +1.

Podoben paket je pripravil tudi Loet Leydesdorff.

Omrežja sosednjih vozlišč

Recimo, da imamo na množici enot $\mathcal{U}$ dano mero različnosti $d(u, v)$. Glede na d lahko vpeljemo dve vrsti omrežij:

k-najbližjih sosedov: $\mathcal{N}(k) = (\mathcal{U}, \mathcal{A}, d)$

$$(u, v) \in \mathcal{A} \Leftrightarrow v \text{ je med } k \text{ najbližjimi sosedi vozlišča } u$$

Za utež povezave $a(u, v) \in \mathcal{A}$ postavimo $w(a) = d(u, v)$.

Omrežje r-okolic: $\mathcal{N}(r) = (\mathcal{U}, \mathcal{E}, d)$

$$(u : v) \in \mathcal{E} \Leftrightarrow d(u, v) \leq r$$

Za utež povezave $e(u : v) \in \mathcal{E}$ postavimo $w(e) = d(u, v)$.

Ta omrežja so povezava z običajno analizo podatkov. Še vedno odprto vprašanje: učinkoviti postopki za določanje teh omrežij. Multivariatne pajčevine – določitev omrežij sosedov v R-ju.

Najbližjih k sosedov v R-ju

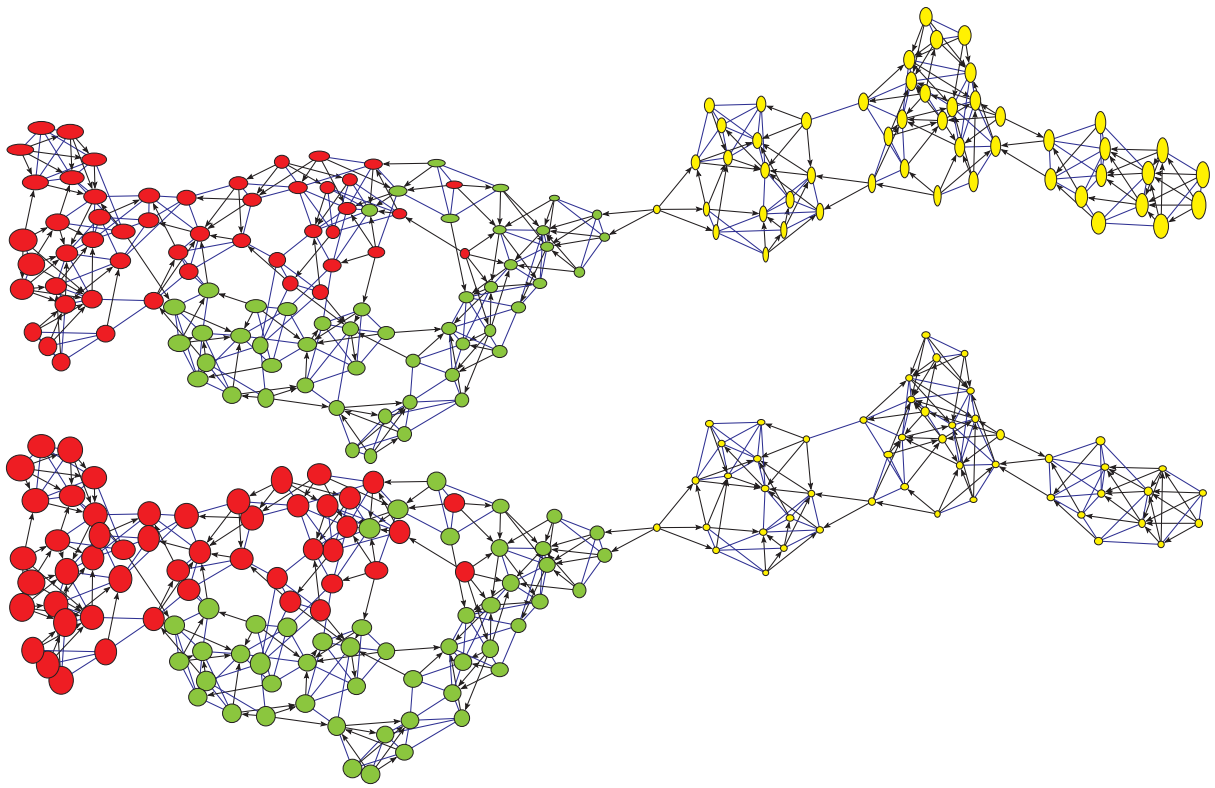
```
k.neighbor2Net <-
# stores network of first  k  neighbors for
# dissimilarity matrix  d  to file  fnet  in Pajek format.
function(fnet,d,k){
  net <- file(fnet,"w")
  n <- nrow(d); rn <- rownames(d)
  cat("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i," \"",rn[i],"\"\n",sep="",file=net)
  cat("*arcs\n",file=net)
  for (i in 1:n) for (j in order(d[i,])[1:k+1]) {
    cat(i,j,d[i,j],"\n",file=net)
  }
  close(net)
}
stand <-
# standardizes vector  x .
function(x){
  s <- sd(x)
  if (s > 0) (x - mean(x))/s  else  x - x
}
data(iris)
ir <- cbind(stand(iris[,1]),stand(iris[,2]),stand(iris[,3]),
  stand(iris[,4]))
k.neighbor2Net("iris5.net",as.matrix(dist(ir)),5)
```

Hitri postopek za najbližjih *k* sosedov v R-ju

David M. Mount je ustvaril Approximate Nearest Neighbor Library (http://www.cs.umd.edu/~mount/ANN), ki vsebuje hitre algoritme za (približno) določitev najbližjih sosedov. V R-ju so ti algoritmi dostopni s funkcijo `ann` v knjižnici `yaImpute`.

```
k.neighbor2NetF <-
# stores network of first  k  neighbors for data matrix  d  to file  fnet
# in Pajek format.
# Example:
#   data(iris); stand <- function(x){(x-mean(x))/sd(x)}
#   ir <- cbind(stand(iris[,1]),stand(iris[,2]),stand(iris[,3]),
#     stand(iris[,4]))
#   k.neighbor2NetF("iris5Y.net",ir,5)
# V. Batagelj, 8.8.2009 yaImpute / 9.9.2008 knnFinder
function(fnet,d,k){
  library(yaImpute)
  NN <- ann(ir,target=ir,k=k+1)
  net <- file(fnet,"w")
  n <- nrow(d)
  rn <- if (is.null(rownames(d))) paste("U-",1:n,sep='') else rownames(d)
  cat("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i," \"",rn[i],"\"\n",sep="",file=net)
  cat("*arcs\n",file=net)
  for (i in 1:n) for (j in 1:k)
    cat(i,NN$knnIndexDist[i,j+1],NN$knnIndexDist[i,j+k+2],"\n",file=net)
  close(net)
}
```

Fisherjeve perunike z dodatnimi podatki



`Draw/Network+First Partition+First Vector+Second Vector`

Velikost vozlišč je sorazmerna normaliziranima (Sepal.Length, Sepal.Width) oziroma (Petal.Length, Petal.Width). Barva vozlišč je določena z razbitjem iz podatkov. *Podatki*.

r-sosedi v R-ju

```
r.neighbor2Net <-
# stores network of  r-neighbors (d(v,u) <= r) for
# dissimilarity matrix  d  to file  fnet  in Pajek format.
function(fnet,d,r){
  net <- file(fnet,"w")
  n <- nrow(d); rn <- rownames(d)
  cat("*vertices",n,"\n",file=net)
  for (i in 1:n) cat(i," \"",rn[i],"\"\n",sep="",file=net)
  cat("*edges\n",file=net)
  for (i in 1:n){
    s <- order(d[i,]); j <- 1
    while (d[i,s[j]] <= r) {
      k <- s[j]; if (i < k) cat(i,k,d[i,k],"\n",file=net)
      j <- j+1
    }
  }
  close(net)
}
```

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

Transformacije

Besedni graf – vozlišča so besede; besedi sta povezani, če lahko eno dobimo iz druge s spremembo ene črke. DIC28, *Članek*.

Omrežja iz besedil – besedi sta povezani, če se v besedilu pojavita dovolj blizu skupaj. Utež povezave je število takih ponovitev. Primer *CRA*.

Grafi iger – vozlišča so stanja v igri, povezave pa dovoljeni prehodi med njimi. Omrežja in zabavna matematika.

Z zbiranjem podatkov o mobilnih telefonih ali s spremljanjem značk RFID (Radio-frequency identification) lahko ustvarjamo omrežja (so)delovanj njihovih lastnikov.

Internetska omrežja

KartOO network

Internet Mapping Project. Sosednost na spletu (Najdi.si).
E-mail, blogi, strežniški dnevniki in druge storitve.
Povezave med stranmi na spletu: KartOO, TouchGraph.
Cybergeography, *CAIDA*.
Orodja za pridobivanje podatkov s spleta: *MedlineR*, *SocSciBot*.

Pomenski splet (URI, RDF, OWL). LOD, FreeBase, DBpedia.

Rabutanje

Za pridobivanje izbranih podatkov iz (večih) spletnih strani lahko napišemo posebne programe *web wrappers*. Ti iz posamezne strani izluščijo iskane podatke in jih shranijo – pogosto v obliki XML.
Primeri v R-ju: Naslovi patentov, Knjige z Amazon.
Ker je pisanje teh programov za običajnega uporabnika prezapleteno, je bilo razvitih več *orodij*, ki jih ustvarijo iz uporabnikovih opisov/zahtev (članek / seznam / LAPIS).
Med prostimi orodji sta zanimiva še XWRAP (opis / stran) in TSIMMIS (opis / stran).
Med tržnimi orodji trenutno prevladuje lixto.
Še nekaj naslovov 1, 2, 3.
Nutch, IssueCrawler, W4F.
Python: lxml: nadgradnja etree, dostop do najboljših knjižnic; nadgradnja Beautiful Soup.
Amazon web services, Google Data, Google+, YouTube, Twitter, Last.fm, MusicBrainz3, Flickr, LinkedIn, ...

Omrežje z Amazona

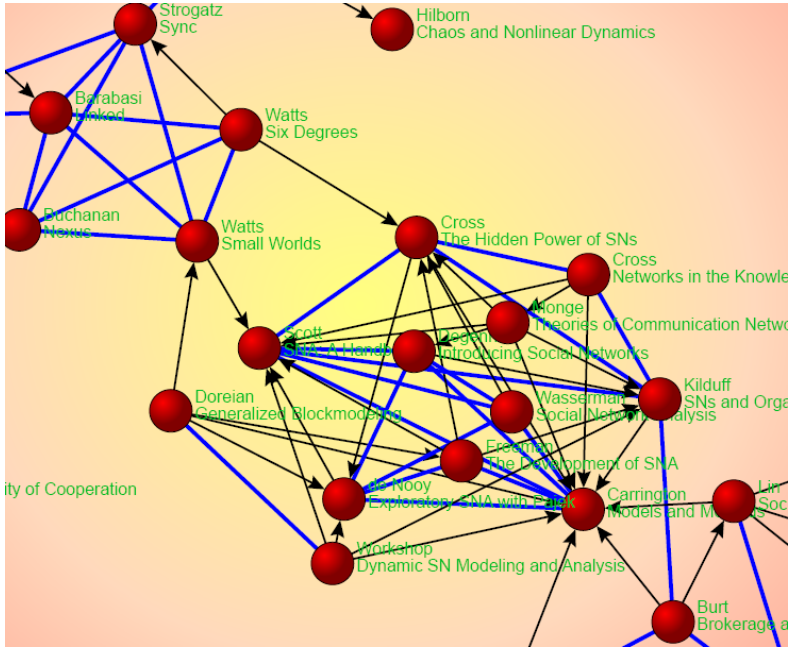
!!! zaradi sprememb zgradbe strani ne deluje več

```
amazon <- function(fvtx,flnk,ftit,maxver){
# ustvari omrezje knjig s spletisca Amazon
#  amazon('v.txt','a.txt','t.txt',10)
# Vladimir Batagelj, 20-21. nov. 2004 / 10. nov. 2006
  opis <- function(line){
    i <- regexpr('\">',line); l <- i[1]+attr(i,"match.length")[1]
    j <- regexpr('</a>',line); r <- j[1]-1; substr(line,l,r)
  }
  vid <- new.env(hash=TRUE,parent=emptyenv())
  vtx <- file(fvtx,"w");  cat('*vertices\n', file=vtx)
  tit <- file(ftit,"w");  cat('*vertices\n', file=tit)
  lnk <- file(flnk,"w");  cat('*arcs\n',file=lnk)
  url1 <- 'http://www.amazon.com/exec/obidos/tg/detail/-/'
  url2 <- '?v=glance';
  book <- '0521840856'
  auth <- "Patrick Doreian"
  titl <- "Generalized Blockmodeling"
  narc <- 0;  nver <- 1
  page <- paste(url1,book,url2,sep='')
  cat(nver, ' "', book, '" URL "',page,'"\n', sep='', file=vtx)
  cat(nver, ' "', auth, ':\\n',titl, '"\n', sep='', file=tit)
  assign(book,nver,env=vid)
  cat('new vertex ',nver,' - ',book,'\n')
  books <- c(book)
```

Omrežje z Amazona

```
  while (length(books)>0){
    bk <- books[1]; books <- books[-1]
    vini <- get(bk,env=vid); cat(vini,'\n')
    page <- paste(url1,bk,url2,sep='')
    stran <- readLines(con<-url(page)); close(con)
    i <- grep("Customers who bought",stran,ignore.case=TRUE)[1]
    if (is.na(i)) break
    j <- grep("Explore Similar Items",stran,ignore.case=TRUE)[1]
    izrez <- stran[i:j]; izrez <- izrez[-which(izrez=="")]
    izrez <- izrez[-which(izrez=="   ")]
    ik <- regexpr("/dp/",izrez); ii <- ik+attr(ik,"match.length")
    for (k in 1:length(ii)) {
      j <- ii[k];
      if (j > 0) {
        bk <- substr(izrez[k],j,j+9); cat('test',k,bk,'\n')
        if (exists(bk,env=vid,inherits=FALSE)){
          vter <- get(bk,env=vid,inherits=FALSE)
        } else {
          nver <- nver + 1; vter <- nver; line <- izrez[k]
          assign(bk,nver,env=vid)
          if (nver <= maxver) {books <- append(books,bk)}
          cat(nver,' "',bk,'" URL "',url1,bk,url2,'"\n',sep='',file=vtx)
          cat('new vertex ',nver,' - ',bk,'\n');
          t <- opis(line); line <- izrez[k+1]
          if (substr(line,1,2)=='by') {a <- substr(line,4,100)}
            else { a <- 'UNKNOWN' }
          cat(nver, ' "', a, ':\\n', t, '"\n', sep='', file=tit)
        }
        narc <- narc + 1; cat(vini,vter,'\n', file=lnk)
      }
    }
    flush.console()
  }
  close(lnk); close(vtx); cat('Amazon - END\n')
}
```

Omrežje z Amazona – knjige o analizi omrežij



Knjige o analizi omrežij z Amazona, 10. november 2006; začetno vozlišče P. Doreian &: Generalized Blockmodeling. Slika SVG. Datoteke/ZIP. Program v R-ju je le zasnova. Možne izpopolnitve: seznam začetnih vozlišč; nadaljevanje po prekinitvi zveze; ...
Najbrž program danes ni več uporaben, ker se zgradba Amazonovih strani občasno spremeni.

Kako do omrežja?
Rodovniki
Molekule
GraphML
Analiza besedil
Odnosi med državami
Dvovrstna omrežja
Multivariatna omrežja
Internet
Slučajna omrežja
Viri

Slučajna omrežja

Omrežja lahko tudi sami ustvarimo z nekim slučajnim postopkom. Ozadja teh postopkov bomo spoznali kasneje.
Vgrajeni so v Pajka (`Network / Create Random network`), lahko pa jih tudi sami zapišemo z razmeroma kratkimi postopki v R-ju.

Na voljo je tudi program `GeneoRnd` za ustvarjanje slučajnih rodovnikov.

Za ustvarjanje slučajnih omrežij, ki pripadajo posebnim razredom omrežij, lahko uporabimo slučajne induktivne razrede [**?**].

Slučajni neusmerjeni Erdős-Rényijev graf

```
dice <- function(n=6){return(1+trunc(n*runif(1,0,1)))}

ErdosRenyiNet <-
# generates a random undirected graph of Erdos-Renyi type
# with n vertices and m edges, and stores it on the file
# fnet  in Pajek's format.
# Example:   ErdosRenyiNet('testER.net',100,175)
# ------------------------------------------------------
# by Vladimir Batagelj, R version: Ljubljana, 20. Dec 2004
# based on ALG.2 from: V. Batagelj, U. Brandes:
#   Efficient generation of large random networks
function(fnet,n,m){
  net <- file(fnet,"w"); cat("*vertices",n,"\n",file=net)
  cat('% random Erdos-Renyi undirected graph G(n,m) / m = ',
    m,'\n',file=net)
#  for (i in 1:n) cat(i," \"v",i,"\"\n",sep="",file=net)
  cat("*edges\n",file=net); L <- new.env(TRUE,NULL)
  for (i in 1:m){
    repeat { u <- dice(n); v <- dice(n)
      if (u!=v) {
        edge <- if (u<v) paste(u,v) else paste(v,u)
        if (!exists(edge,env=L,inherits=FALSE)) break }
    }
    assign(edge,0,env=L); cat(edge,'\n',file=net)
  }
  close(net)
}
```

Dodatni viri

- Batagelj V., Brandes U.: Efficient Generation of Large Random Networks. Physical Review E 71, 036113, 2005.
- Bodlaj, J., Cerinšek, M.: Network Data File Formats. Encyclopedia of Social Network Analysis and Mining. Reda Alhajj, Jon Rokne (Eds.), 2nd Ed, Springer, 2018. PDF
- Cerinšek, M., Batagelj, V.: Sources of Network Data. Encyclopedia of Social Network Analysis and Mining. Reda Alhajj, Jon Rokne (Eds.), 2nd Ed, Springer, 2018. PDF
- V. Hlebec, T. Kogovšek: *Merjenje socialnih omrežij*. Skripta, Študentska založba, Ljubljana, 2006.
- Marsden P.: Network Data and Measurement.
- Kejžar, N., Nikoloski, Z., Batagelj, V.: Probabilistic Inductive Classes of Graphs. Journal of Mathematical Sociology 32: 85-109, 2008

Statistika

Homomorfizmi

Razbitja

Skrčitev

Izrezi

Izrezi

Analiza omrežij

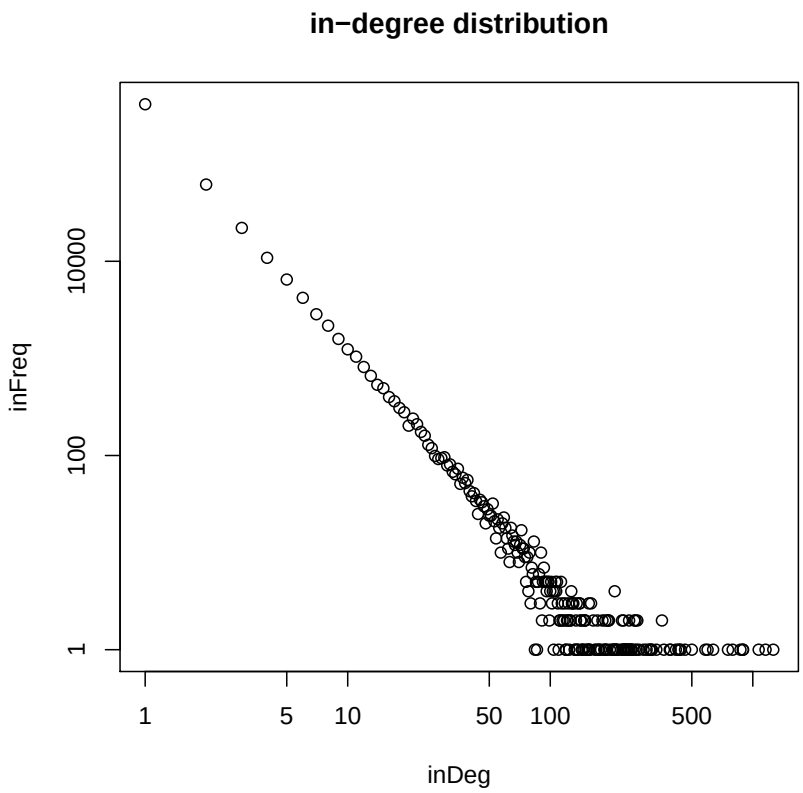
4. Zgradba omrežij podomrežja

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, April 2024

Kazalo

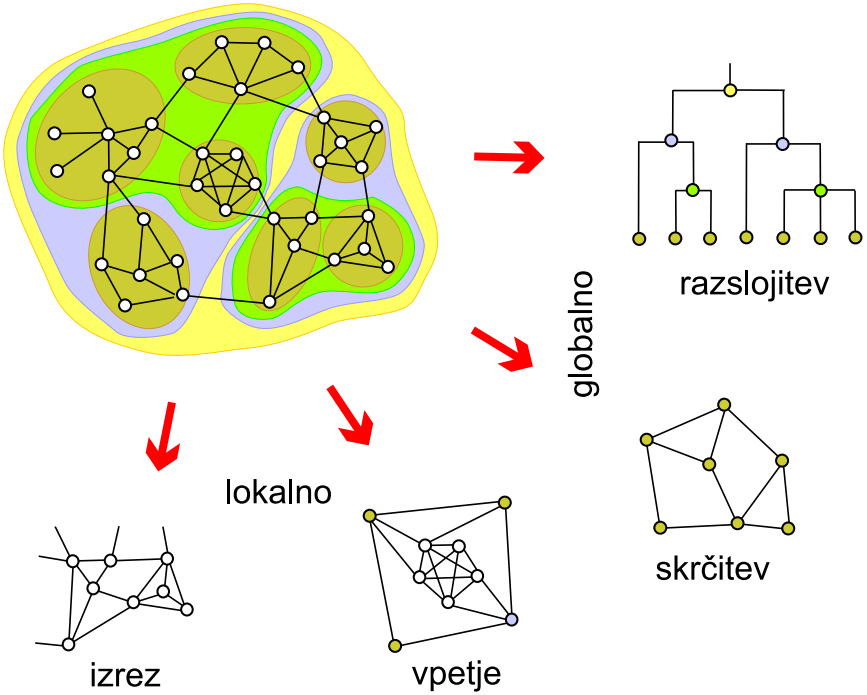
1 Statistika
2 Homomorfizmi
3 Razbitja
4 Skrčitev
5 Izrezi
6 Izrezi



prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

22. april 2024 ob 01 : 17/ marec 2013

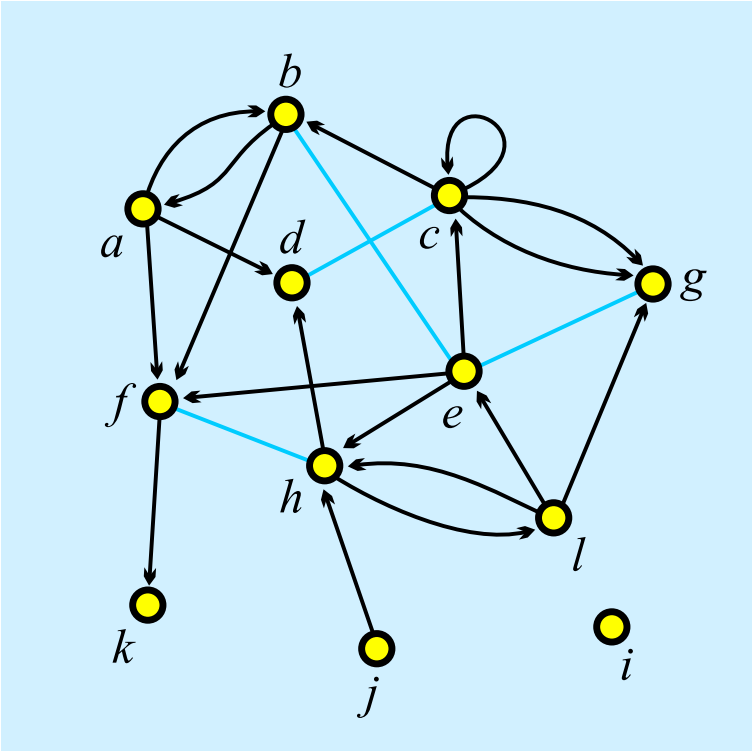
Pristopi k velikim omrežjem



Pri *velikih* omrežjih (več tisoč ali milijonov vozlišč, omrežje je mogoče shraniti v pomnilniku) se moramo odpovedati celoviti sliki, uporabni so le redki postopki.

Za analizo velikih omrežij lahko uporabimo statistiko ali pa zanimiva mala in srednja podomrežja.

Stopnje vozlišč grafa



stopnja vozlišča v, $\deg(v)$ = je število povezav, ki imajo vozlišče v za krajišče;

vhodna stopnja vozlišča v, $\mathrm{indeg}(v)$ = je število povezav, ki imajo vozlišče v za konec (krajišče neusmerjene povezave je hkrati njen začetek in konec);

izhodna stopnja vozlišča v, $\mathrm{outdeg}(v)$ = je število povezav, ki imajo vozlišče v za začetek.

$$n = 12,\ \ m = 23,\ \ \mathrm{indeg}(e) = 3,\ \ \mathrm{outdeg}(e) = 5,\ \ \deg(e) = 6$$

$$\sum_{v\in\mathcal{V}} \mathrm{indeg}(v) = \sum_{v\in\mathcal{V}} \mathrm{outdeg}(v) = |\mathcal{A}| + 2|\mathcal{E}|,\ \ \sum_{v\in\mathcal{V}} \deg(v) = 2|\mathcal{L}| - |\mathcal{L}_0|$$

Statistika

Vhodni podatki o vozliščih

- številski → `vector`
- urejenostni → `permutation`
- imenski → `clustering` (razbitje)

Izračunane lastnosti vozlišč

globalne: število vozlišč, usmerjenih/neusmerjenih povezav, komponent; največje središčno število, ...

lokalne: stopnje, središčna števila, indeksi (vmesnost, dostopnost, kazala in viri, ...)

pregledi: razbitja, vektorji, vrednosti povezav, ...

Analiza povezanosti med izračunanimi (strukturnimi) lastnostmi in vhodnimi (izmerjenimi) lastnostmi.

... Statistika

Globalne lastnosti izpišejo `Pajekovi` ukazi v poročilo; največ jih je dosegljivih v izbiri `Info`. Pri uporabi *ponavljajočih* ukazov se shranijo v vektorje.

Lokalne lastnosti izračunajo razni `Pajekovi` ukazi in jih shranijo v vektorje ali razbitja. Njihove vrednosti / porazdelitev si lahko ogledamo v izbiri `Info`.

Za primer si oglejmo omrežje The Edinburgh Associative Thesaurus. EAT je omrežje asociacij med besedami zbranih na študentski populaciji. Točke so besede. Povezave (X, Y) pa so določene z vprašanjem: Katera beseda Y vam pride prva na misel, ko slišite besedo X ? Utež povezave pove, kolikokrat je bila izbrana.

```
File/Network/Read  eatRS.net
Info/Network/General
```

Ima 23219 vozlišč in 325624 usmerjenih povezav (564 zank); 227481 povezav ima utež 1.

... Statistika

Točke z največjimi stopnjami dobimo takole:

```
Network/Create Partition/Degree/All
Partition/Copy to Vector
Info/Vector  +10
```

V EAT so to:

	vertex	deg	label
1	12720	1108	ME
2	12459	1074	MAN
3	8878	878	GOOD
4	18122	875	SEX
5	13793	803	NO
6	13181	799	MONEY
7	23136	732	YES
8	15080	723	PEOPLE
9	13948	720	NOTHING
10	22973	716	WORK

Pajek in R

Pajek 0.89 (in kasnejši) omogoča uporabo statističnega programa R in tudi drugih programov kot orodij (izbira Tools).

V programu Pajek določimo stopnje vozlišč in jih 'podtaknemo' R-ju

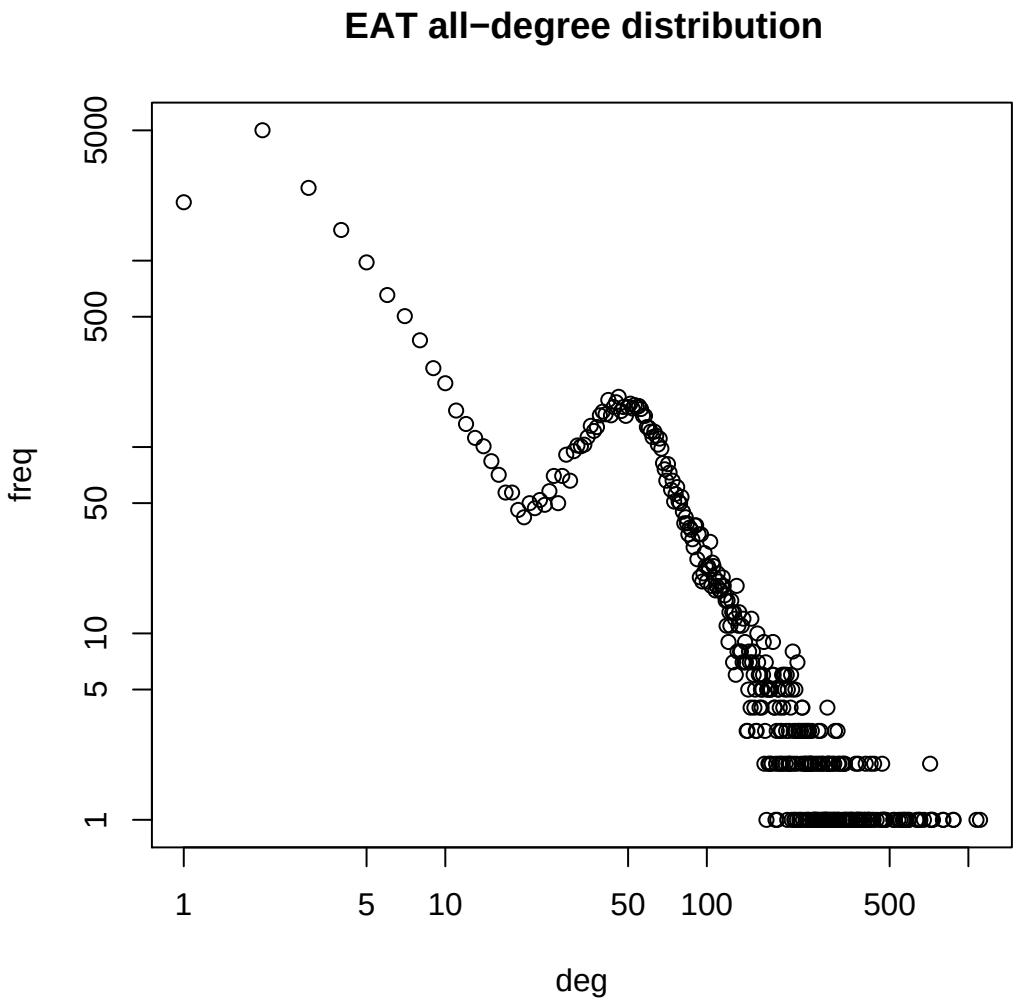
```
Network/Info/General
Network/Create Vector/Centrality/Degree/All
Tools/R/Send to R/Current Vector
```

Tu določimo porazdelitev stopenj in jo narišemo

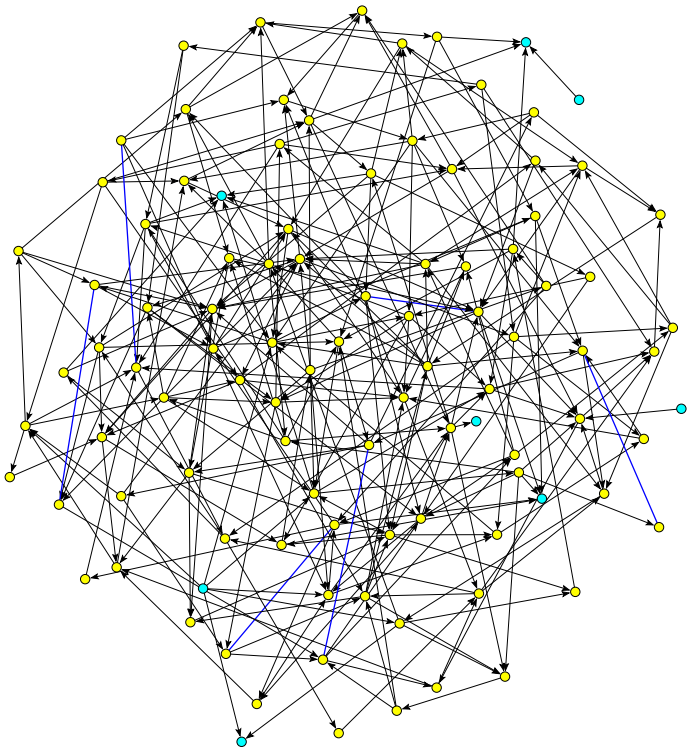
```
summary(v2)
t <- table(v2)
x<-as.numeric(names(t))
plot(x,t,log='xy',main='degree distribution',
  xlab='deg',ylab='freq')
```

Dobljeno sliko lahko s File/Save as shranimo v izbrani obliki (PDF ali PS za LaTeX; Windows metafile za Word).
Pozor! Vozlišča stopnje 0 delajo težave pri log='xy'.

EAT – porazdelitev stopenj



Slučajni grafi



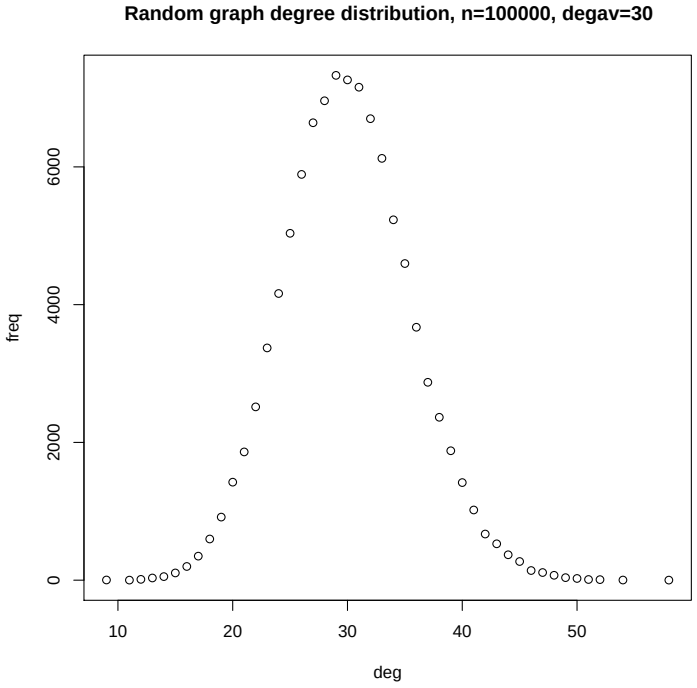
Erdős in Rényi sta definirala *slučajni graf* takole: vsako mogočo povezavo vključimo v slučajni graf z dano verjetnostjo p.

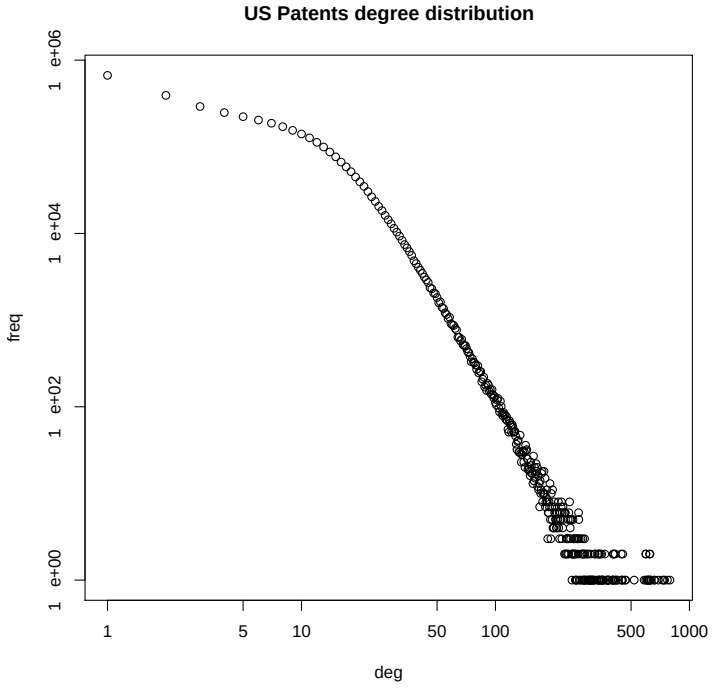
V programu `Pajek` v `Network/Create Random Network/Bernoulli/Poisson/Undirected/General [100] [2.5]` uporabljamo namesto verjetnosti p povprečno stopnjo

$$\overline{\deg} = \frac{1}{n} \sum_{v \in \mathcal{V}} \deg(v)$$

Velja $p = \frac{m}{m_{max}}$ in, za grafe brez zank, še $\overline{\deg} = \frac{2m}{n}$.
Na sliki je prikazan slučajni graf na 100 vozliščih z $\overline{\deg} = 3$.

Porazdelitve stopenj





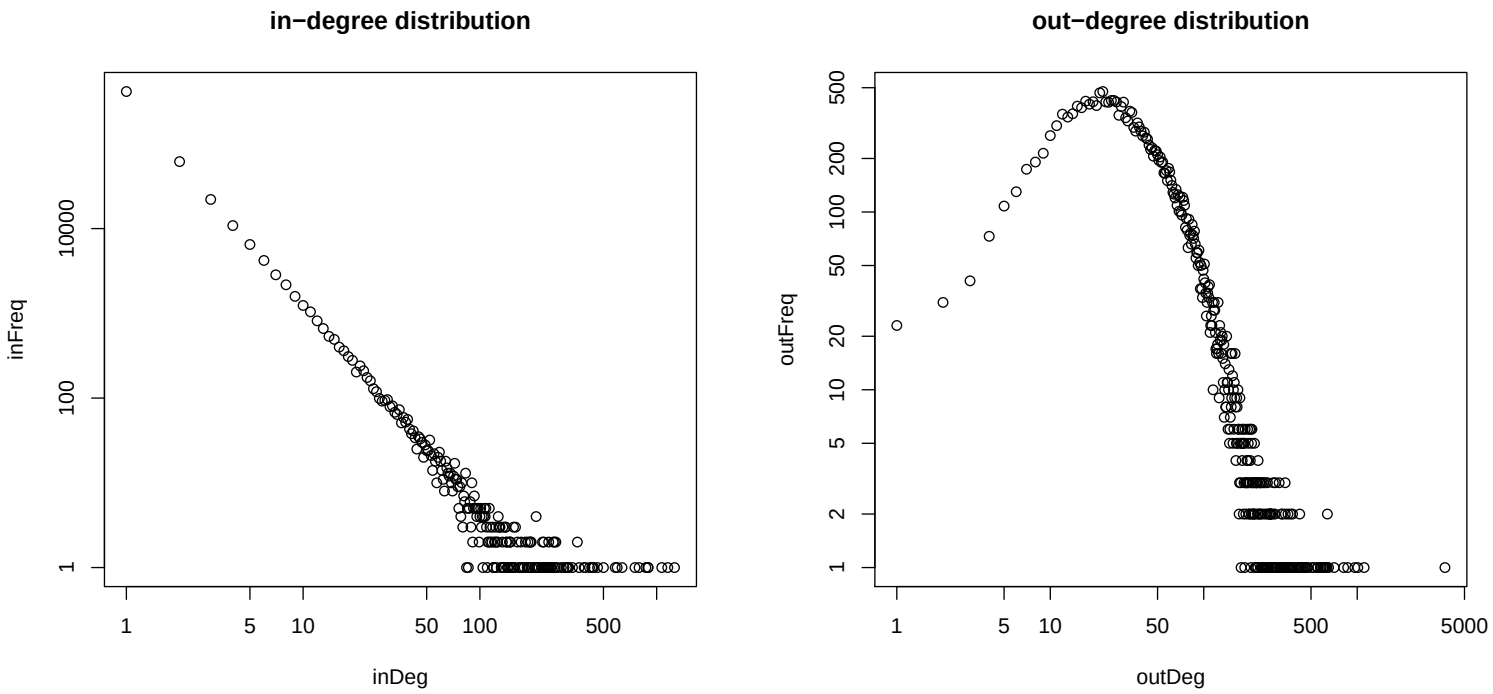
Dejanska omrežja so vse prej kot slučajna. Analiza porazdelitev je dala nov pogled na zgradbo dejanskih omrežij – Watts (Small worlds), Barabási (nd/networks, Linked).

Porazdelitve v/iz-hodnih stopenj

V Pajek-a preberemo omrežje sklicevanj cite.net in odstranimo zanke ter večkratne povezave. Nato določimo vhodne in izhodne stopnje ter iz Pajek-a pokličemo R in mu 'podtaknemo' vse vektorje.

```
#######################################
 R called from Pajek
 The following vectors read:
 v3  :  From partition 1 (548600)
 v4  :  From partition 2 (548600)
-----------------------------------------------------------------
> inTab <- table(v3)
> indeg <- as.integer(names(inTab))
> inDeg <- indeg[indeg>0]
> inFreq <- as.vector(inTab[indeg>0])
> plot(inDeg,inFreq,log='xy',main="in-degree distribution")
> ouTab <- table(v4)
> outdeg <- as.integer(names(ouTab))
> outDeg <- outdeg[outdeg>0]
> outFreq <- as.vector(ouTab[outdeg>0])
> plot(outDeg,outFreq,log='xy',main="out-degree distribution")
```

Porazdelitve v/iz-hodnih stopenj



Porazdelitev vhodnih stopenj kaže na brezlestvičnost (scale-free). Parametre lahko ocenimo s paketom, ki so ga pripravili Clauset, Shalizi and Newman. Glejte še Stumpf, et al.: Critical Truths About Power Laws.

Števila objav člankov po letih

omrežja na temo središčnosti

Iz datoteke `Year.clu`, ki vsebuje letnice objav člankov, lahko dobimo porazdelitve *števila objav člankov po letih*. Za omrežja na temo središčnosti smo dobili:

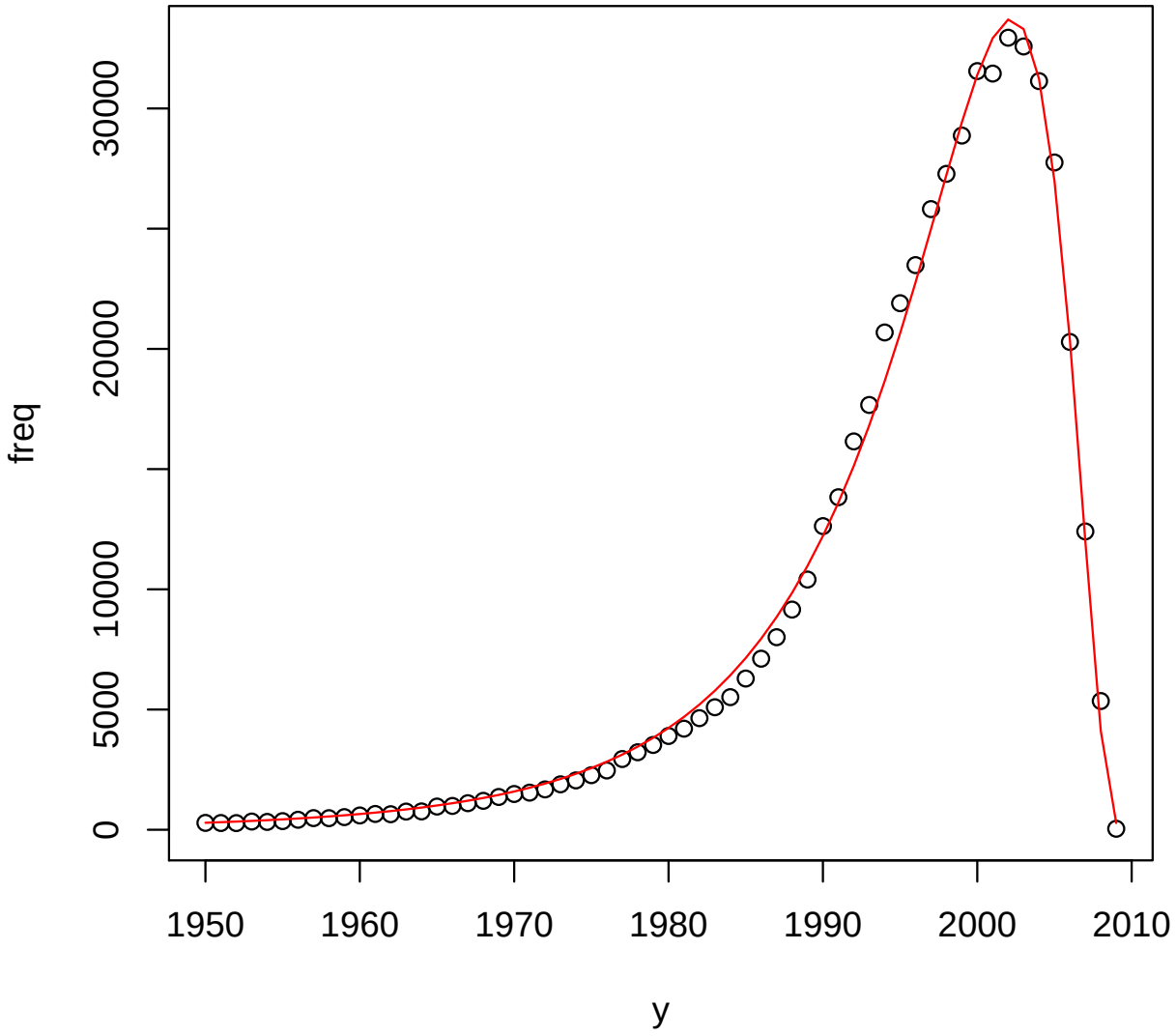
```
> setwd("C:/Users/Batagelj/work/Python/WoS/Central")
> years <- read.table(file="Year.clu",header=FALSE,skip=2)$V1
> t <- table(years)
> year <- as.integer(names(t))
> freq <- as.vector(t[1950<=year & year<=2009])
> y <- 1950:2009
> plot(y,freq)
> model <- nls(freq~c*dlnorm(2010-y,a,b),start=list(c=350000,a=2,b=0.7))
> model
Nonlinear regression model
  model:  freq ~ c * dlnorm(2010 - y, a, b)
   data:  parent.frame()
         c          a          b
5.427e+05 2.491e+00 6.624e-01
 residual sum-of-squares: 20474181

Number of iterations to convergence: 7
Achieved convergence tolerance: 3.978e-06
> lines(y,predict(model,list(x=2010-y)),col='red')
```

Porazdelitev lahko dobro povzamemo *logaritemsko normalno porazdelitvijo*, toda tudi s funkcijo $c * (x + d)^{\frac{a}{b+x}}$.

Števila objav člankov po letih

omrežja na temo središčnosti



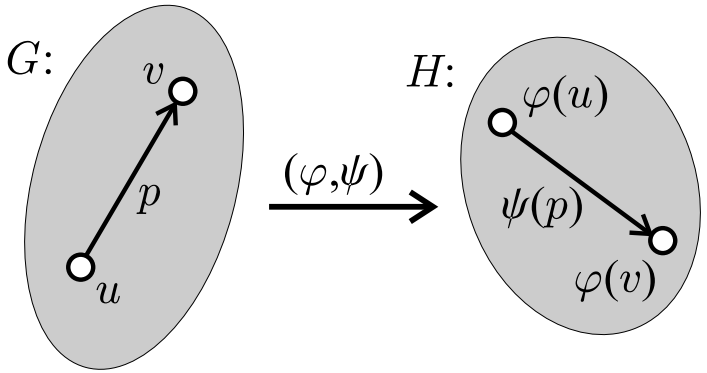
Povezanosti med grafi

Preslikavi (φ, ψ), $\varphi\colon \mathcal{V} \to \mathcal{V}'$ in $\psi\colon \mathcal{L} \to \mathcal{L}'$ določata *šibki homomorfizem* grafa $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ v graf $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ ntk velja:

$$\forall u, v \in \mathcal{V}\, \forall p \in \mathcal{L} : (p(u : v) \Rightarrow \psi(p)(\varphi(u) : \varphi(v)))$$

in določa *(krepki) homomorfizem* grafa $\mathcal{G}$ v graf $\mathcal{H}$ ntk velja:

$$\forall u, v \in \mathcal{V}\, \forall p \in \mathcal{L} : (p(u, v) \Rightarrow \psi(p)(\varphi(u), \varphi(v)))$$

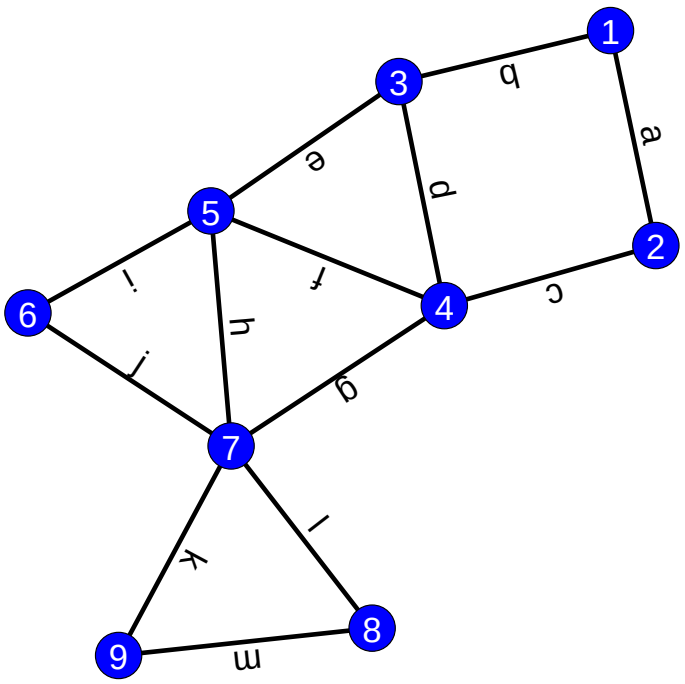


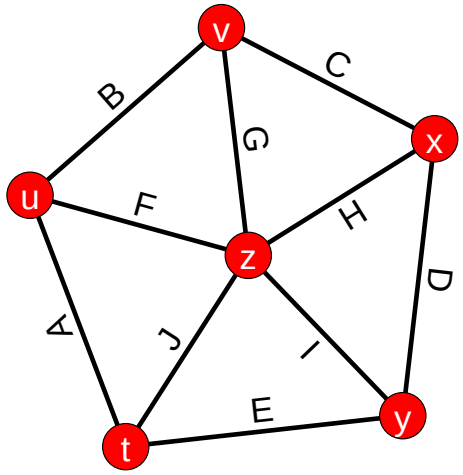
Ko sta φ in ψ bijekciji in ustrezni pogoj velja v obe smeri, govorimo o *izomorfizmu* grafov $\mathcal{G}$ in $\mathcal{H}$. Da sta grafa šibko izomorfna zapišemo $\mathcal{G} \sim \mathcal{H}$; da sta (krepko) izomorfna pa $\mathcal{G} \approx \mathcal{H}$. Velja $\approx \subset \sim$.

Stalnica ali *invarianta* grafa imenujemo vsako grafu prirejeno število, ki je enako za vse med seboj izomorfne grafe. .

EulerGT

Homomorfizem

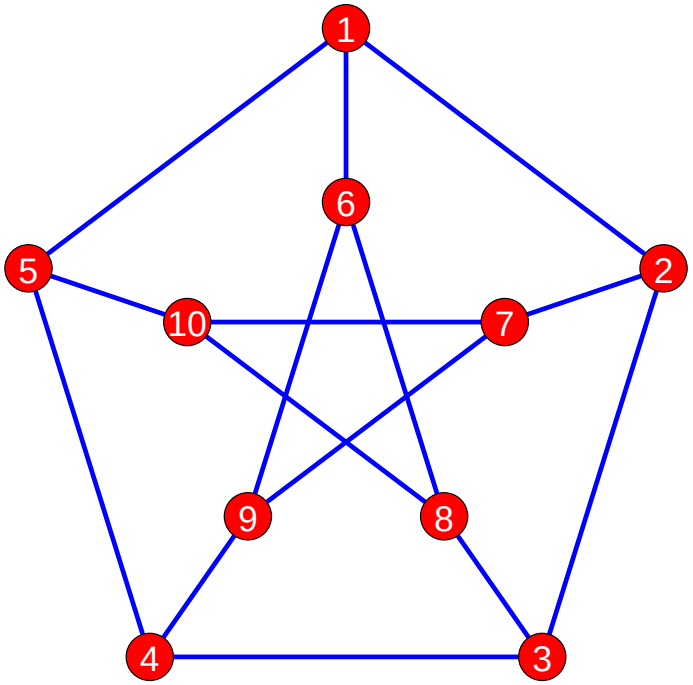


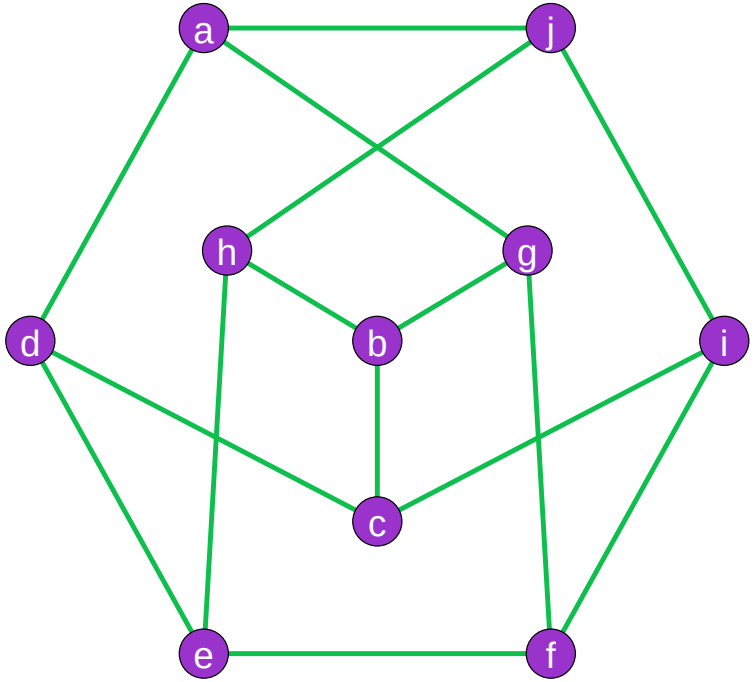


φ	1	2	3	4	5	6	7	8	9
	x	y	t	z	x	z	t	z	x

ψ	a	b	c	d	e	f	g	h	i	j	k	l	m
	D	E	H	J	E	I	J	E	I	J	E	J	I

Izomorfna grafa





φ	1	2	3	4	5	6	7	8	9	10
	b	h	j	a	g	c	e	i	d	f

Skupine, razvrstitve, razbitja, razslojitve

Neprazno podmnožico $C \subseteq \mathcal{V}$ imenujemo *skupina*. Neprazna množica skupin $\mathbf{C} = \{C_i\}$ je *razvrstitev*.
Razvrstitev $\mathbf{C} = \{C_i\}$ je *razbitje* ntk

$$\cup\mathbf{C} = \bigcup_i C_i = \mathcal{V} \quad \text{in} \quad i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

Razvrstitev $\mathbf{C} = \{C_i\}$ je *razslojitev* ali *hierarhija* ntk

$$C_i \cap C_j \in \{\emptyset, C_i, C_j\}$$

Razslojitev $\mathbf{C} = \{C_i\}$ je *polna*, če je $\cup\mathbf{C} = \mathcal{V}$; in je *osnovna*, če je za vsak $v \in \cup\mathbf{C}$ tudi $\{v\} \in \mathbf{C}$.

Primer razbitja in razslojitve

Množica vozlišč:

$$\mathcal{V} = \{a, b, c, d, e, f, g\}$$

Razbitje:

$$\mathbf{C} = \{\{a, b, e\}, \{c, g\}, \{d, f\}\}$$

Skupina:

$$C_2 = \{c, g\}$$

Razslojitev:

$$\mathbf{H} = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{a, e\}, \{c, g\}, \{d, f\}, \{a, b, e\}, \{c, d, f, g\}, \{a, b, c, d, e, f, g\}\}$$

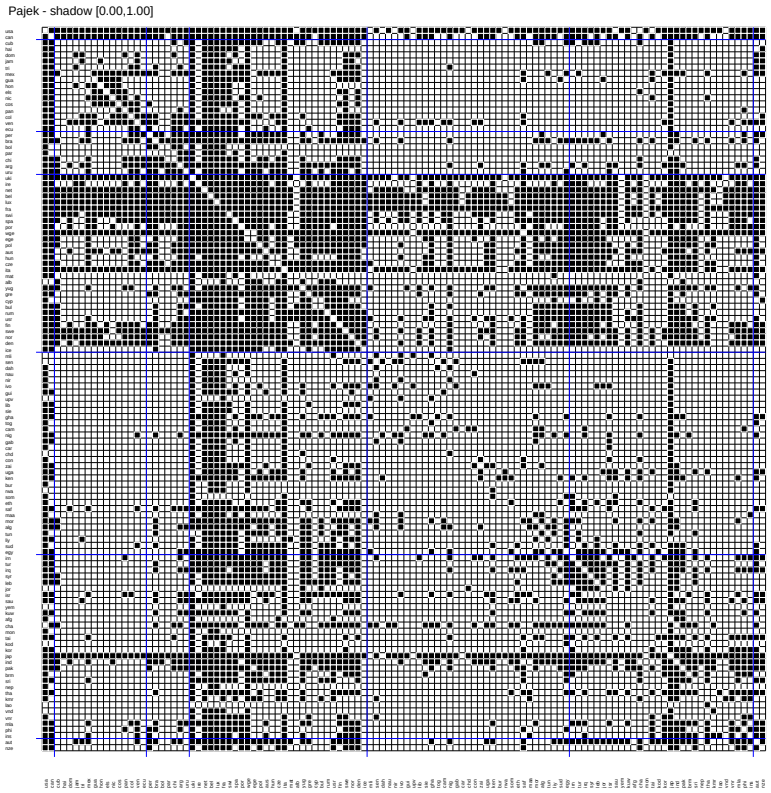
Skrčitev skupine

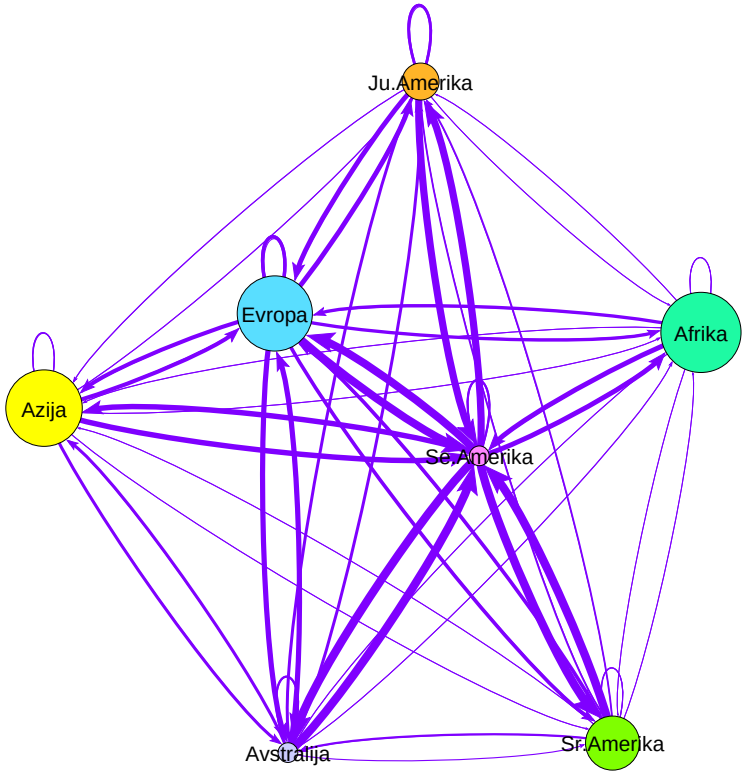
Skrčitev skupine C imenujemo graf $\mathcal{G}/C$, ki ga dobimo tako da vsa vozlišča skupine C zamenjamo z enim vozliščem, recimo c. Natančneje $\mathcal{G}/C = (\mathcal{V}', \mathcal{L}')$, kjer je $\mathcal{V}' = (\mathcal{V} \setminus C) \cup \{c\}$ in $\mathcal{L}'$ sestavljajo povezave iz $\mathcal{L}$, ki imajo obe krajišči v $\mathcal{V} \setminus C$. Večkratne vzporedne povezave združimo v eno. Poleg teh pa še 'zvezda' z vrhom c in krakom (v, c), če $\exists p \in \mathcal{L}, u \in C : p(v, u)$, oziroma krakom (c, v), če $\exists p \in \mathcal{L}, u \in C : p(u, v)$. V vozlišču c je zanka (c, c), če $\exists p \in \mathcal{L}, u, v \in C : p(u, v)$.

V omrežju nad grafom $\mathcal{G}$ moramo povedati še, kako so določene vrednosti/uteži v skrčenem delu. Običajno kar kot vsota ali maksimum/minimum izvornih vrednosti.

`Operations/Network + Partition/Shrink Network`

Skrčitev skupin – trgovanje med državami





Snyder in Kickovi podatki o trgovanju med državami. Matrični prikaz gostih omrežij. Uteži v skrčitvi (makro):

$$w(C_i, C_j) = \frac{n(C_i, C_j)}{n(C_i) \cdot n(C_j)}$$

Izračun uteži w

```
File/Pajek Project File/Read [SKtrade.paj]
Network/Create New Network/Transform/Remove/Loops [No]
Network/Create New Network/Transform/Edges -> Arcs [No]
Operations/Network+Partition/Shrink Network [1 0]
```

	1	2	3	4	5	6	7	Label
1.	2	30	13	56	42	45	4	#usa
2.	30	74	25	196	20	37	12	#cub
3.	12	28	33	124	16	36	5	#per
4.	55	217	130	694	427	483	41	#uki
5.	42	8	14	406	122	117	11	#mli
6.	43	37	43	444	142	307	30	#irn
7.	4	4	5	39	9	30	2	#aut

```
Partition/Make Permutation
[select partition (Sub)continents]
Operations/Partition+Permutation/
  Functional Composition Partition*Permutation
Partition/Count
```

count	2	15	7	29	33	30	2

... Izračun uteži w

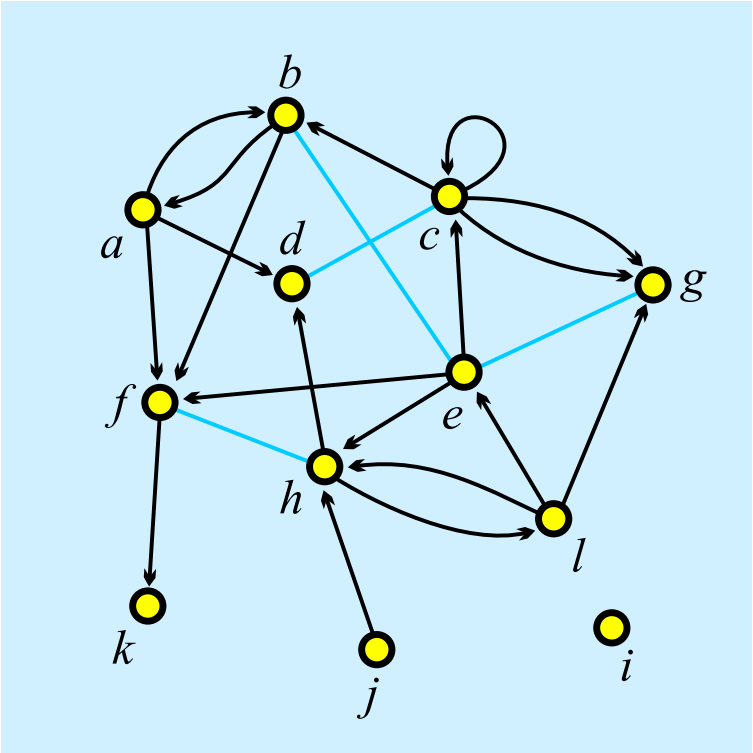
```
Partition/Copy to Vector
Vector/Create Constant Vector [7 1.0]
[select as second vector Copy of partition ...]
Vectors/Divide (First/Second)
Network/Create Vector/Get Loops
Vectors/Add (First+Second)
Operations/Network+Vector/Transform/Put Loops/as Arcs
[select vector Divide V? by ...]
Operations/Network+Vector/Vector#Network/input
Operations/Network+Vector/Vector#Network/output
```

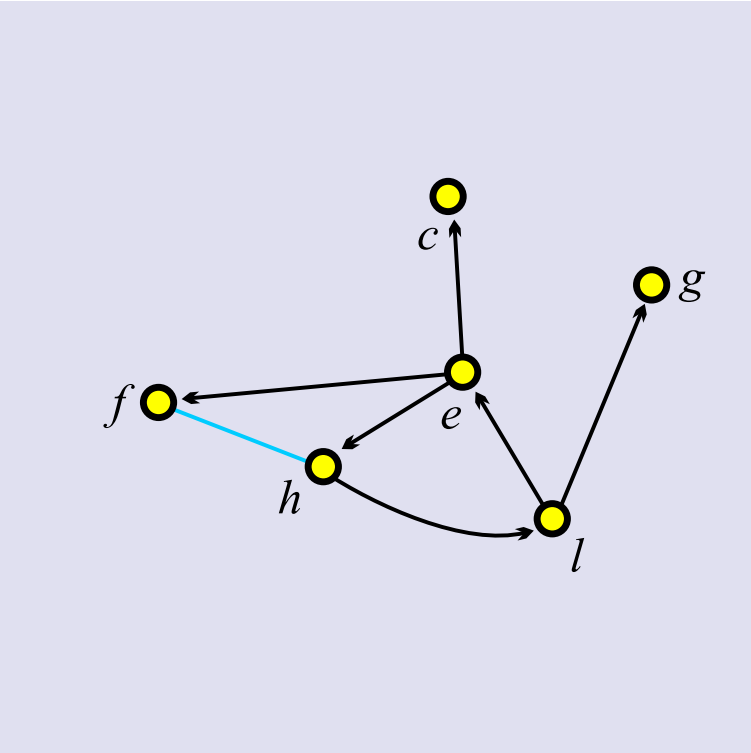
		1	2	3	4	5	6	7
#usa	1.	0.50	1.00	0.93	0.97	0.64	0.75	1.00
#cub	2.	1.00	0.33	0.24	0.45	0.04	0.08	0.40
#per	3.	0.86	0.27	0.67	0.61	0.07	0.17	0.36
#uki	4.	0.95	0.50	0.64	0.83	0.45	0.56	0.71
#mli	5.	0.64	0.02	0.06	0.42	0.11	0.12	0.17
#irn	6.	0.72	0.08	0.20	0.51	0.14	0.34	0.50
#aut	7.	1.00	0.13	0.36	0.67	0.14	0.50	0.50

Ker so na diagonali 0, bi jih bilo smiselno pred izračunom postaviti na 1 ali pa diagonalne vrednosti deliti z $n(C_i)(n(C_i)-1)$, če je $n(C_i) > 1$.

Macro `weights`.

Podgraf



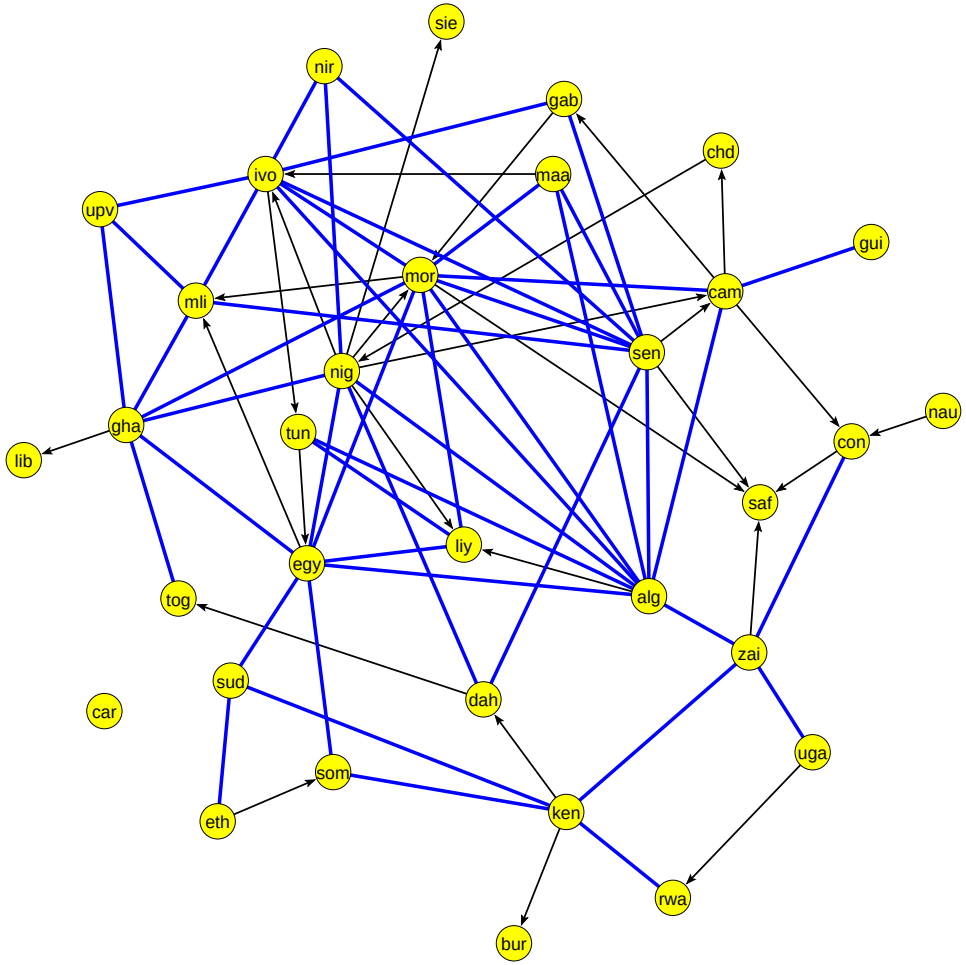


Podgraf $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ danega grafa $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ je graf, katerega povezave $\mathcal{L}'$ so vsebovane v povezavah grafa $\mathcal{G}$, $\mathcal{L}' \subseteq \mathcal{L}$, vozlišča $\mathcal{V}'$ pa v vozliščih grafa $\mathcal{G}$, $\mathcal{V}' \subseteq \mathcal{V}$, in vsebujejo tudi vsa krajišča povezav $\mathcal{L}'$.

Podgraf je lahko *porojen* z dano podmnožico vozlišč ali povezav.

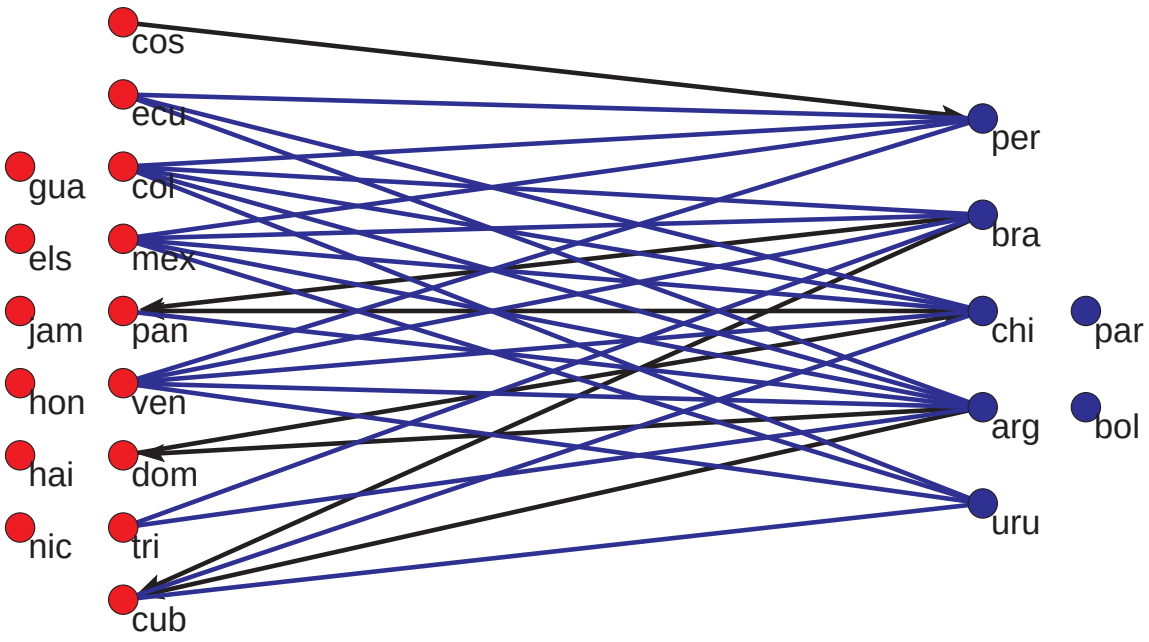
Podgraf je *vpet*, če je $\mathcal{V}' = \mathcal{V}$.

Izrez: Snyder in Kick – Afrika



`Operations/Network + Partition/Extract Subnetwork [6]`

Izrez: Snyder in Kick
Latinska Amerika : Južna Amerika




```
Operations/Network + Partition/Extract Subnetwork  [3,4]
Operations/Network + Partition/Transform/Remove lines/
  Inside clusters   [3,4]
```

Vozlišča lahko ročno razmestimo po pravokotni mreži ustvarjeni z

```
[Draw] Move/Grid
```

Izrezi

Vozliščni izrez omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \to \mathbb{R}$, na *ravni* t je podomrežje $\mathcal{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, določeno z množico vozlišč

$$\mathcal{V}' = \{v \in \mathcal{V} : p(v) \geq t\}$$

kjer je $\mathcal{L}(\mathcal{V}')$ množica vseh povezav iz $\mathcal{L}$, ki imajo obe krajišči v $\mathcal{V}'$.

Povezavni izrez omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \to \mathbb{R}$, na *ravni* t je določeno z množico povezav

$$\mathcal{L}' = \{e \in \mathcal{L} : w(e) \geq t\}$$

To je podomrežje $\mathcal{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, kjer je $\mathcal{V}(\mathcal{L}')$ množica vseh krajišč povezav iz $\mathcal{L}'$.

Analiza omrežja z izrezi

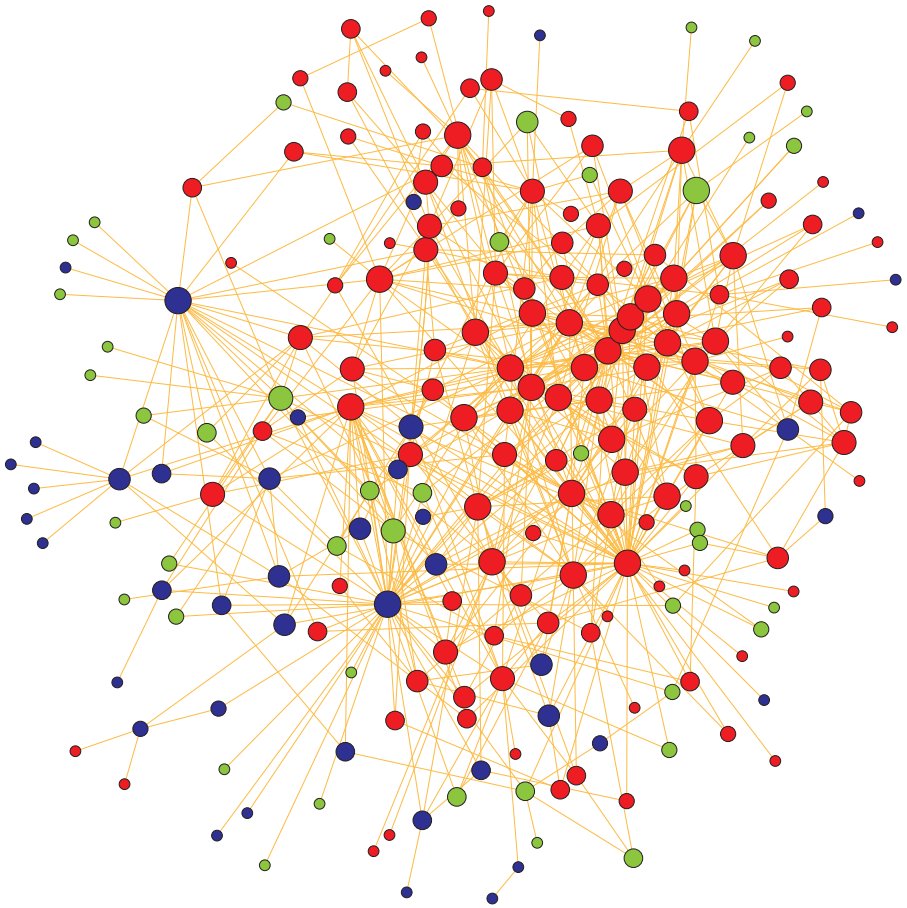
Izrezi ponujajo enostaven pristop k analizi omrežij. Za izbrano lastnost/utež in raven t določimo pripadajoči izrez $\mathcal{N}(t)$. Pozornost posvetimo njegovim komponentam – povezanim delom.

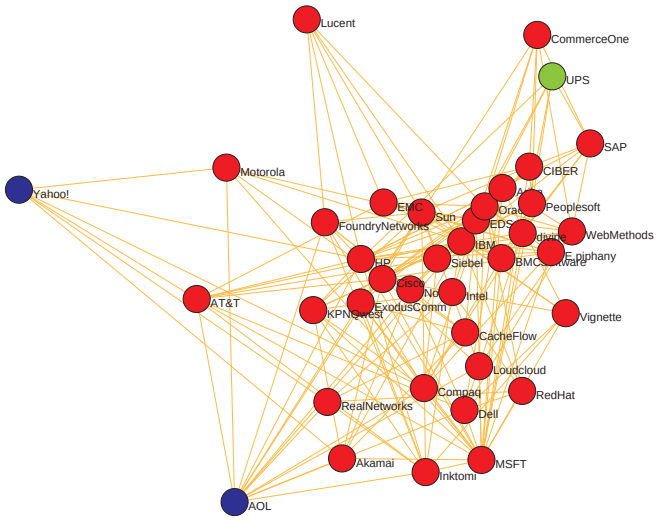
Število in velikost komponent je odvisna od ravni t. Pogosto se pojavi več majhnih komponent. Pri analizi nas običajno zanimajo le komponente 'prave' velikosti – vsaj k in ne večje kot K. Premajhne komponente zavržemo kot 'nezanimive'; prevelike komponente pa ponovno izrežemo na neki višji ravni.

Vrednost t, k in K določimo s pregledom porazdelitve vrednosti lastnosti/uteži in z upoštevanjem dodatnega vedenja o značilnostih omrežja in ciljih raziskave.

V program `Pajek` je vgrajenih nekaj novih, učinkovito izračunljivih lastnosti/uteži.

Vozliščni izrez: Krebsova spletna podjetja, core=6

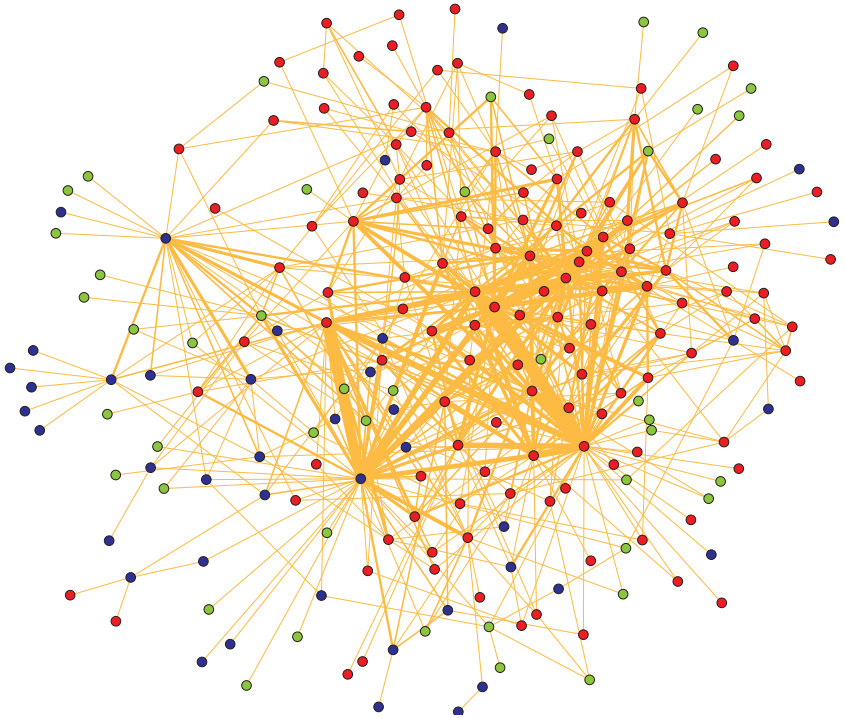


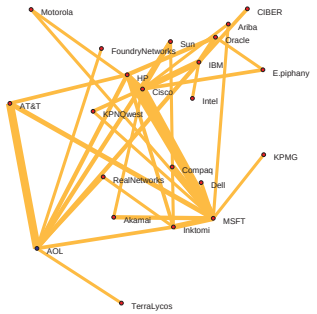


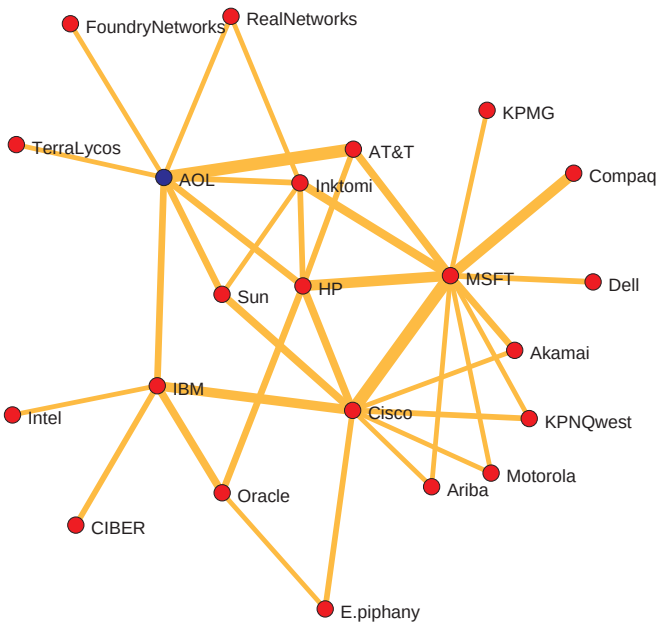
Vsako vozlišče predstavlja spletno podjetje dejavno v obdobju 1998 do 2001. $n = 219$, $m = 631$. rdeča – vsebina, modra – podporne storitve, zelena – trgovina.

Podjetji sta povezani, če sta najavili skupni posel ali drugo obliko sodelovanja.

Povezavni izrez: Krebsova spletna podjetja, $w_3 \geq 5$







Analiza omrežja z izrezi

Vrednosti ravni t določimo na osnovi porazdelitve vrednosti funkcij w oziroma p. Običajno nas zanimajo komponente izreza, ki niso niti prevelike, niti premajhne.

Vozliščni izrez: p shranjena v vektorju

```
Vector/Info  [+10] [#10]
Vector/Make Partition/by Intervals/Selected Thresholds  [t]
Operations/Network + Partition/Extract Subnetwork  [2]
```

Povezavni izrez: omrežje z utežmi

```
Network/Info/Line values   [#10]
Network/Create New Network/Transform/Remove/Lines with Value/
  lower than   [t]
Network/Create Partition/Degree/All
Operations/Network + Partition/Extract Subnetwork  [1-*]
```

Analiza omrežij

5. Zgradba omrežij povezanosti

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, april 2024

Kazalo

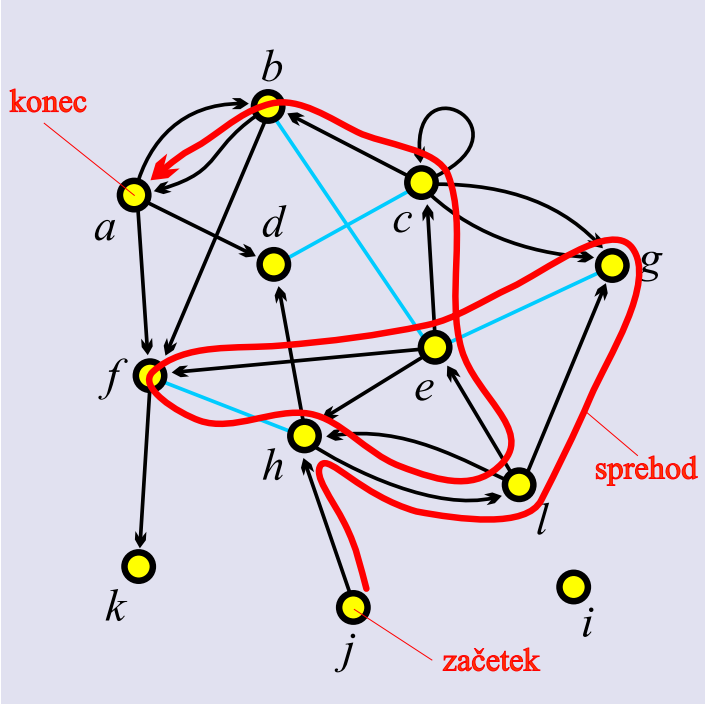
1 Sprehodi
2 Povezanosti
3 Pomembna vozlišča

London Tube

prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

30. april 2024 ob 03 : 03/ april 2013

Sprehodi



dolžina $|s|$ sprehoda s je število povezav, ki ga sestavljajo.

$s = (j, h, l, g, e, f, h, l, e, c, b, a)$

$|s| = 11$

Sprehod je *sklenjen* ali *obhod* ntk. njegov začetek in konec sovpadata.

Če ne upoštevamo smeri povezav v 'sprehodu', dobimo *polsprehod* ali *verigo*.

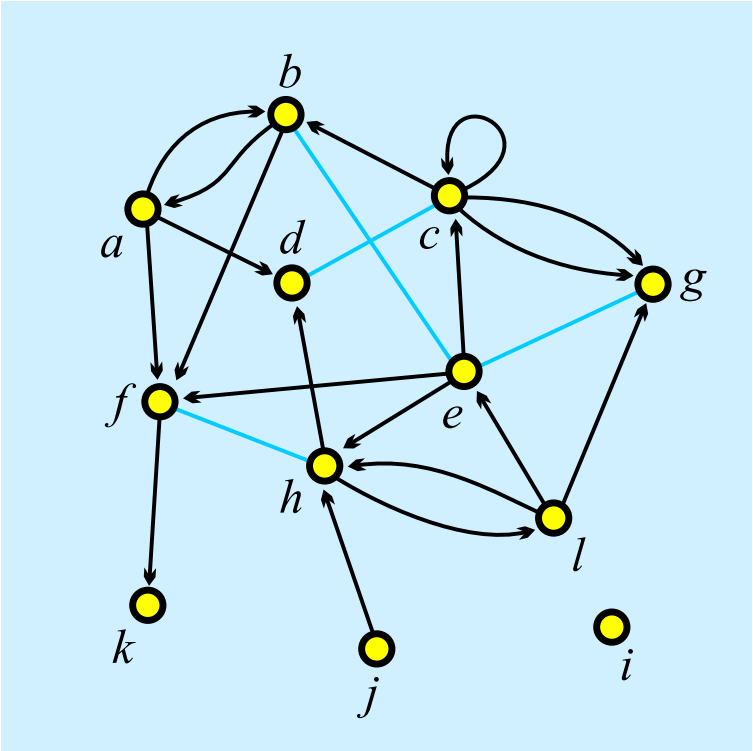
sled – sprehod z različnimi povezavami

pot – sprehod z različnimi vozlišči

cikel – sklenjen sprehod z različnimi notranjimi vozlišči.

Graf je *acikličen*, ntk. ne vsebuje nobenega cikla.

Najkrajše poti



Dolžino najkrajše poti iz u v v označimo z $d(u, v)$.
Če ne obstaja sprehod iz u v v, postavimo $d(u, v) = \infty$.

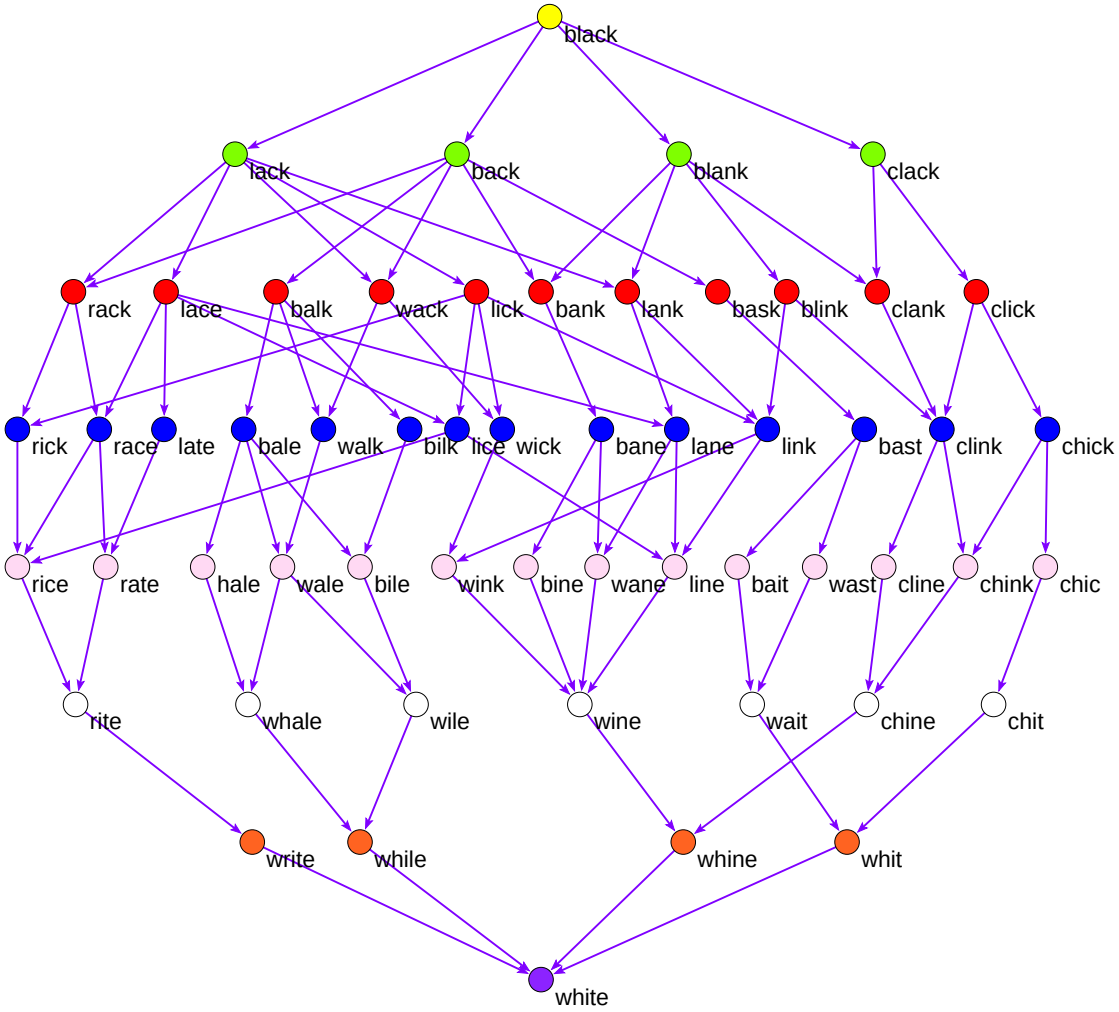
$d(j, a) = |(j, h, d, c, b, a)| = 5$
$d(a, j) = \infty$

$\hat{d}(u, v) = \max(d(u, v), d(v, u))$
je *razdalja*:
$\hat{d}(v, v) = 0$, $\hat{d}(u, v) = \hat{d}(v, u)$,
$\hat{d}(u, v) \leq \hat{d}(u, t) + \hat{d}(t, v)$.

Premer grafa je enak razdalji med, glede na $d(u, v)$, najoddaljenejšima vozliščema: $D = \max_{u,v \in \mathcal{V}} d(u, v)$.

`Network/Create New Network/Subnetwork with Paths/`

Najkrajše poti



DICT28.

Enakovrednosti in razbitja

Relacija R na $\mathcal{V}$ je *enakovrednost* (ekvivalenčna) ntk. je *refleksivna* $\forall v \in \mathcal{V} : vRv$, *simetrična* $\forall u, v \in \mathcal{V} : (uRv \Rightarrow vRu)$ in *tranzitivna* $\forall u, v, z \in \mathcal{V} : uRz \wedge zRv \Rightarrow uRv$.

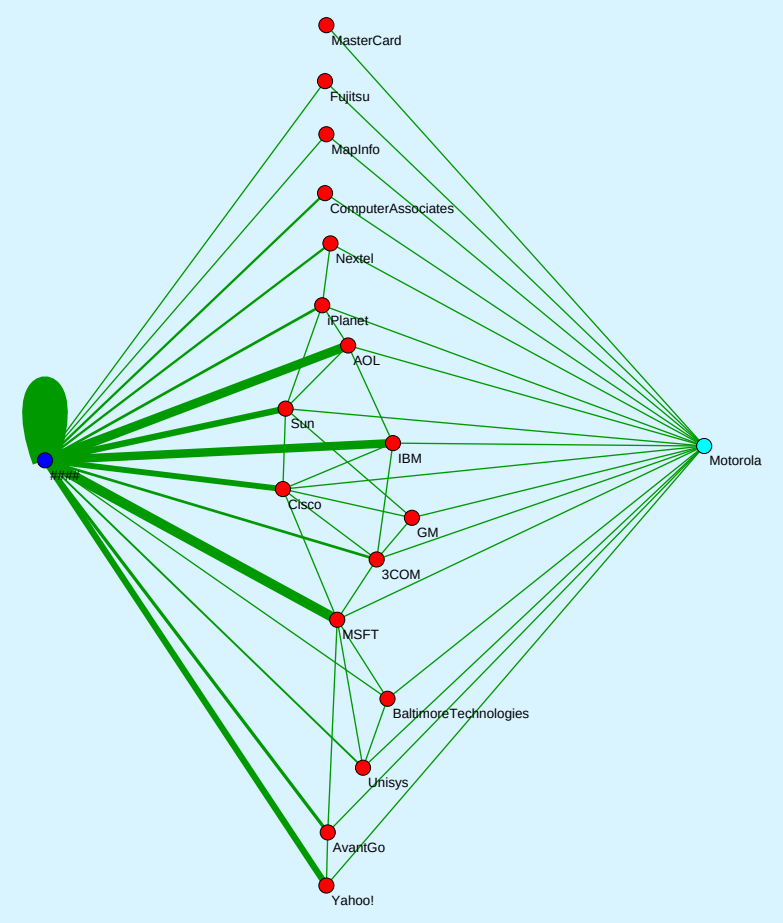
Vsaka enakovrednost R določa neko razbitje v *razrede* $[v] = \{u : vRu\}$.

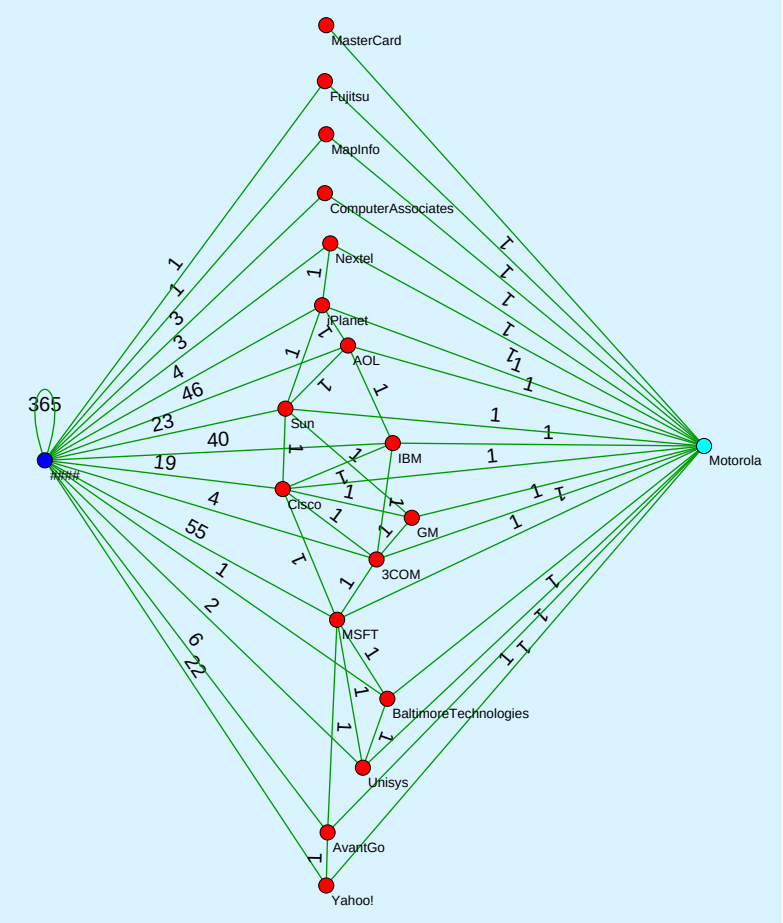
Vsako razbitje $\mathbf{C}$ določa neko enakovrednost $uRv \Leftrightarrow \exists C \in \mathbf{C} : u \in C \wedge v \in C$.

k-sosedi vozlišča v je množica vozlišč, ki so za k oddaljena od v, $N^k(v) = \{u \in v : d(v, u) = k\}$.
Množica vseh množic k-sosedov, $k = 0, 1, \ldots$ vozlišča v je razbitje množice $\mathcal{V}$.
k-soseščina vozlišča v, $N^{(k)}(v) = \{u \in v : d(v, u) \leq k\}$.

`Network/Create Partition/k-Neighbors/`

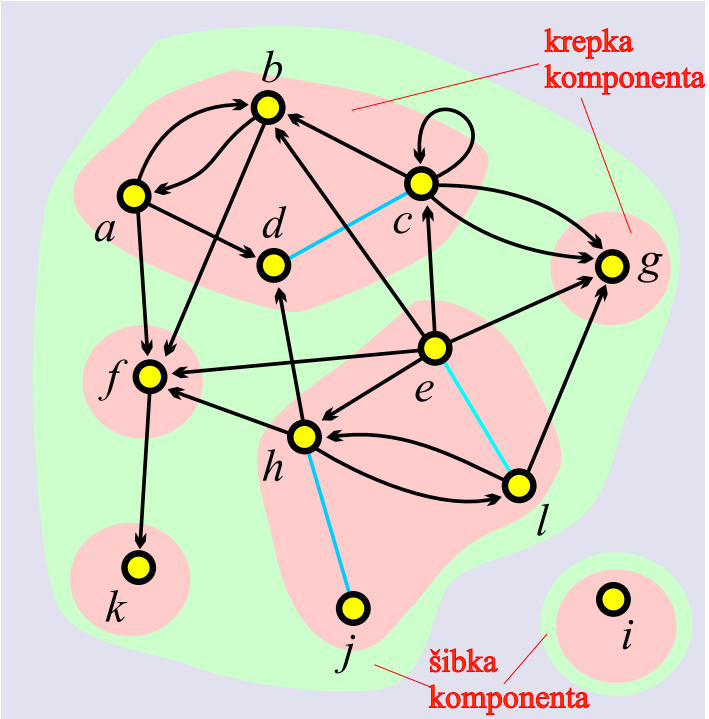
Soseščina Motorole





Debelina povezav je koren iz vrednosti.

Povezanosti



Vozlišče u je *dosegljivo* iz vozlišča v ntk. obstaja sprehod z začetkom v in koncem u.

Vozlišče v je *šibko povezano* z vozliščem u ntk. obstaja veriga s krajiščema v in u.

Vozlišče v je *krepko povezano* z vozliščem u ntk. sta vzajemno dosegljivi.

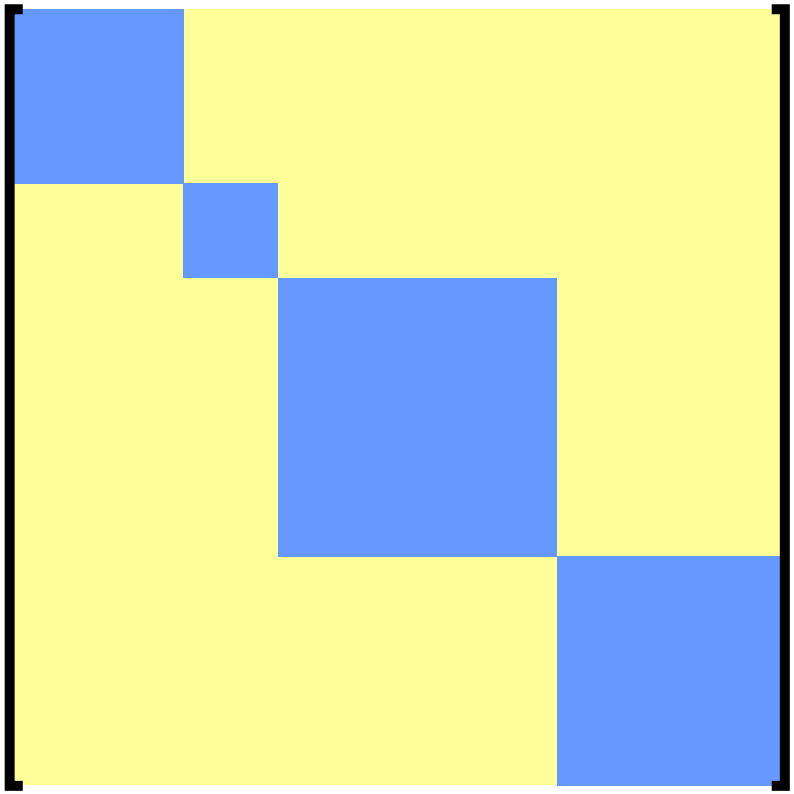
Šibka in krepka povezanost sta enakovrednosti.

`graphCon.net`

Razredi porajajo šibke/krepke *komponente* ali *kose* grafa.

`Network/Create Partition/Components/`

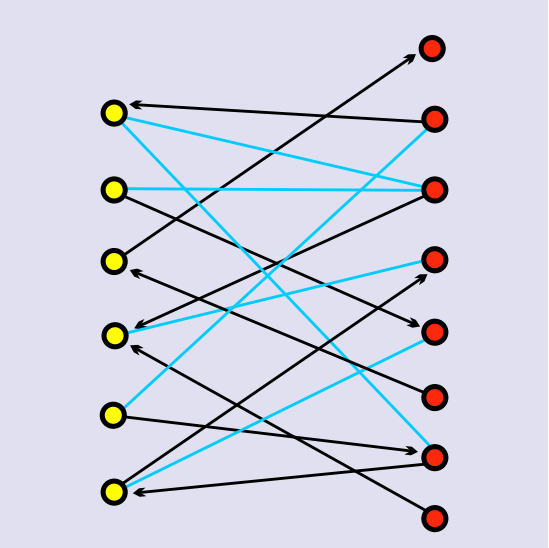
Šibke komponente

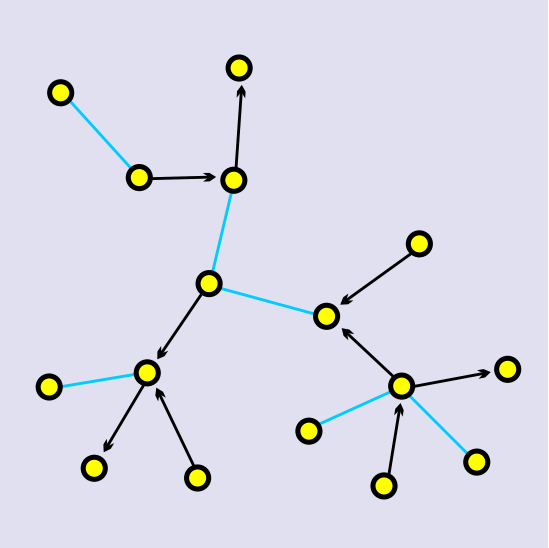


Če preuredimo vozlišča omrežja, tako da vozlišča iz iste skupine šibkega razbitja postavimo skupaj, dobimo matrični prikaz sestavljen iz diagonalnih blokov – šibkih komponent.

Za večino problemov velja, da jih lahko ločeno rešimo za vsako šibko komponento posebej in nato dobljene rešitve združimo v rešitev za celotno omrežje.

Posebni grafi – dvodelni, drevo

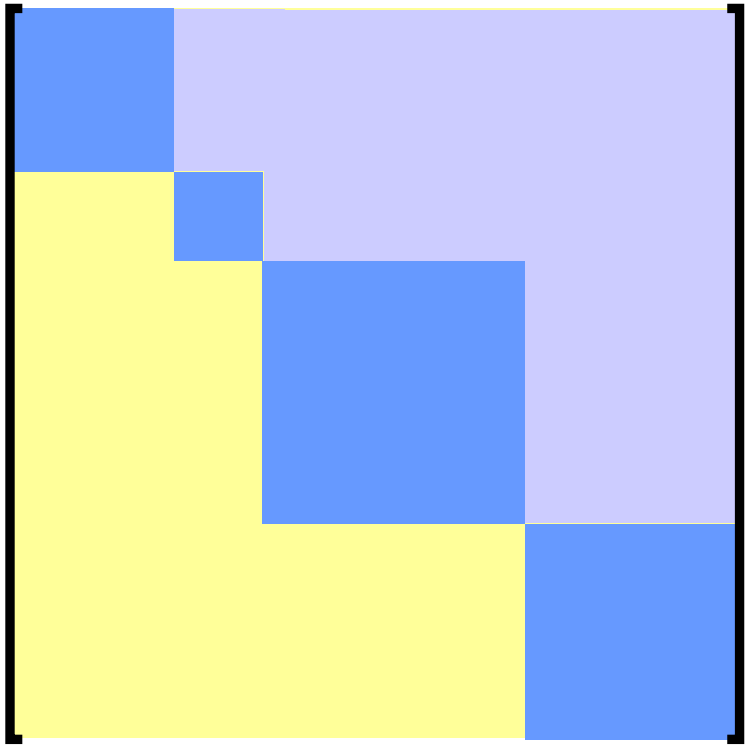


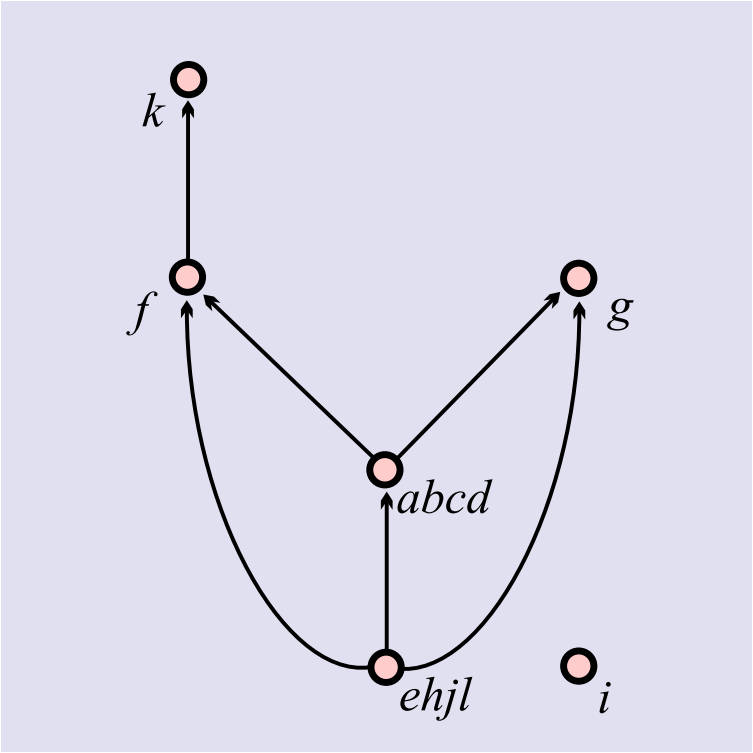


Graf $G = (\mathcal{V}, \mathcal{L})$ je *dvodelen* ntk. lahko množico vozlišč $\mathcal{V}$ razbijemo na podmnožici V_1 in V_2, tako da ima vsaka povezava iz $\mathcal{L}$ eno krajišče v $\mathcal{V}_1$ drugo pa v $\mathcal{V}_2$.

Šibko povezan graf G je *drevo* ntk. ne vsebuje (zank in) polciklov dolžine vsaj 3.

Krepka skrčitev grafa (kondenzacija)

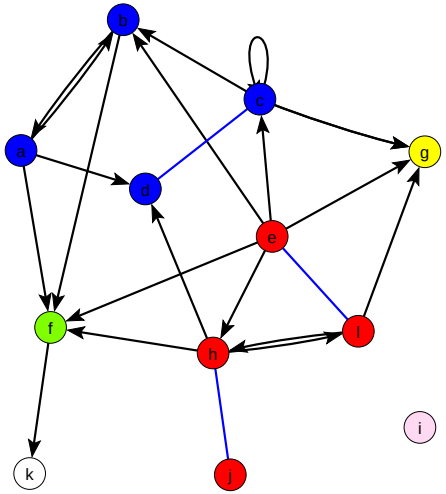


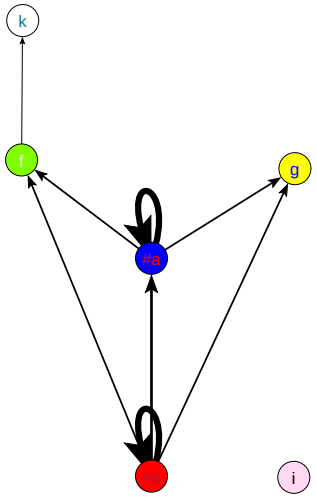


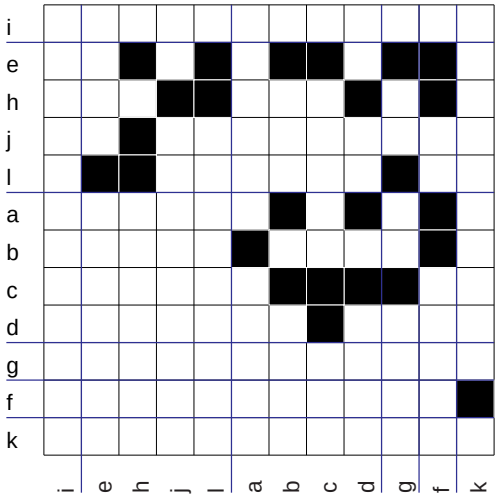
Če v danem grafu skrčimo vsako krepko komponento v ustrezno vozlišče, odstranimo zanke in združimo vzporedne povezave, je tako dobljeni *skrčeni* graf acikličen. Za vsak aciklični graf obstaja *urejenost* / *oštevilčenje* $i : \mathcal{V} \to \mathbb{N}$, tako da velja $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.

Skrčitev grafa – primer

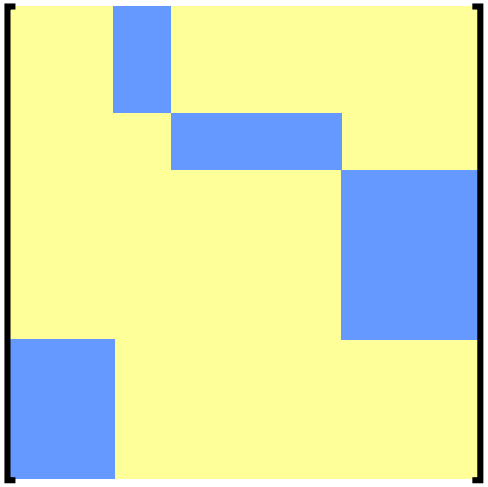
```
Network/Create Partition/Components/Strong [1]
Operations/Network+Partition/Shrink Network  [1][0]
Network/Create New Network/Transform/Remove/Loops [yes]
Network/Acyclic Network/Depth Partition/Acyclic
Partition/Make Permutation
Permutation/Inverse Permutation
select partition [Strong Components]
Operations/Partition+Permutation/Functional Composition/Partition*Permutation
Partition/Make Permutation
select [original network]
File/Network/Export Matrix to EPS/Using Permutation
```

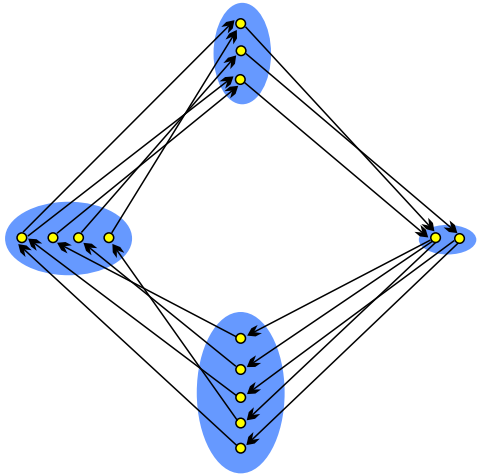






Zgradba krepkih komponent





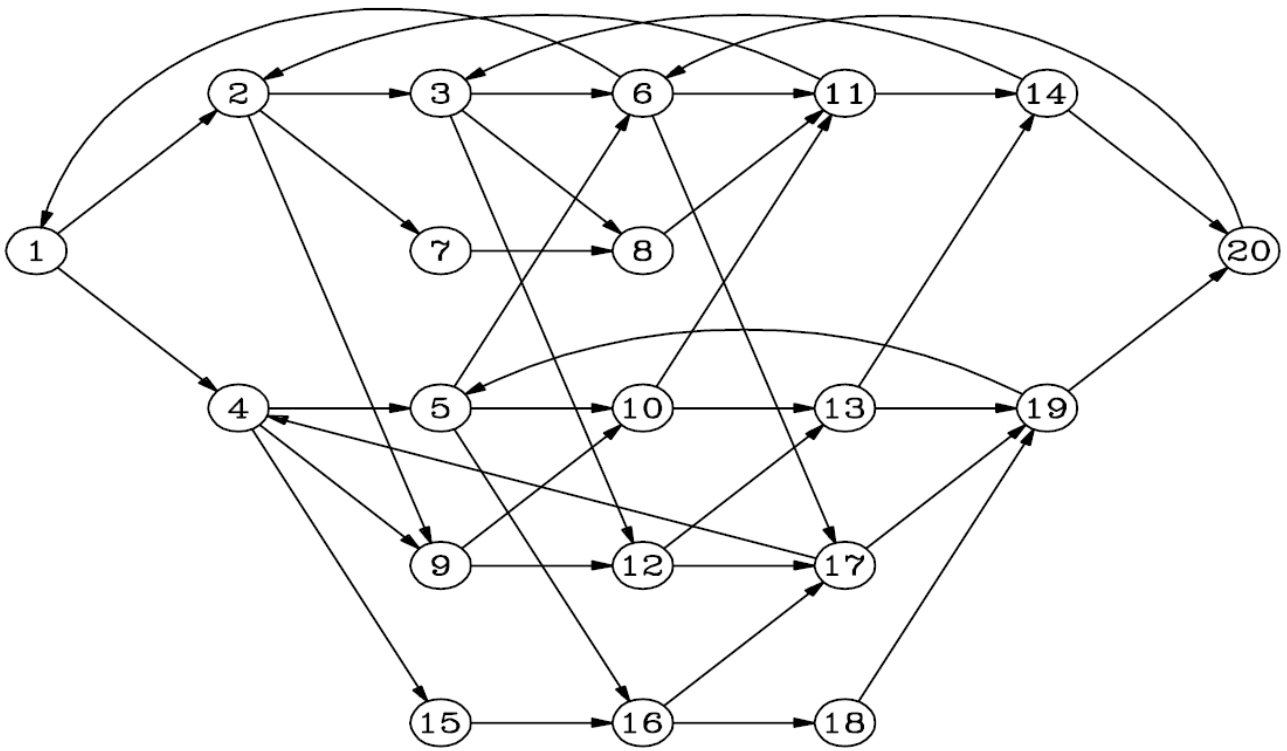
Notranja zgradba krepke komponente – naj bo d največji skupni delitelj dolžin obhodov v krepki komponenti.

Komponenta je *enostavna*, če je $d = 1$; sicer je *periodična* s periodo d.

Množico vozlišč $\mathcal{V}$ krepko povezanega usmerjenega grafa $\mathcal{G} = (\mathcal{V}, R)$ je mogoče razbiti na d skupin $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_d$, tako da za vsako povezavo $(u, v) \in R$ velja $u \in \mathcal{V}_i \Rightarrow v \in \mathcal{V}_{(i \bmod d)+1}$.

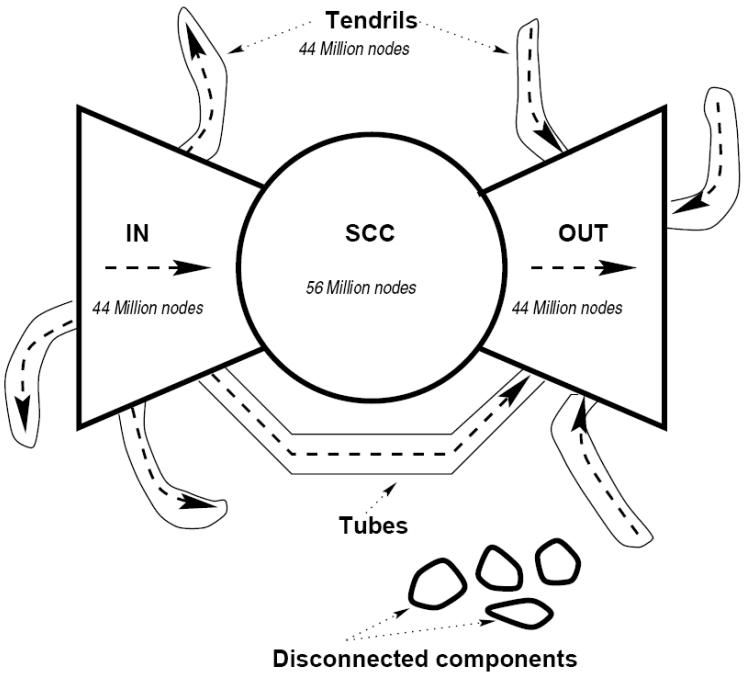
`Network/Create Partition/`
`Components/Strong-Periodic`

... Zgradba krepkih komponent



Bonhourejev periodični graf. Pajek – matrike

Spletni metuljček (Bow-tie)



Kumar &: The Web as a graph

Naj bo $\mathcal{S}$ *największa krepka komponenta* v omrežju $\mathcal{N}$; $\mathcal{W}$ šibka komponenta, ki vsebuje $\mathcal{S}$; $\mathcal{I}$ množica vozlišč, iz katerih je $\mathcal{S}$ dosegljiva; $\mathcal{O}$ množica vozlišč dosegljivih iz $\mathcal{S}$; $\mathcal{T}$ (cevi) vozlišča (niso v $\mathcal{S}$) na poteh iz $\mathcal{I}$ v $\mathcal{O}$; $\mathcal{R} = \mathcal{W} \setminus (\mathcal{I} \cup \mathcal{S} \cup \mathcal{O} \cup \mathcal{T})$ (lovke); in $\mathcal{D} = \mathcal{V} \setminus \mathcal{W}$. Razbitje

$$\{\mathcal{I}, \mathcal{S}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \mathcal{D}\}$$

imenujemo *metuljčno* razbitje $\mathcal{V}$.

Opomba: za splošna usmerjena omrežja slika ne prikazuje vseh možnosti – v množici $\mathcal{R}$ so lahko tudi verige.

`Network/Create Partition/Bow-Tie`

Dvopovezanost

Vozlišči u in v sta *dvopovezani* ntk. sta povezani (v obe smeri) s po dvema neodvisnima (brez skupnih notranjih vozlišč) potema. Dvopovezanost določa razbitje množice **povezav**.

Vozlišče je *stično* vozlišče ali *stičišče* ntk. njegova odstranitev poveča število šibkih komponent grafa.

Povezava je *most* ntk. njena odstranitev poveča število šibkih komponent grafa.

`Network/Create New Network/with Bi-Connected Components`

k-povezanost

Vozliščna povezanost κ grafa G je enaka najmanjšemu številu vozlišč, ki jih je potrebno odvzeti iz grafa, tako da je graf porojen s prestalimi vozlišči nepovezan ali trivialen (enak K_1).

Povezavna povezanost λ grafa G je enaka najmanjšemu številu povezav, ki jih je potrebno odvzeti iz grafa, tako da je graf porojen s prestalimi povezavami nepovezan ali trivialen.

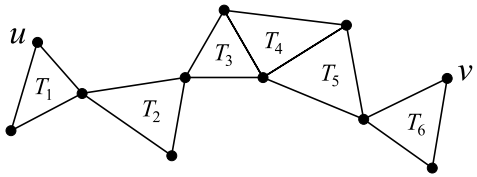
Velja Whitneyeva neenakost: $\kappa(G) \leq \lambda(G) \leq \delta(G)$.

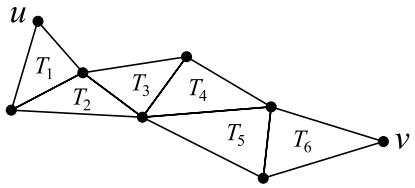
Graf G je *(po vozliščih) $k-$povezan*, če je $\kappa(G) \geq k$ in je *po povezavah $k-$povezan*, če je $\lambda(G) \geq k$.

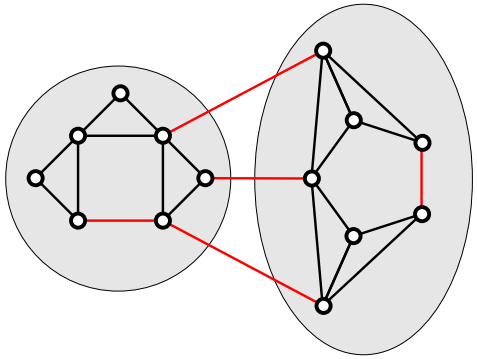
Velja Whitneyeva različica Mengerjevega izreka: Graf G je po vozliščih/povezavah $k-$povezan ntk. vsak par vozlišč povezuje vsaj k po vozliščih/povezavah ločenih sprehodov.

Trikotniška povezanost – neusmerjeni grafi

V neusmerjenem grafu imenujemo *trikotnik* podgraf izomorfen K_3. Zaporedje trikotnikov $(T_1, T_2, \ldots, T_s)$ grafa G *(vozliščno) trikotniško povezuje* vozlišči $u, v \in \mathcal{V}$ ntk. $u \in T_1$ in $v \in T_s$ ali $u \in T_s$ in $v \in T_1$ ter $\mathcal{V}(T_{i-1}) \cap \mathcal{V}(T_i) \neq \emptyset$, $i = 2, \ldots s$; in *povezavno trikotniško povezuje* vozlišči $u, v \in \mathcal{V}$ ntk zadošča še strožji različici zadnjega pogoja $\mathcal{E}(T_{i-1}) \cap \mathcal{E}(T_i) \neq \emptyset$, $i = 2, \ldots s$.

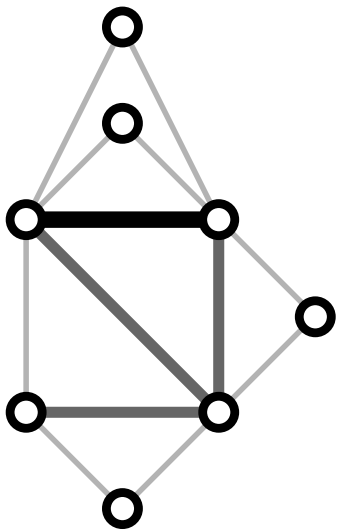






Vozliščna trikotniška povezanost je enakovrednost na vozliščih; povezavna pa na povezavah. Članek.

Trikotniško omrežje

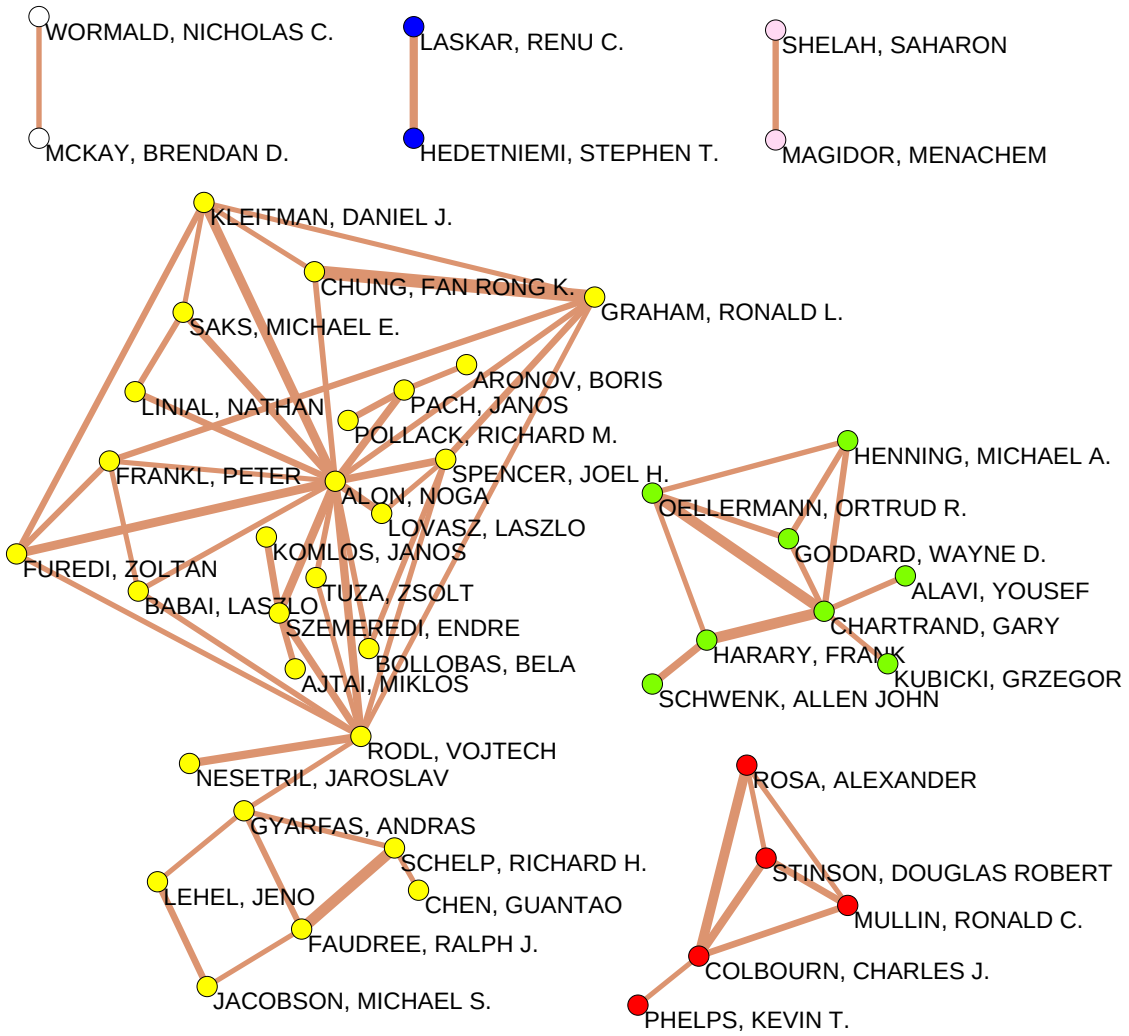


Naj bo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ enostaven neusmerjen graf. Prirejeno *trikotniško omrežje* $\mathcal{N}_T(\mathcal{G}) = (\mathcal{V}, \mathcal{E}_T, w)$ določeno z $\mathcal{G}$ je podgraf $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ grafa $\mathcal{G}$, kjer je E_T množica tistih povezav iz $\mathcal{E}$, ki leže na vsaj enem trikotniku. Utež $w(e)$ povezave $e \in \mathcal{E}_T$ je enaka številu različnih trikotnikov, ki jim povezava e pripada.

Trikotniška omrežja omogočajo učinkovito razkrivanje gostih delov omrežja. Če povezava e pripada *k-kliki* (podgrafu izomorfnemu K_k) v $\mathcal{G}$, je $w(e) \geq k - 2$.

`Network/Create New Network/with Ring Counts/3-Rings/Undirected`

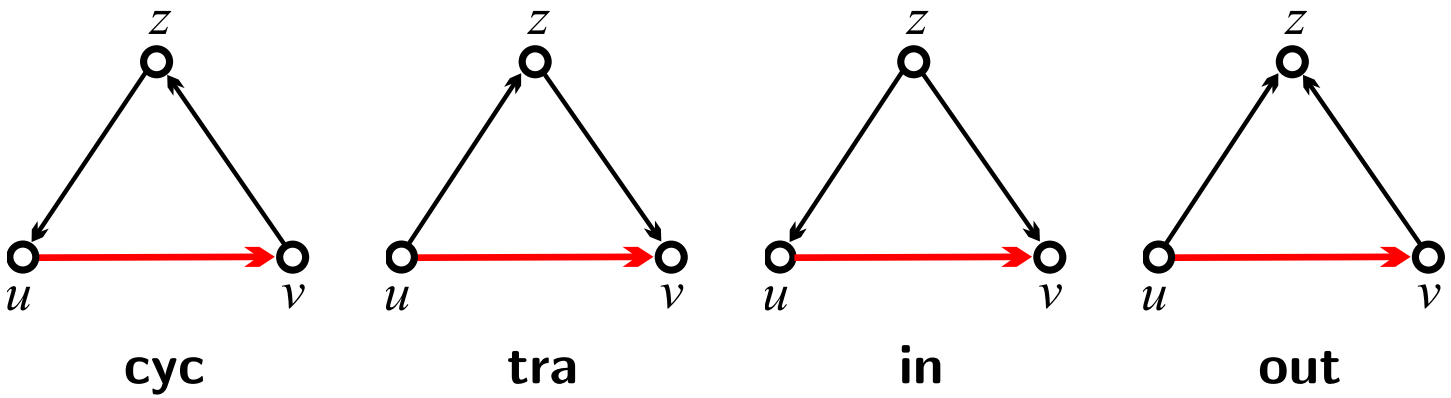
Povezavni izrez na ravni 16 trikotniškega omrežja Erdős-ovega grafa sodelovanj



brez Erdősa,
$n = 6926$,
$m = 11343$

Trikotniška povezanost – usmerjeni grafi

Če je graf $\mathcal{G}$ mešan, zamenjamo neusmerjene povezave s pari nasprotno usmerjenih. Naj bo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ enostaven usmerjen graf brez zank. Za izbrano usmerjeno povezavo $(u, v) \in \mathcal{A}$ obstajajo štiri vrste *usmerjenih trikotnikov*: **cyc**lic, **tra**nsitive, **in**put in **out**put.



... Trikotniška povezanost – usmerjeni grafi

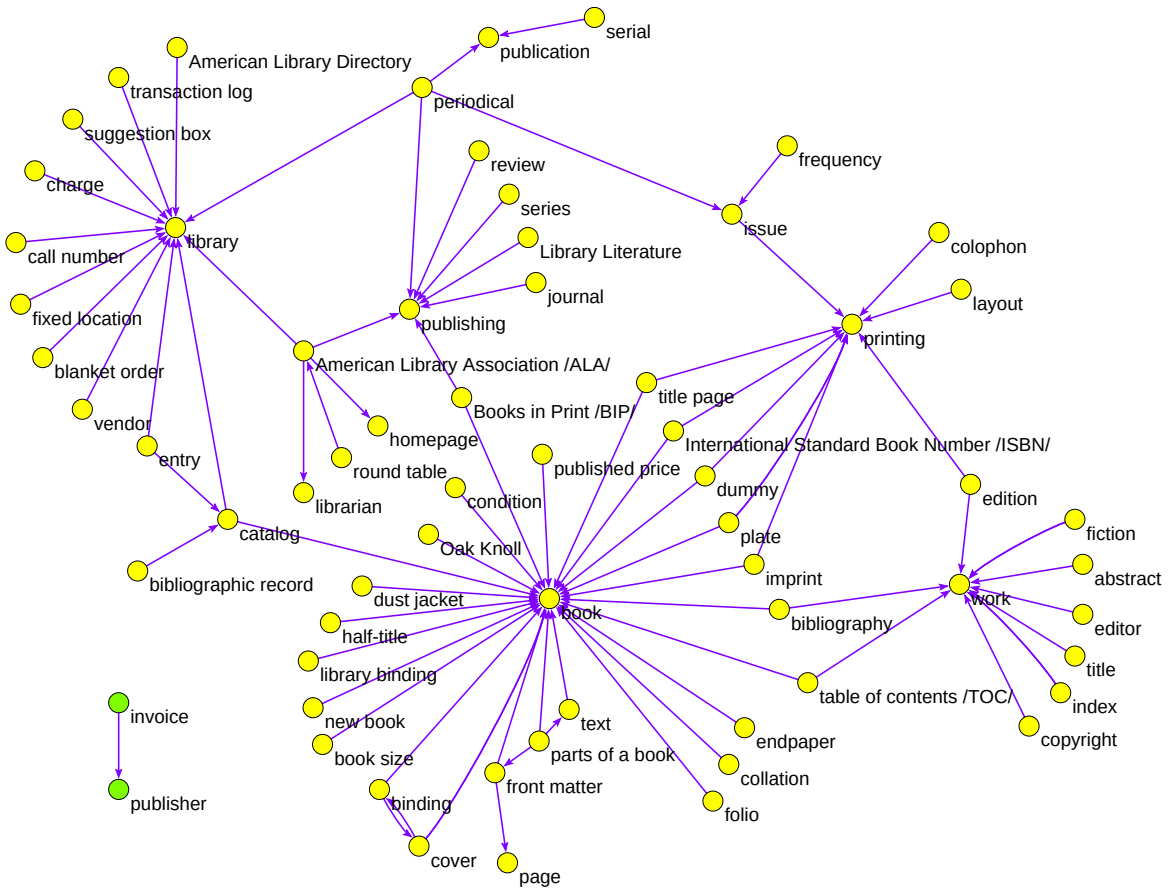
Če pozabimo na vlogo izbrane povezave, imamo le dve vrsti trikotnikov, ki jim povezava lahko pripada: ciklične (**cyc**) in tranzitivne (**tra**, **in**, **out**). V programu `Pajek` ukaz

`Network/Create New Network/with Ring Counts/3-Rings/Directed`

omogoča določiti ustrezna omrežja ($\mathcal{N}_{cyc}$ – ciklične uteži, $\mathcal{N}_{tra}$ – tranzitivnostne uteži, $\mathcal{N}_{sc}$ – tranzitivne bližnjice).

Pojem trikotniške povezanosti lahko posplošimo na *povezanost s kratkimi (pol)cikli – obroči* in ustrezna omrežja; lahko pa tudi na povezanost z majhnimi podgrafi (npr. "klikami", glejte Palla).

Povezavni izrez na ravni 11 tranzitivnega trikotniškega omrežja slovarja ODLIS



Pomembna vozlišča v omrežju

Za določitev pomembnih/zanimivih elementov (vozlišč/povezav) v omrežju ponavadi poskusimo izraziti naša pričakovanja z mero (indeks, utež), ki zadošča zahtevi

> *večja je za dani element vrednost mere, pomembnejši/zanimivejši je ta element*

V analizi omrežij raziskovalci prepogosto nekritično kar uporabijo mere iz literature/programov.

Za formalni pristop glej Roberts.

Pomembna vozlišča v omrežju

Pri izgradnji *mer pomembnosti* moramo najprej upoštevati ali je omrežje usmerjeno ali neusmerjeno. Meram pomembnosti na neusmerjenih omrežjih pravimo mere *središčnosti*; na usmerjenih omrežjih pa mere *veljave*. Slednje se naprej delijo na mere *ugleda* ali *podpore* (upoštevamo vstopajoče povezave) in mere *vpliva* (upoštavamo izstopajoče povezave).

Če zamenjamo dano usmerjeno omrežje z njemu nasprotnim (obrnemo smeri povezav) preidejo mere ugleda v mere vpliva, in obratno.

Dejanski pomen mere pomembnosti je odvisen od relacije (omrežja). Tako npr. je 'najuglednejša' oseba glede na relacijo '__ ne mara sodelovati z __' dejansko najmanj priljubljena oseba.

Odstranitev pomembnega vozlišča iz omrežja povzroči občutno spremembo v zgradbi/delovanju omrežja.

Normalizacija

Naj bo $p : \mathcal{V} \to \mathbb{R}$ neka mera pomembnosti vozlišč omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L})$. Če želimo vrednosti mere p primerjati med različnimi omrežji, moramo poskrbeti za primerljivost. Pogosto jo poskušamo zagotoviti tako, da mero p *normaliziramo*.

Naj bo $\mathcal{N} \in \mathbf{N}(\mathcal{V})$, kjer je $\mathbf{N}(\mathcal{V})$ izbrana množica omrežij nad isto množico $\mathcal{V}$,

$$p_{max} = \max_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \max_{v \in \mathcal{V}} p_{\mathcal{N}}(v) \qquad \text{in} \qquad p_{min} = \min_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} \min_{v \in \mathcal{V}} p_{\mathcal{N}}(v)$$

Tedaj je normalizirana mera enaka

$$p'(v) = \frac{p(v) - p_{min}}{p_{max} - p_{min}} \in [0, 1]$$

Stopnje

Najpreprostejšo mero pomembnosti predstavljajo stopnje vozlišč. Ker sta za enostavna omrežja $\deg_{min} = 0$ in $\deg_{max} = n - 1$, je ustrezna normalizirana mera

središčnost $\quad \deg'(v) = \dfrac{\deg(v)}{n-1}$

in podobno

ugled $\quad \mathrm{indeg}'(v) = \dfrac{\mathrm{indeg}(v)}{n}$

vpliv $\quad \mathrm{outdeg}'(v) = \dfrac{\mathrm{outdeg}(v)}{n}$

Namesto stopenj glede na osnovno omrežje lahko vzamemo tudi stopnje glede na relacijo dosegljivosti (tranzitivna ovojnica).

```
Network/Create Partition/Degree
Network/Create Vector/Centrality/Degree
Network/Create Vector/Centrality/Proximity Prestige
```

Dostopnost

Če upoštevamo razdalje $d(u,v)$ med vozlišči v omrežju $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ lahko vpeljemo

polmer $\quad r(v) = \max_{u \in \mathcal{V}} d(v,u)$

Količino $D = \max_{u,v \in \mathcal{V}} d(v,u)$ imenujemo *premer* omrežja.

skupna dostopnost $\quad S(v) = \sum_{u \in \mathcal{V}} d(v,u)$

Za usmerjeno omrežje sta vpeljani meri meri vpliva. Meri ugleda dobimo, če v obrazcih $d(u,v)$ zamenjamo z $d(v,u)$.

Če omrežje ni krepko povezano, sta r_{max} in S_{max} enaki ∞. Sabidussi (1966) je zato kot mero dostopnosti vpeljal $1/S(v)$ oziroma v normalizirani obliki

dostopnost $\quad cl(v) = \dfrac{n-1}{\sum_{u \in \mathcal{V}} d(v,u)}$

`Network/Create Vector/Centrality/Closeness`

`Network/Create New Network/Subnetwork with Paths/Info on Diameter`

Vmesnost

Pomembna so tudi vozlišča, ki lahko nadzirajo pretok podatkov po omrežju. Če privzamemo, da so za prenos pomembne le najkrajše poti, dobimo kot mero *vmesnosti* (Anthonisse 1971, Freeman 1977)

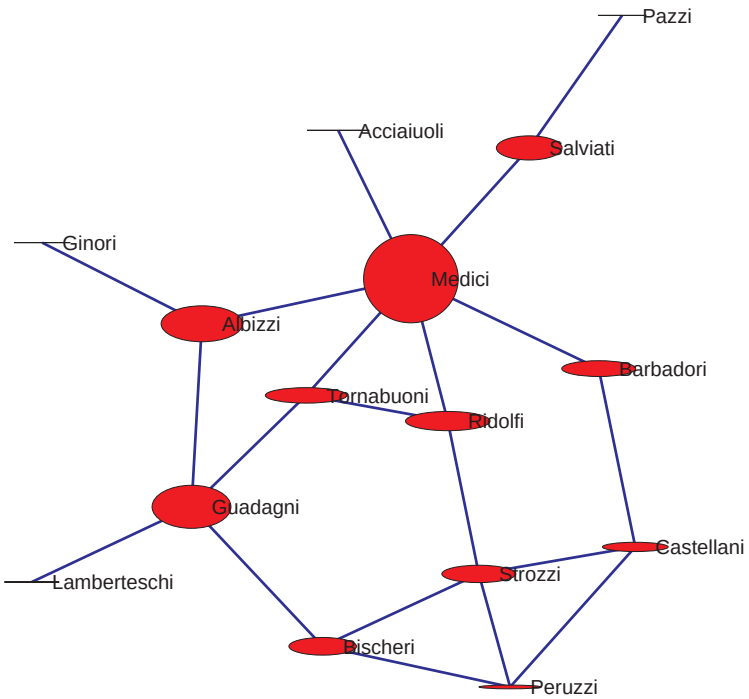
$$b(v) = \frac{1}{(n-1)(n-2)} \sum_{\substack{u,t \in \mathcal{V}: g_{u,t} > 0 \\ u \neq v, t \neq v, u \neq t}} \frac{g_{u,t}(v)}{g_{u,t}}$$

kjer je $g_{u,t}$ število najkrajših poti iz u v t; in $g_{u,t}(v)$ število takih med njimi, ki gredo skozi vozlišče v.

Hiter postopek za izračun vmesnosti je razvil Brandes.

`Network/Create Vector/Centrality/Betweenness`

Padgett-ove floretinske rodbine



		close	between
1.	Acciaiuoli	0.368421	0.000000
2.	Albizzi	0.482759	0.212454
3.	Barbadori	0.437500	0.093407
4.	Bischeri	0.400000	0.104396
5.	Castellani	0.388889	0.054945
6.	Ginori	0.333333	0.000000
7.	Guadagni	0.466667	0.254579
8.	Lamberteschi	0.325581	0.000000
9.	Medici	0.560000	0.521978
10.	Pazzi	0.285714	0.000000
11.	Peruzzi	0.368421	0.021978
12.	Ridolfi	0.500000	0.113553
13.	Salviati	0.388889	0.142857
14.	Strozzi	0.437500	0.102564
15.	Tornabuoni	0.482759	0.091575

Kazala in viri

Vozliščem povezanega usmerjenega omrežja $\mathcal{N} = (\mathcal{V}, \mathcal{L})$ priredimo dve vrednosti: kakovost vira (vsebine) x_v in kakovost kazala y_v (Kleinberg, 1998). Na dober vir kažejo dobra kazala in dobro kazalo kaže na dobre vire

$$x_v = \sum_{u:(u,v)\in\mathcal{L}} y_u \qquad \text{in} \qquad y_v = \sum_{u:(v,u)\in\mathcal{L}} x_u$$

Naj bo $\mathbf{W}$ matrika omrežja $\mathcal{N}$ in $\mathbf{x}$ ter $\mathbf{y}$ vektorja obeh lastnosti. Tedaj lahko zvezi zapišemo $\mathbf{x} = \mathbf{W}^T\mathbf{y}$ oziroma $\mathbf{y} = \mathbf{W}\mathbf{x}$.

Začnimo z $\mathbf{y} = [1, 1, \ldots, 1]$ in nato zaporedoma izračunamo po obeh zvezah nove približke za $\mathbf{x}$ in $\mathbf{y}$. Oba vektorja po vsakem koraku normaliziramo. To ponavljamo dokler se vektorja ne ustalita. Pokazati je mogoče, da opisani postopek konvergira. Limitni vektor $\mathbf{x}^*$ je glavni lastni vektor matrike $\mathbf{W}^T\mathbf{W}$; $\mathbf{y}^*$ pa matrike $\mathbf{W}\mathbf{W}^T$.

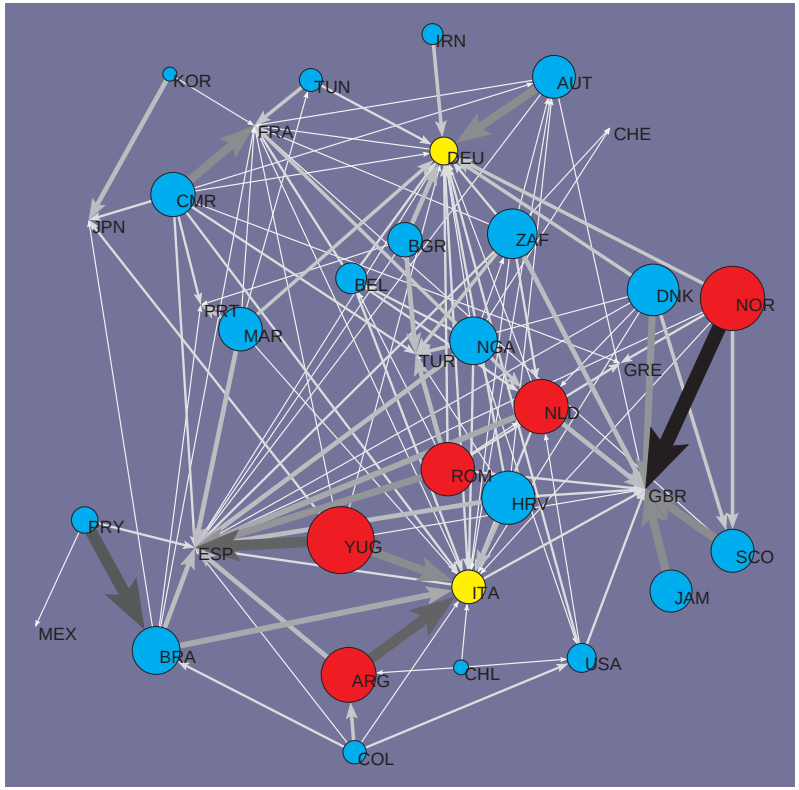
...Kazala in viri

Podobni postopki se uporabljajo v spletnih iskalnikih za ocenjevanje pomembnosti posameznih strani.
PageRank, PageRank / Google, HITS / AltaVista, SALSA, teorija.

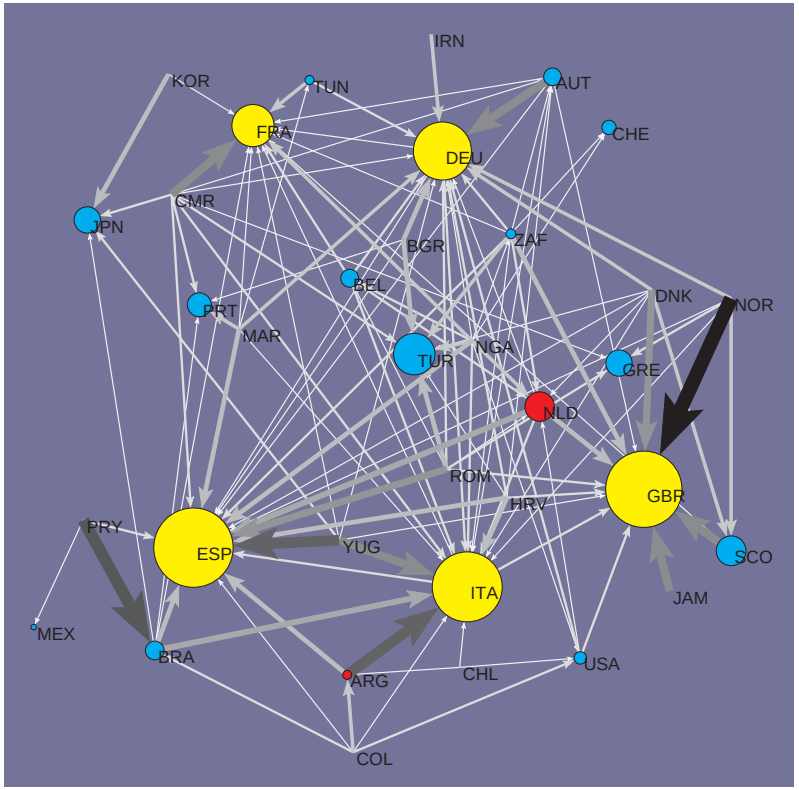
`Network/Create Vector/Centrality/Hubs-Authorities`

Na naslednji prosojnici: Na svetovnem nogometnem prvenstvu v Parizu leta 1998 je sodelovalo 22 nogometnih reprezentanc. V omrežju so vse države, iz katerih so nogometaši igrali v ligah teh 22 držav, in vse države, v katerih ligah so igrali nogometaši iz teh 22 držav. Relacija je *igralec iz države x igra v državi y*; utež je število takih igralcev. Podatke je zbral Lothar Krempel. `football.net`

...Kazala in viri: nogometaši



Izvozniki (kazala/hubs)



Uvozniki (viri/authorities)

Nakopičenost

Nakopičenost v vozlišču v je določena kot razmerje med številom vseh povezav v podgrafu $G^1(v)$ porojenim s soseščino danega vozlišča in številom povezav v polnem grafu na teh vozliščih

$$C(v) = \frac{2|\mathcal{L}(G^1(v))|}{\deg(v)(\deg(v)-1)}$$

za $\deg(v) > 1$; in $C(v) = 0$ sicer.

Vpliv velikosti soseščine lahko zagotovimo z naslednjim popravkom

$$C_1(v) = \frac{\deg(v)}{\Delta} C(v)$$

kjer je Δ največja stopnja v grafu G. Ta doseže največjo možno vrednost le na vozliščih, ki pripadajo osamljeni kliki reda Δ.

`Network/Create Vector/Clustering Coefficients/CC2`

članek

Izračun indeksov in uteži v Pajek-u

Xingqin Qi et al. so v članku Terrorist Networks, Network Energy and Node Removal vpeljali mero *Laplaceova središčnost*

$$L(v) = \deg(v)(\deg(v)+1) + 2\sum_{u \in N(v)} \deg(u)$$

```
select the network
Network/Create Vector/Centrality/Degree/All
Operations/Network+Vector/Neighbours/Sum/All [False]
Vector/Transform/Multiply by [2]
select the degree vector as First
select the degree vector as Second
Vectors/Multiply (First*Second)
Vectors/Add (First+Second)
select the 2*sum on neighbors as Second
Vectors/Add (First+Second)
dispose auxiliary vectors
File/Vector/Change Label [Laplace All centrality]
```

Za določitev pomembnih podomrežij v cestnem omrežju lahko uporabimo utež $w(u,v) = \deg(u)\deg(v)$.

```
Network/Create vector/Centrality/Degree/All
Operations/Network+Vector/Transform/Vector(s)->Line Values/Multiply
```

Usredinjenost omrežja

Mero pomembnosti $p : \mathcal{V} \to \mathbb{R}$ lahko povzamemo na celotnem omrežju kot njegovo usredinjenost $C(p)$:

$$p^* = \max_{v \in \mathcal{V}} p(v)$$

$$D(p) = \sum_{v \in \mathcal{V}} (p^* - p(v))$$

$$D^* = \max_{\mathcal{N} \in \mathbf{N}(\mathcal{V})} D(p_{\mathcal{N}})$$

Tedaj je *usredinjenost* glede na p

$$C(p) = \frac{D(p)}{D^*}$$

Za večino mer je najbolj usredinjena zvezda S_n in najmanj polni graf K_n.

Analiza omrežij

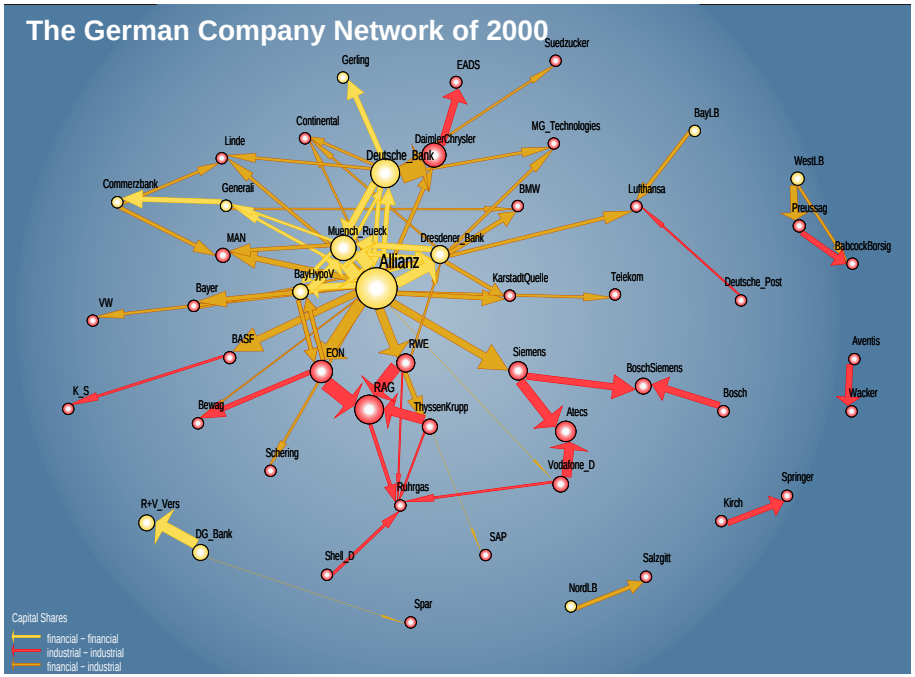
6. Zgradba omrežij otoki in gruče

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, maj 2024

Kazalo

1 Otoki
2 Gruče



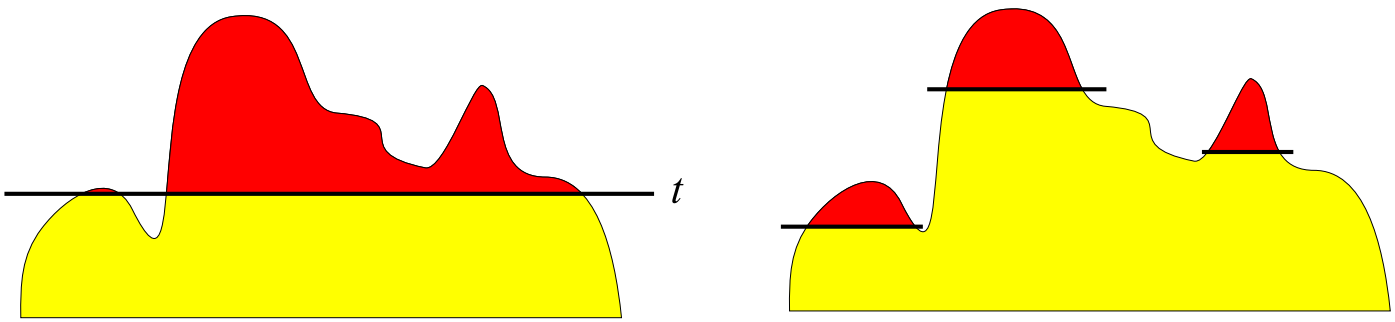
L. Krempl, MPI.

prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

30. april 2024 ob 03 : 05/ april 2013

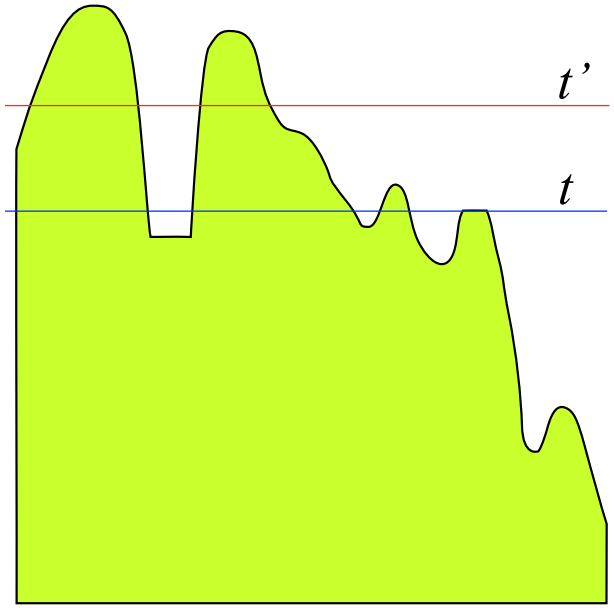
Otoki

Če dano lastnost/utež vozlišč/povezav predstavimo kot njihovo višino, nam naše omrežje določa nekakšno pokrajino s hribi in dolinami. Če to pokrajino potopimo v vodo do izbrane višine, dobimo kot izrez *otoke*. S spreminjanjem višine vode dobivamo različne otoke. V uporabah nas običajno zanimajo ne preveliki in ne premajhni otoki – le otoki velikosti med izbranima k in K. Postopek 'otoki' začne s pokrajino popolnoma potopljeno v vodo. Nato znižujemo višino, dokler se ne pojavi otok prave velikosti ...



Otoki so zelo splošen in učinkovit postopek za določanje 'pomembnih' podomrežij v danem omrežju.

... otoki



V omrežju $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \to \mathbb{R}$ imenujemo *pravi vozliščni otok* skupino vozlišč $\mathcal{C} \subseteq \mathcal{V}$, ki poraja povezan podgraf in so vozlišča v skupini 'višje' od sosednjih vozlišč

$$\max_{u \in N(\mathcal{C})} p(u) < \min_{v \in \mathcal{C}} p(v)$$

V omrežju $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \to \mathbb{R}$ imenujemo *pravi povezavni otok* skupino vozlišč $\mathcal{C} \subseteq \mathcal{V}$, ki poraja povezan podgraf in so vozlišča v skupini 'močneje' povezana, kot s sosedi – v podgrafu porojenem s $\mathcal{C}$ obstaja vpeto drevo $\mathcal{T}$, tako da je

$$\max_{(u,v) \in \mathcal{L}, u \notin \mathcal{C}, v \in \mathcal{C}} w(u, v) < \min_{(u,v) \in \mathcal{T}} w(u, v)$$

Vpeljemo lahko tudi pojem *enostavnih* otokov z enim samim 'vrhom'.

Lastnosti vozliščnih otokov

- Množica vozlišč posamezne komponente vozliščnega izreza na izbrani ravni t določa pravi vozliščni otok.
- Množica $\mathcal{H}_p(\mathcal{N})$ vseh vozliščnih otokov omrežja $\mathcal{N}$ je polna razslojitev:
  - otoka sta ali ločena ali pa je en vsebovan v drugem;
  - vsako vozlišče pripada vsaj enemu otoku.
- Vozliščni otoki so invariantni za strogo naraščajoče transformacije lastnosti p.
- Krajišči iste povezave ne moreta pripadati dvem različnima otokoma.

Postopek za določanje vozliščnih otokov

- Omrežje potopimo v celoti v vodo in jo postopno spuščamo.
- Vsakič, ko se prikaže novo vozlišče v iz vode, pogledamo, s katerimi od že vidnih otokov je povezano.
- Vozlišče v in z njim povezane otoke združimo v nov, večji otok. Otoki, ki so pri tem združeni, so njegovi podotoki. Vozlišče v je pristanišče novega otoka (vozlišče z najmanjšo vrednostjo).
- To je mogoče izvesti v času $\mathcal{O}(\max(n \log n, m))$.

`Operations/Network+Vector/Islands/Vertex Property`

Enostavni vozliščni otoki

- Množica vozlišč $\mathcal{C} \subseteq \mathcal{V}$ je (lokalni) *vozliščni vrh*, če je pravi vozliščni otok in imajo vsa njegova vozlišča isto vrednost.
- Vozliščni otok je *enostaven*, če ima en sam vrh.
- Obstajajo tri vrste vozliščnih otokov:
  - PLOSKI – vsa njegova vozlišča imajo isto vrednost;
  - ENOVRŠNI – otok ima en sam vrh;
  - VEČVRŠNI – otok ima več vrhov.
- PLOSKI ali ENOVRŠNI otoki so enostavni.

Lastnosti povezavnih otokov

- Množice vozlišč povezanih komponent povezavnega izreza na izbrani ravni t so pravi povezavni otoki.
- Množica $\mathcal{H}_w(\mathcal{N})$ vseh pravih povezavnih otokov omrežja $\mathcal{N}$ je razslojitev (ne nujno polna):
  - dva otoka sta ali ločena ali pa je en vsebovan v drugem.
- Povezavni otoki so invariantni za strogo naraščajoče transformacije uteži w.
- Krajišči povezave lahko pripadata dvema ločenima pravima povezavnima otokoma.

Algoritem določitve pravih povezavnih otokov

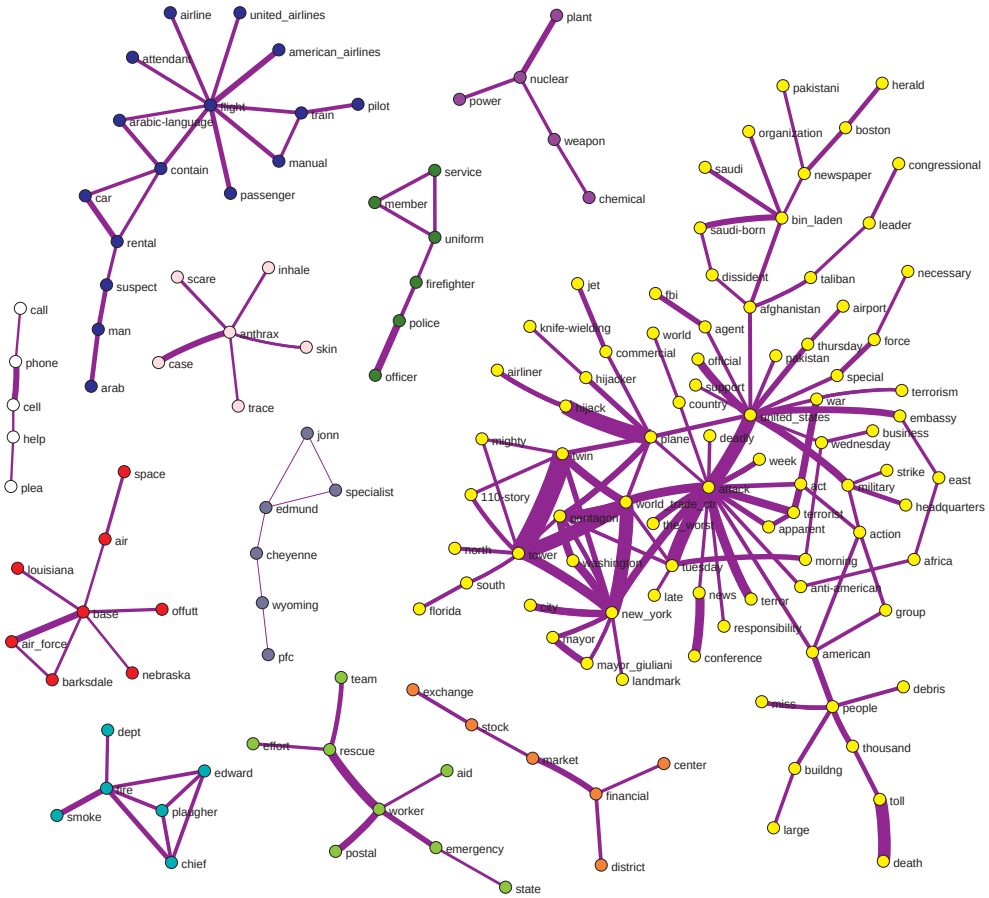
- Omrežje potopimo v celoti v vodo in jo postopno spuščamo.
- Vsakič, ko se prikaže nova povezava e iz vode, pogledamo, s katerimi od že vidnih otokov je povezana (obstajata največ dva).
- Združimo ta otoka s povezavo ali dodamo povezavo otoku. Dobimo večji otok. Združeni otoki so podotoki tega otoka. Povezava e je *pristanišče* novega otoka (najmanjša vrednost v minimalnem vpetem drevesu).
- Prave povezavne otoke je mogoče določiti v času $\mathcal{O}(m \log n)$.

`Network/Create Partition/Islands/Line Weights`

Enostavni povezavni otoki

- Množica vozlišč $\mathcal{C} \subseteq \mathcal{V}$ določa *povezavni vrh*, če je pravi povezavni otok, ki vsebuje minimalno vpeto drevo, v katerem imajo vse povezave enako vrednost, kot je največja vrednost povezave v otoku.
- Povezavni otok je *enostaven*, če ima en sam vrh.
- Obstajajo tri vrste povezavnih otokov:
  - PLOSKI – minimalno vpeto drevo, v katerem imajo vse povezave enako vrednost, kot je največja vrednost povezave v otoku;
  - ENOVRŠNI – otok ima en sam povezavni vrh;
  - VEČVRŠNI – otok ima več povezavnih vrhov.
- PLOSKI ali ENOVRŠNI otoki so enostavni.

Otoki - Reuters terror news



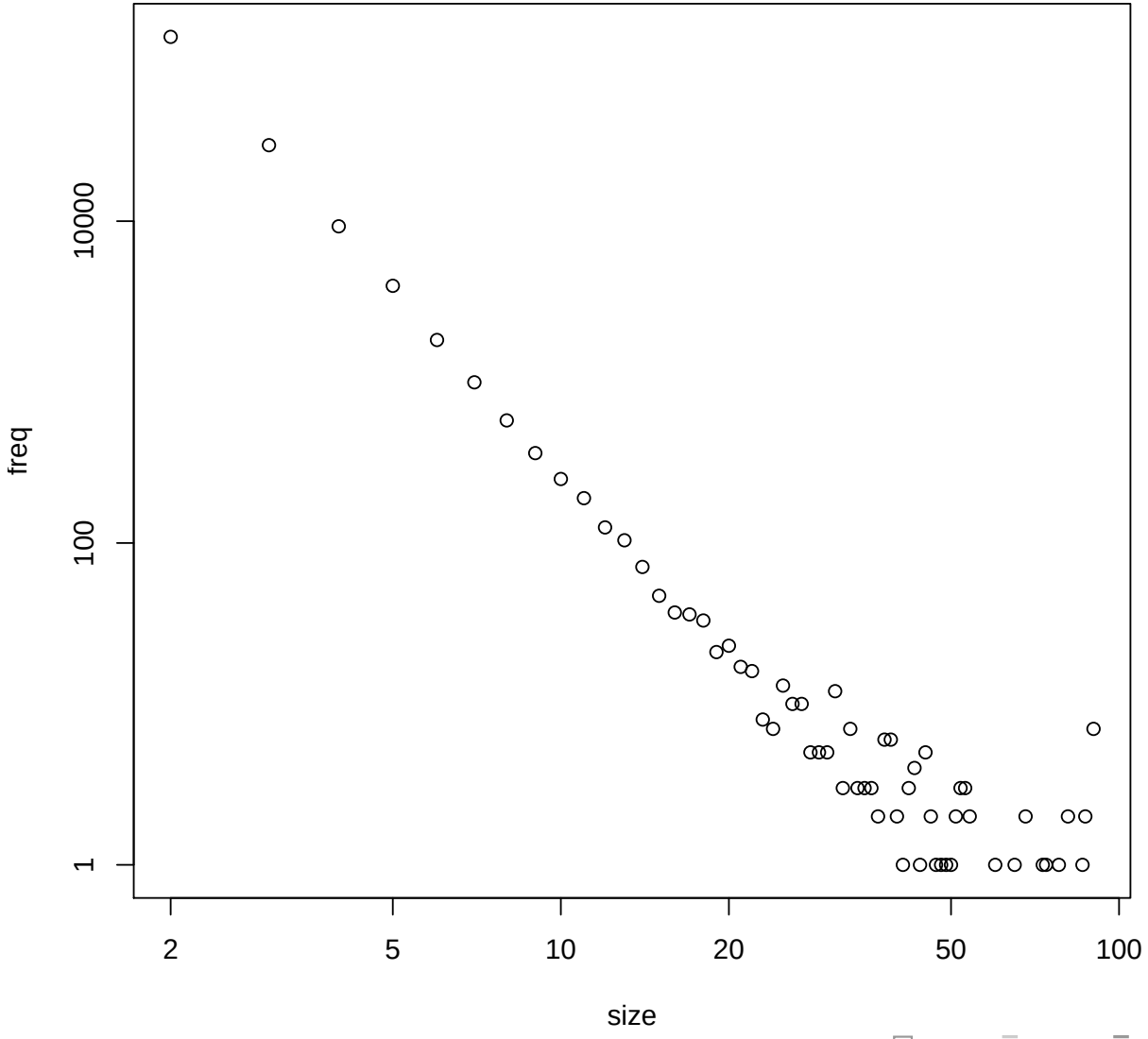
S postopkom CRA sta S. Corman in K. Dooley ustvarila omrežje *Reuters terror news*, ki povzema vse novice, ki jih je izdalo novičarsko podjetje Reuters v 66 zaporednih dnevih po 11. septembru in so se nanašali na ta dogodek. Vozlišča omrežja so besede (izrazi); besedi sta povezani, če se sopojavljata v enoti besedila. Utež povezave je pogostost sopojavljanja. Omrežje ima $n = 13332$ vozlišč in $m = 243447$ povezav.

Primer: Ameriški patenti

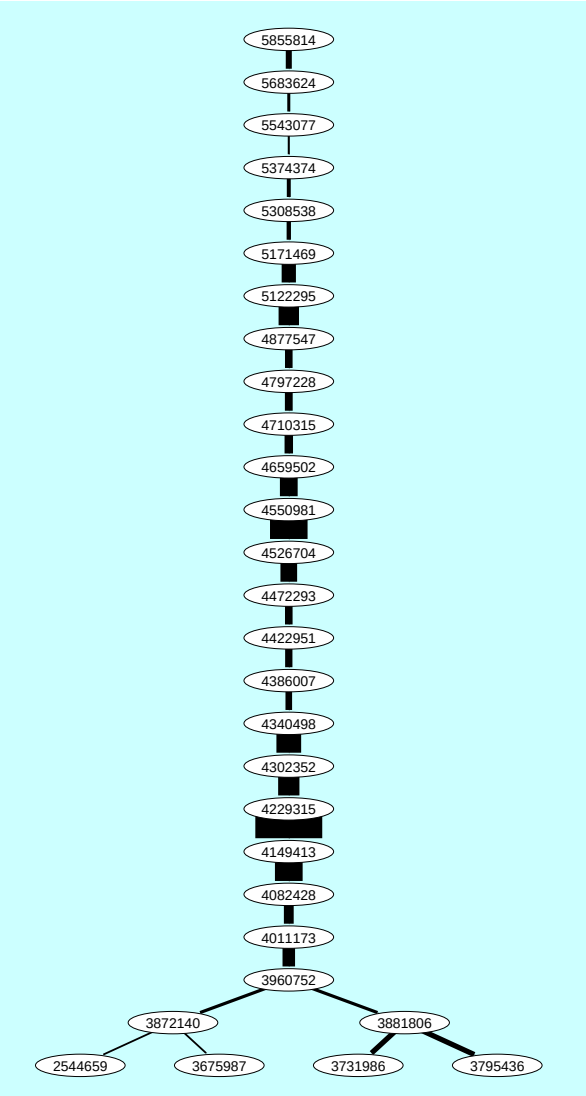
Omrežje ameriških patentov (Nber, US Patents) ima 3774768 vozlišč in 16522438 usmerjenih povezav (1 zanka). Če zanko odstranimo, je omrežje aciklično. Za uteži povezav smo vzeli delež različnih poti od začetkov do koncev, ki vodijo skozi povezavo in določili vse (2,90)-otoke. Z otoki porojeno omrežje ima 470137 vozlišč, 307472 povezav in za različne k: C_2 =187610, C_5 =8859, C_{30} =101, C_{50} =30 otokov. Rolex

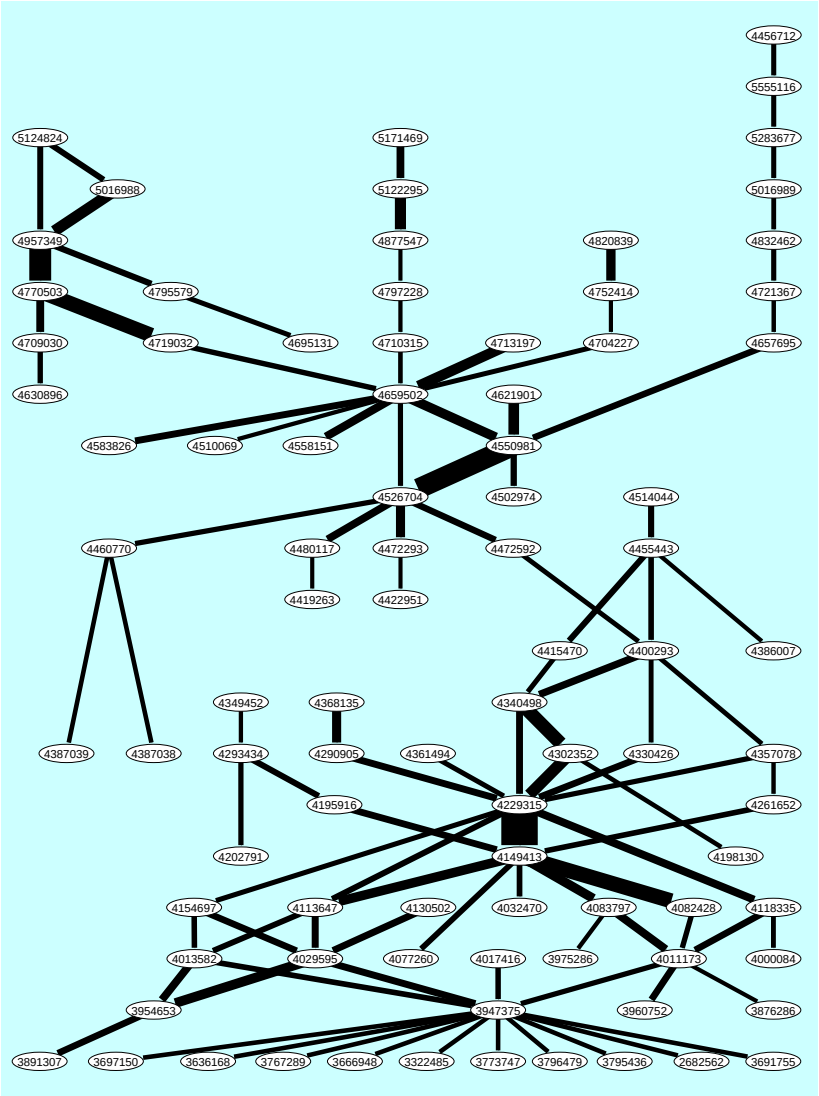
[1]	0	139793	29670	9288	3966	1827	997	578	362	250
[11]	190	125	104	71	47	37	36	33	21	23
[21]	17	16	8	7	13	10	10	5	5	5
[31]	12	3	7	3	3	3	2	6	6	2
[41]	1	3	4	1	5	2	1	1	1	1
[51]	2	3	3	2	0	0	0	0	0	1
[61]	0	0	0	0	1	0	0	2	0	0
[71]	0	0	1	1	0	0	0	1	0	0
[81]	2	0	0	0	0	1	2	0	0	7

Porazdelitev velikosti otokov



Glavna pot in glavni otok v US Patents





Glavni otok – Liquid crystal display

Table 1: Patents on the liquid-crystal display

patent	date	author(s) and title
2544659	Mar 13, 1951	Dreyer. Dichroic light-polarizing sheet and the like and the formation and use thereof
2682562	Jun 29, 1954	Wender, et al. Reduction of aromatic carbinols
3322485	May 30, 1967	Williams. Electro-optical elements utilazing an organic nematic compound
3636168	Jan 18, 1972	Josephson. Preparation of polynuclear aromatic compounds
3666948	May 30, 1972	Mechlowitz, et al. Liquid crystal termal imaging system having an undisturbed image on a disturbed background
3675987	Jul 11, 1972	Rafuse. Liquid crystal compositions and devices
3691755	Sep 19, 1972	Girard. Clock with digital display
3697150	Oct 10, 1972	Wysochi. Electro-optic systems in which an electrophoretic-like or dipolar material is dispersed throughout a liquid crystal to reduce the turn-off time
3731986	May 8, 1973	Fergason. Display devices utilizing liquid crystal light modulation
3767289	Oct 23, 1973	Aviram, et al. Class of stable trans-stilbene compounds, some displaying nematic mesophases at or near room temperature and others in a range up to 100°C
3773747	Nov 20, 1973	Steinstrasser. Substituted azoxy benzene compounds
3795436	Mar 5, 1974	Boller, et al. Nematogenic material which exhibit the Kerr effect at isotropic temperatures
3796479	Mar 12, 1974	Helfrich, et al. Electro-optical light-modulation cell utilizing a nematogenic material which exhibits the Kerr effect at isotropic temperatures
3872140	Mar 18, 1975	Klanderman, et al. Liquid crystalline compositions and method
3876286	Apr 8, 1975	Deutscher, et al. Use of nematic liquid crystalline substances
3881806	May 6, 1975	Suzuki. Electro-optical display device
3891307	Jun 24, 1975	Tsukamoto, et al. Phase control of the voltages applied to opposite electrodes for a cholesteric to nematic phase transition display
3947375	Mar 30, 1976	Gray, et al. Liquid crystal materials and devices
3954653	May 4, 1976	Yamazaki. Liquid crystal composition having high dielectric anisotropy and display device incorporating same
3960752	Jun 1, 1976	Klanderman, et al. Liquid crystal compositions
3975286	Aug 17, 1976	Oh. Low voltage actuated field effect liquid crystals compositions and method of synthesis
4000084	Dec 28, 1976	Hsieh, et al. Liquid crystal mixtures for electro-optical display devices
4011173	Mar 8, 1977	Steinstrasser. Modified nematic mixtures with positive dielectric anisotropy
4013582	Mar 22, 1977	Gavrilovic. Liquid crystal compounds and electro-optic devices incorporating them
4017416	Apr 12, 1977	Inukai, et al. P-cyanophenyl 4-alkyl-4'-biphenylcarboxylate, method for preparing same and liquid crystal compositions using same
4029595	Jun 14, 1977	Ross, et al. Novel liquid crystal compounds and electro-optic devices incorporating them
4032470	Jun 28, 1977	Bloom, et al. Electro-optic device
4077260	Mar 7, 1978	Gray, et al. Optically active cyano-biphenyl compounds and liquid crystal materials containing them
4082428	Apr 4, 1978	Hsu. Liquid crystal composition and method

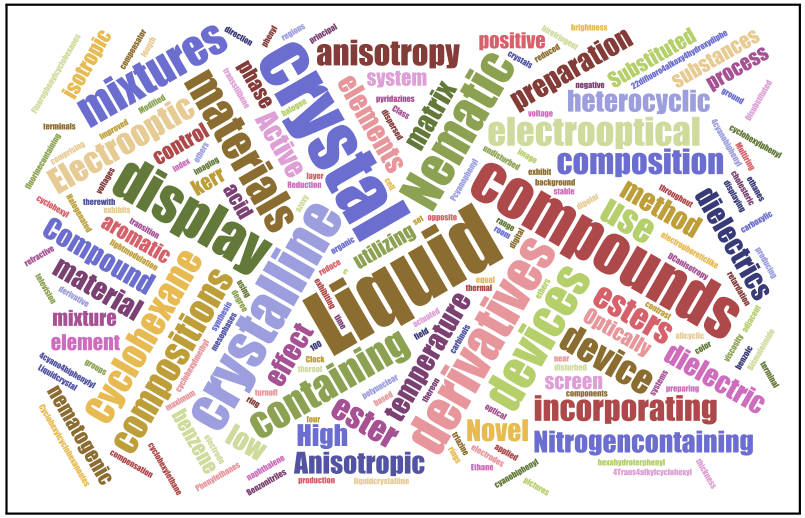
Table 2: Patents on the liquid-crystal display

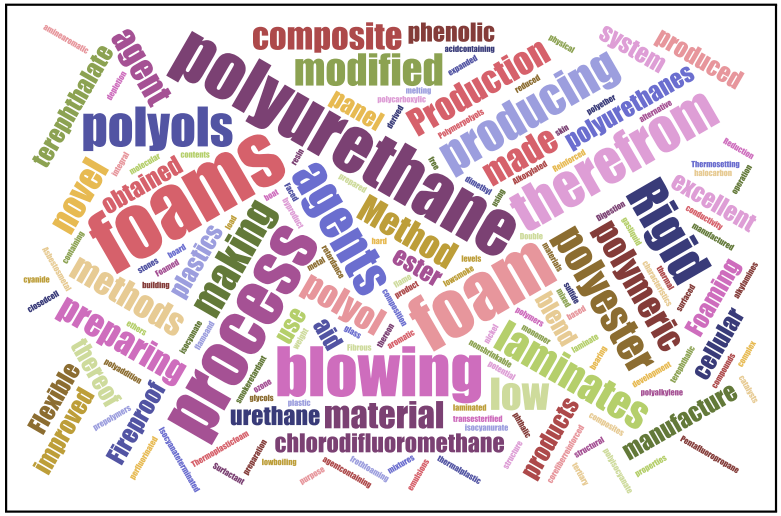
patent	date	author(s) and title
4083797	Apr 11, 1978	Oh. Nematic liquid crystal compositions
4113647	Sep 12, 1978	Coates, et al. Liquid crystalline materials
4118335	Oct 3, 1978	Krause, et al. Liquid crystalline materials of reduced viscosity
4130502	Dec 19, 1978	Eidenschink, et al. Liquid crystalline cyclohexane derivatives
4149413	Apr 17, 1979	Gray, et al. Optically active liquid crystal mixtures and liquid crystal devices containing them
4154697	May 15, 1979	Eidenschink, et al. Liquid crystalline hexahydroterphenyl derivatives
4195916	Apr 1, 1980	Coates, et al. Liquid crystal compounds
4198130	Apr 15, 1980	Boller, et al. Liquid crystal mixtures
4202791	May 13, 1980	Sato, et al. Nematic liquid crystalline materials
4229315	Oct 21, 1980	Krause, et al. Liquid crystalline cyclohexane derivatives
4261652	Apr 14, 1981	Gray, et al. Liquid crystal compounds and materials and devices containing them
4290905	Sep 22, 1981	Kanbe. Ester compound
4293434	Oct 6, 1981	Deutscher, et al. Liquid crystal compounds
4302352	Nov 24, 1981	Eidenschink, et al. Fluorophenylcyclohexanes, the preparation thereof and their use as components of liquid crystal dielectrics
4330426	May 18, 1982	Eidenschink, et al. Cyclohexylbiphenyls, their preparation and use in dielectrics and electrooptical display elements
4340498	Jul 20, 1982	Sugimori. Halogenated ester derivatives
4349452	Sep 14, 1982	Osman, et al. Cyclohexylcyclohexanoates
4357078	Nov 2, 1982	Carr, et al. Liquid crystal compounds containing an alicyclic ring and exhibiting a low dielectric anisotropy and liquid crystal materials and devices incorporating such compounds
4361494	Nov 30, 1982	Osman, et al. Anisotropic cyclohexyl cyclohexylmethyl ethers
4368135	Jan 11, 1983	Osman. Anisotropic compounds with negative or positive DC-anisotropy and low optical anisotropy
4386007	May 31, 1983	Krause, et al. Liquid crystalline naphthalene derivatives
4387038	Jun 7, 1983	Fukui, et al. 4-(Trans-4'-alkylcyclohexyl) benzoic acid 4'''-cyano-4''-biphenylyl esters
4387039	Jun 7, 1983	Sugimori, et al. Trans-4-(trans-4'-alkylcyclohexyl)-cyclohexane carboxylic acid 4'''-cyanobiphenyl ester
4400293	Aug 23, 1983	Romer, et al. Liquid crystalline cyclohexylphenyl derivatives
4415470	Nov 15, 1983	Eidenschink, et al. Liquid crystalline fluorine-containing cyclohexylbiphenyls and dielectrics and electro-optical display elements based thereon
4419263	Dec 6, 1983	Praefcke, et al. Liquid crystalline cyclohexylcarbonitrile derivatives
4422951	Dec 27, 1983	Sugimori, et al. Liquid crystal benzene derivatives
4455443	Jun 19, 1984	Takatsu, et al. Nematic halogen Compound
4456712	Jun 26, 1984	Christie, et al. Bismaleimide triazine composition
4460770	Jul 17, 1984	Petrzilka, et al. Liquid crystal mixture
4472293	Sep 18, 1984	Sugimori, et al. High temperature liquid crystal substances of four rings and liquid crystal compositions containing the same
4472592	Sep 18, 1984	Takatsu, et al. Nematic liquid crystalline compounds
4480117	Oct 30, 1984	Takatsu, et al. Nematic liquid crystalline compounds
4502974	Mar 5, 1985	Sugimori, et al. High temperature liquid-crystalline ester compounds
4510069	Apr 9, 1985	Eidenschink, et al. Cyclohexane derivatives

Table 3: Patents on the liquid-crystal display

patent	date	author(s) and title
4514044	Apr 30, 1985	Gunjima, et al. 1-(Trans-4-alkylcyclohexyl)-2-(trans-4'-(p-substituted phenyl) cyclohexyl)ethane and liquid crystal mixture
4526704	Jul 2, 1985	Petrzilka, et al. Multiring liquid crystal esters
4550981	Nov 5, 1985	Petrzilka, et al. Liquid crystalline esters and mixtures
4558151	Dec 10, 1985	Takatsu, et al. Nematic liquid crystalline compounds
4583826	Apr 22, 1986	Petrzilka, et al. Phenylethanes
4621901	Nov 11, 1986	Petrzilka, et al. Novel liquid crystal mixtures
4630896	Dec 23, 1986	Petrzilka, et al. Benzonitriles
4657695	Apr 14, 1987	Saito, et al. Substituted pyridazines
4659502	Apr 21, 1987	Fearon, et al. Ethane derivatives
4695131	Sep 22, 1987	Balkwill, et al. Disubstituted ethanes and their use in liquid crystal materials and devices
4704227	Nov 3, 1987	Krause, et al. Liquid crystal compounds
4709030	Nov 24, 1987	Petrzilka, et al. Novel liquid crystal mixtures
4710315	Dec 1, 1987	Schad, et al. Anisotropic compounds and liquid crystal mixtures therewith
4713197	Dec 15, 1987	Eidenschink, et al. Nitrogen-containing heterocyclic compounds
4719032	Jan 12, 1988	Wachtler, et al. Cyclohexane derivatives
4721367	Jan 26, 1988	Yoshinaga, et al. Liquid crystal device
4752414	Jun 21, 1988	Eidenschink, et al. Nitrogen-containing heterocyclic compounds
4770503	Sep 13, 1988	Buchecker, et al. Liquid crystalline compounds
4795579	Jan 3, 1989	Vauchier, et al. 2,2'-difluoro-4-alkoxy-4'-hydroxydiphenyls and their derivatives, their production process and their use in liquid crystal display devices
4797228	Jan 10, 1989	Goto, et al. Cyclohexane derivative and liquid crystal composition containing same
4820839	Apr 11, 1989	Krause, et al. Nitrogen-containing heterocyclic esters
4832462	May 23, 1989	Clark, et al. Liquid crystal devices
4877547	Oct 31, 1989	Weber, et al. Liquid crystal display element
4957349	Sep 18, 1990	Clerc, et al. Active matrix screen for the color display of television pictures, control system and process for producing said screen
5016988	May 21, 1991	Iimura. Liquid crystal display device with a birefringent compensator
5016989	May 21, 1991	Okada. Liquid crystal element with improved contrast and brightness
5122295	Jun 16, 1992	Weber, et al. Matrix liquid crystal display
5124824	Jun 23, 1992	Kozaki, et al. Liquid crystal display device comprising a retardation compensation layer having a maximum principal refractive index in the thickness direction
5171469	Dec 15, 1992	Hittich, et al. Liquid-crystal matrix display
5283677	Feb 1, 1994	Sagawa, et al. Liquid crystal display with ground regions between terminal groups
5308538	May 3, 1994	Weber, et al. Supertwist liquid-crystal display
5374374	Dec 20, 1994	Weber, et al. Supertwist liquid-crystal display
5543077	Aug 6, 1996	Rieger, et al. Nematic liquid-crystal composition
5555116	Sep 10, 1996	Ishikawa, et al. Liquid crystal display having adjacent electrode terminals set equal in length
5683624	Nov 4, 1997	Sekiguchi, et al. Liquid crystal composition
5855814	Jan 5, 1999	Matsui, et al. Liquid crystal compositions and liquid crystal display elements

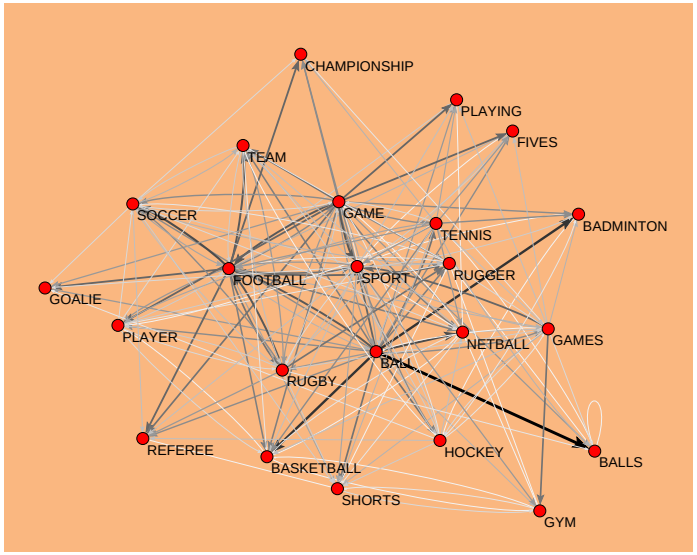
Besedna oblaka za otok "LCD" in otok "foam"

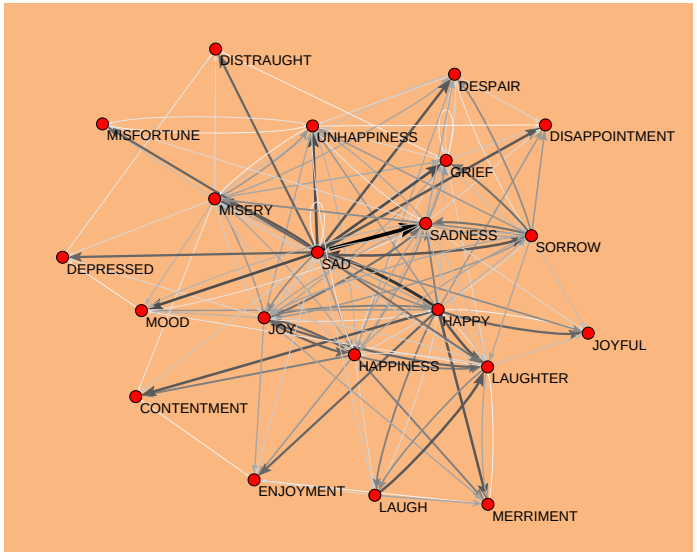


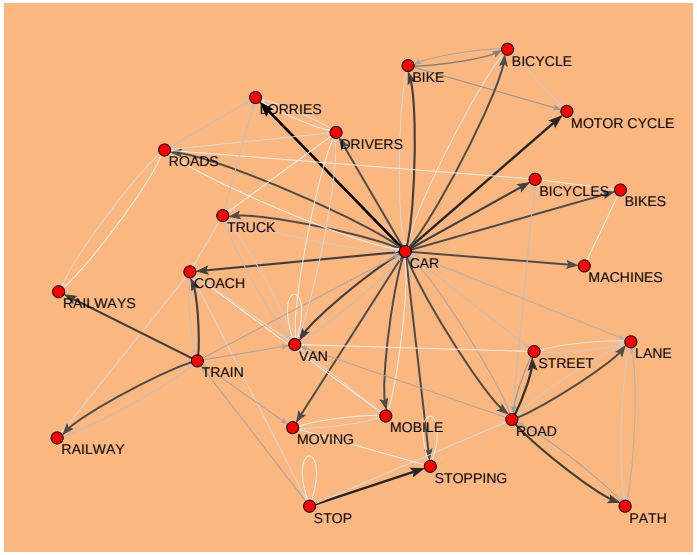


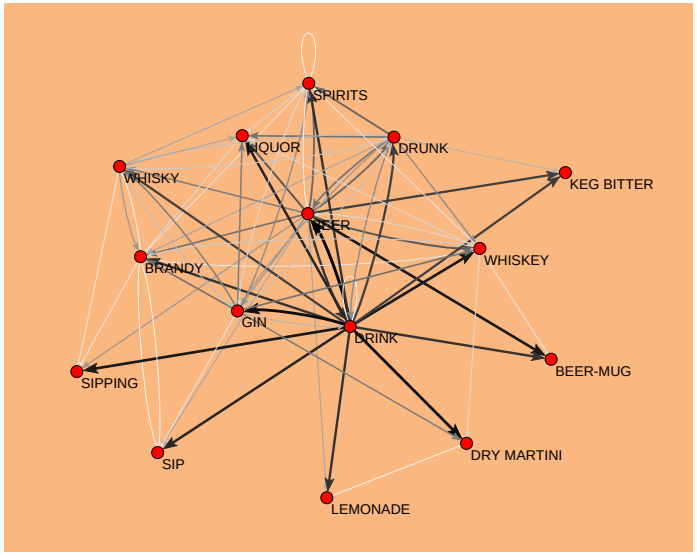
Otoki – The Edinburgh Associative Thesaurus

$n = 23219$, $m = 325624$, tranzitivnostna utež

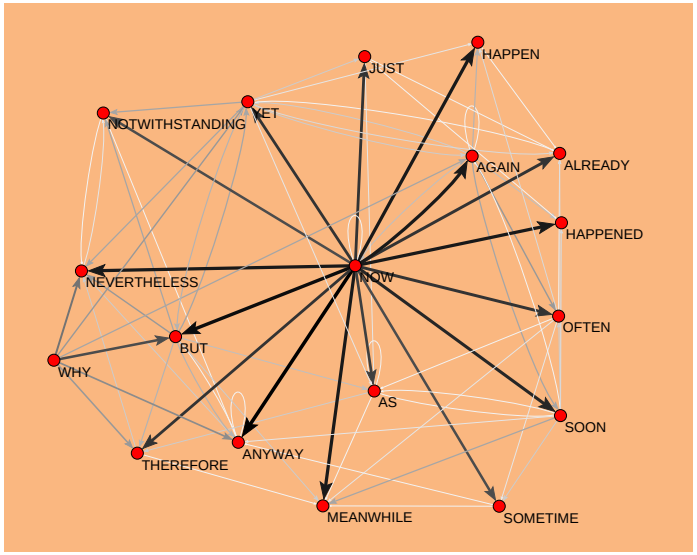


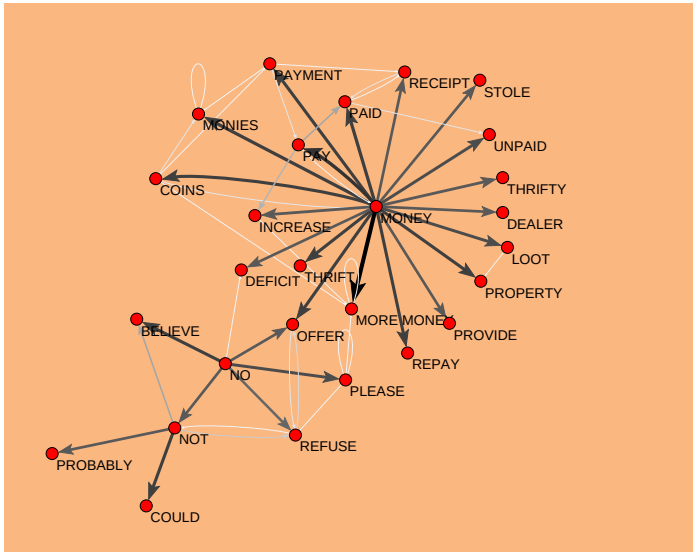


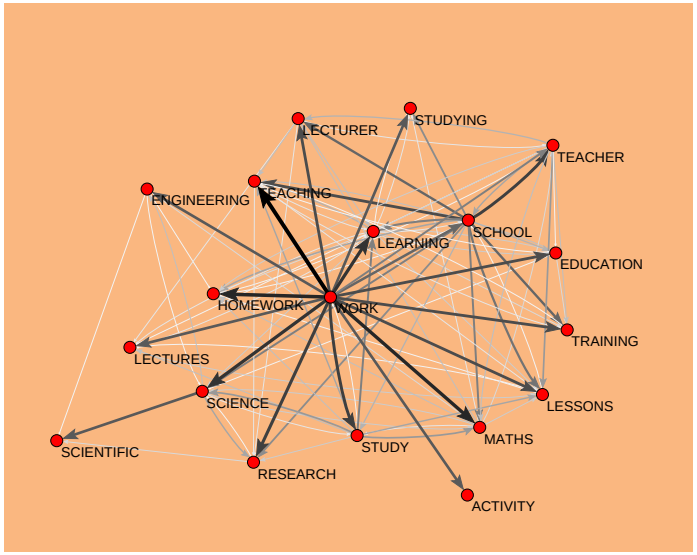


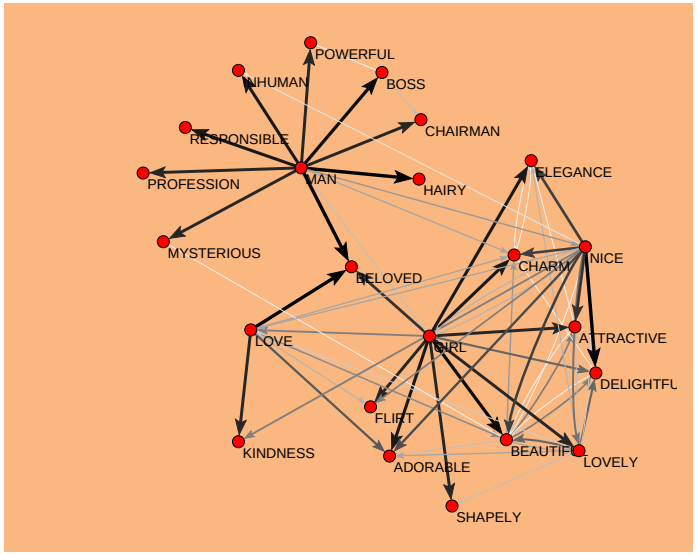


...Otoki – The Edinburgh Associative Thesaurus









Gruče

Predlaganih je bilo več pojmov, ki opisujejo zgoščene skupine ali gruče v omrežjih.

Klika reda k v grafu je maksimalni polni (izomorfen K_k) podgraf, $k \geq 3$.

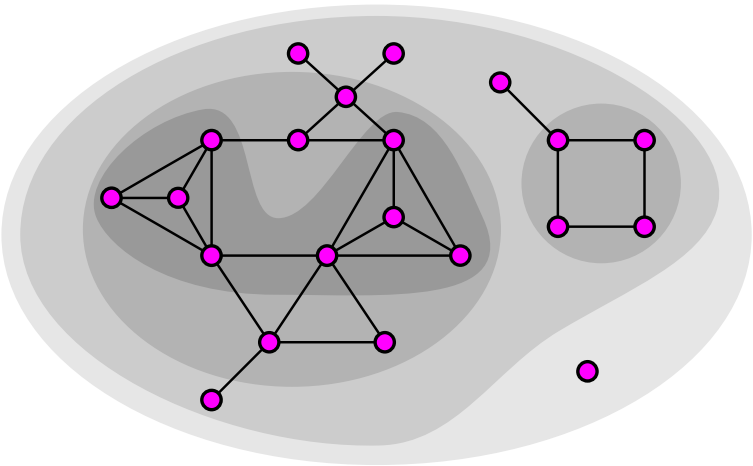
s-klika je maksimalni podgraf $\mathcal{H}$, v katerem najkrajša pot med poljubnima njenima vozliščema ne presega s.
$\forall u, v \in \mathcal{V}(\mathcal{H}) : d(u, v) \leq s$.

s-klan je s-klika s premerom s. $\forall u, v \in \mathcal{V}(\mathcal{H}) : d_{\mathcal{H}}(u, v) \leq s$.

s-pleksi, množice LS, množice lambda, sredice, ...

Za vse, razen za sredice, se je pokazalo, da jih je težko določiti.

Sredice in posplošene sredice



Pojem sredice je vpeljal Seidman leta 1983. Podgraf $\mathcal{H} = (\mathcal{W}, \mathcal{L}(\mathcal{W}))$ grafa $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ porojen z množico $\mathcal{W}$ je k-*sredica* ali *sredica reda* k ntk. $\forall v \in \mathcal{W}$: $\deg_{\mathcal{H}}(v) \geq k$ in je $\mathcal{H}$ maksimalni podgraf s to lastnostjo.
Sredici največjega reda v grafu pravimo tudi *glavna* sredica.

Srednčnost vozlišča v je enaka največjemu redu sredice, ki še vsebuje to vozlišče.

Stopnja $\deg(v)$ v definiciji sredice je lahko: vhodna, izhodna, celotna, itd., kar da različne vrste sredic.

Lastnosti sredic

S slike na prejšnji prosojnici, ki prikazuje sredice redov 0, 1, 2 in 3, vidimo:

- Sredice so gnezdene – določajo razslojitev:
  $i < j \quad \Longrightarrow \quad \mathcal{H}_j \subseteq \mathcal{H}_i$
- Sredice danega reda niso vselej povezani podgrafi.

Učinkovit postopek za določitev sredic temelji na lastnosti:

> *Če iz danega grafa $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ zaporedoma odstranjujemo vsa vozlišča (in pripadajoče povezave), ki imajo stopnjo manjšo od k, bo podgraf, ki bo ostal, sredica reda k.*

Postopek določanja sredic

Input: Graf $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ predstavljen s seznami sosedov
Output: Tabela $core[\mathcal{V}]$ sredičnosti vozlišč

Določi stopnje vozlišč *degree*
Uredi množico vozlišč $\mathcal{V}$ v naraščajočem vrstnem redu njihovih stopenj
for $v \in \mathcal{V}$ glede na tekočo urejenost **do**
    $core[v] = degree[v]$
    for $u \in N(v)$ **do**
    if $degree[u] > degree[v]$ **then**
        $degree[u] = degree[u] - 1$
        $\mathcal{V} = \mathcal{V} \setminus \{v\}$; Preuredi $\mathcal{V}$

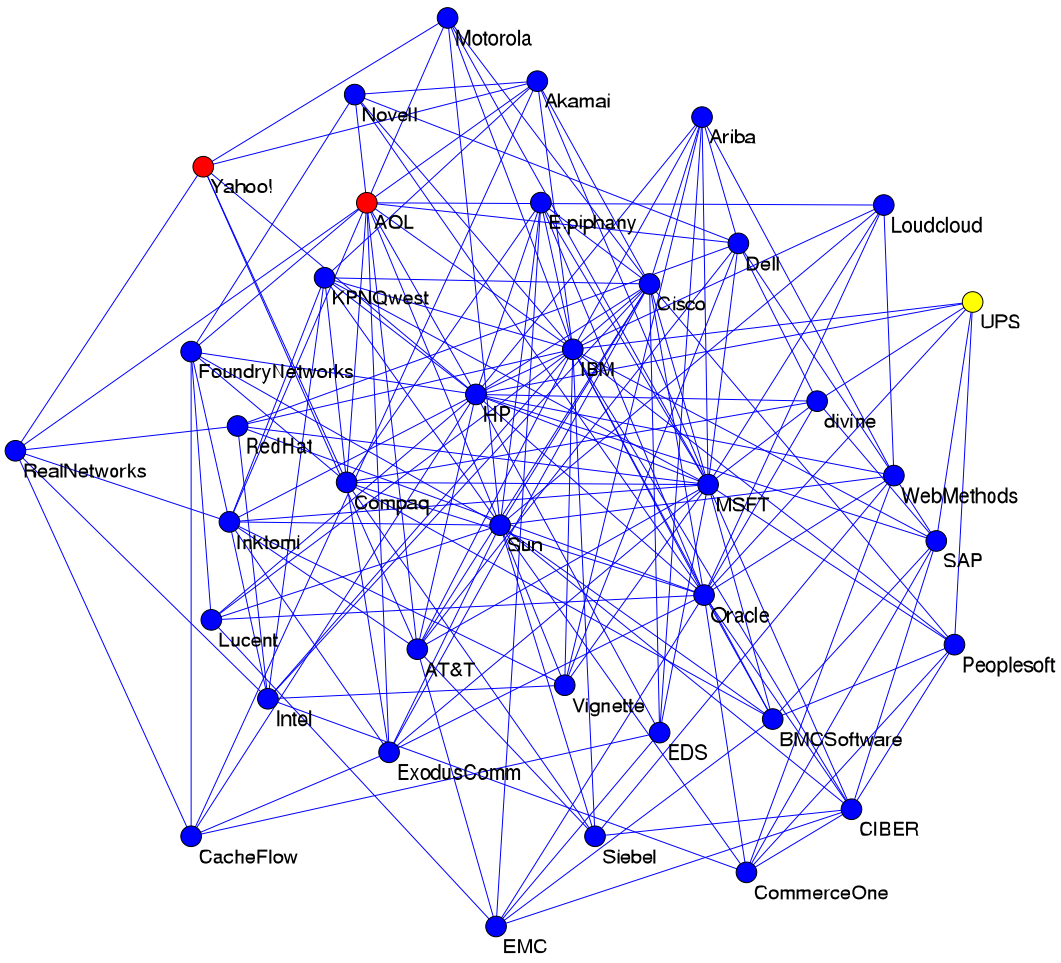
... postopek določanja sredic

Pri razdelavi postopka je potrebno poskrbeti za hitro urejanje tabele *degree* in njeno preurejanje. Ker so vrednosti stopenj med 0 in $n-1$ lahko s postopkom 'kupčkanja' to naredimo v času $O(n)$; urejenost lahko popravljamo v konstantnem času. Celoten postopek lahko izvedemo v času $O(m)$.

Sredice so eden od redkih učinkovitih pristopov za določanje gostih delov grafa. Uporabimo jih lahko tudi za pohitritve drugih, zahtevnejših postopkov saj so na primer k-komponente in klike reda k vsebovane v sredici reda k. Velja tudi naslednja ocena za barvnost grafa $\mathcal{G}$

$$\chi(\mathcal{G}) \leq 1 + \mathrm{core}(\mathcal{G})$$

Sredica reda 6 za Krebsova Internetska podjetja



Posplošene sredice

Pojem sredice lahko posplošimo na omrežja. Naj bo $\mathcal{N}$ omrežje nad grafom $\mathcal{G} = (\mathcal{V}, \mathcal{L})$. Privzemimo, da imamo na $\mathcal{N}$ določeno funkcijo $p(v, \mathcal{U})$, $v \in \mathcal{V}$, $\mathcal{U} \subseteq \mathcal{V}$ z realnimi vrednostmi – *p-funkcijo*.

Označimo $N_{\mathcal{U}}(v) = N(v) \cap \mathcal{U}$. Poleg stopenj so p-funkcije npr. še:

$$
\begin{aligned}
p_S(v, \mathcal{U}) &= \sum_{u \in N_{\mathcal{U}}(v)} w(v, u), \text{ kjer je } w : \mathcal{L} \to \mathbb{R}_0^+ \\
p_M(v, \mathcal{U}) &= \max_{u \in N_{\mathcal{U}}(v)} w(v, u), \text{ kjer je } w : \mathcal{L} \to \mathbb{R} \\
p_t(v, \mathcal{U}) &= \frac{|\mathcal{L}(\mathcal{U}) \cap \mathcal{L}(K(N^+(v)))|}{|\mathcal{L}(K(N^+(v)))|} \\
p_k(v, \mathcal{U}) &= \text{število ciklov dolžine } k \text{ skozi vozlišče } v \text{ v } (\mathcal{U}, \mathcal{L}(\mathcal{U}))
\end{aligned}
$$

Podgraf $\mathcal{H} = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$ porojen z množico $\mathcal{C} \subseteq \mathcal{V}$ je *p-sredica* na ravni $t \in \mathbb{R}$ ntk. $\forall v \in \mathcal{C} : t \leq p(v, \mathcal{C})$ in je $\mathcal{C}$ maksimalna taka množica.

Še nekaj p-funkcij

relativna gostota

$$p_\gamma(v,\mathcal{C}) = \frac{\deg(v,\mathcal{C})}{\max_{u\in N(v)} \deg(u)}, \text{ če je } \deg(v) > 0;\ 0, \text{ sicer}$$

raznolikost

$$p_\delta(v,\mathcal{C}) = \max_{u\in N^+(v,\mathcal{C})} \deg(u) - \min_{u\in N^+(v,\mathcal{C})} \deg(u)$$

povprečna utež

$$p_a(v,\mathcal{C}) = \frac{1}{|N(v,\mathcal{C})|} \sum_{u\in N(v,\mathcal{C})} w(v,u), \text{ če je } N(v,\mathcal{C}) \neq \emptyset;\ 0, \text{ sicer}$$

Postopek za posplošene sredice

Funkcija p je *monotona* ntk. zanjo velja

$$\mathcal{C}_1 \subset \mathcal{C}_2 \Rightarrow \forall v \in \mathcal{V} : (p(v, \mathcal{C}_1) \leq p(v, \mathcal{C}_2))$$

Stopnje in funkcije p_S, p_M, p_t ter p_k so monotone.

Za monotone funkcije lahko ustrezno p-sredico na ravni t določimo, kot pri navadnih sredicah, z zaporednim odmetavanjem vozlišč z vrednostjo p manjšo od t. Sredice na različnih ravneh so gnezdene

$$t_1 < t_2 \Rightarrow \mathcal{H}_{t_2} \subseteq \mathcal{H}_{t_1}$$

p-funkcija je *lokalna* ntk. $p(v, \mathcal{U}) = p(v, N_{\mathcal{U}}(v))$.

Stopnje, p_S, p_M in p_t so lokalne; toda p_k **ni** lokalna za $k \geq 4$. Za lokalno p-funkcijo obstaja postopek reda $O(m \max(\Delta, \log n))$ za določitev razslojitve na p-sredice, če le lahko $p(v, N_{\mathcal{C}}(v))$ določimo v času $O(\deg_{\mathcal{C}}(v))$.

Omrežje sodelovanj GeomBib

Iz B. Jones-ove bibliografije računalniške geometrije *Computational geometry database* (2002), *FTP*, *Geom.net*. smo sestavili omrežje sodelovanj. Avtorja določata povezavo, če sta napisala skupno delo. Utež povezave je enaka številu skupnih del.

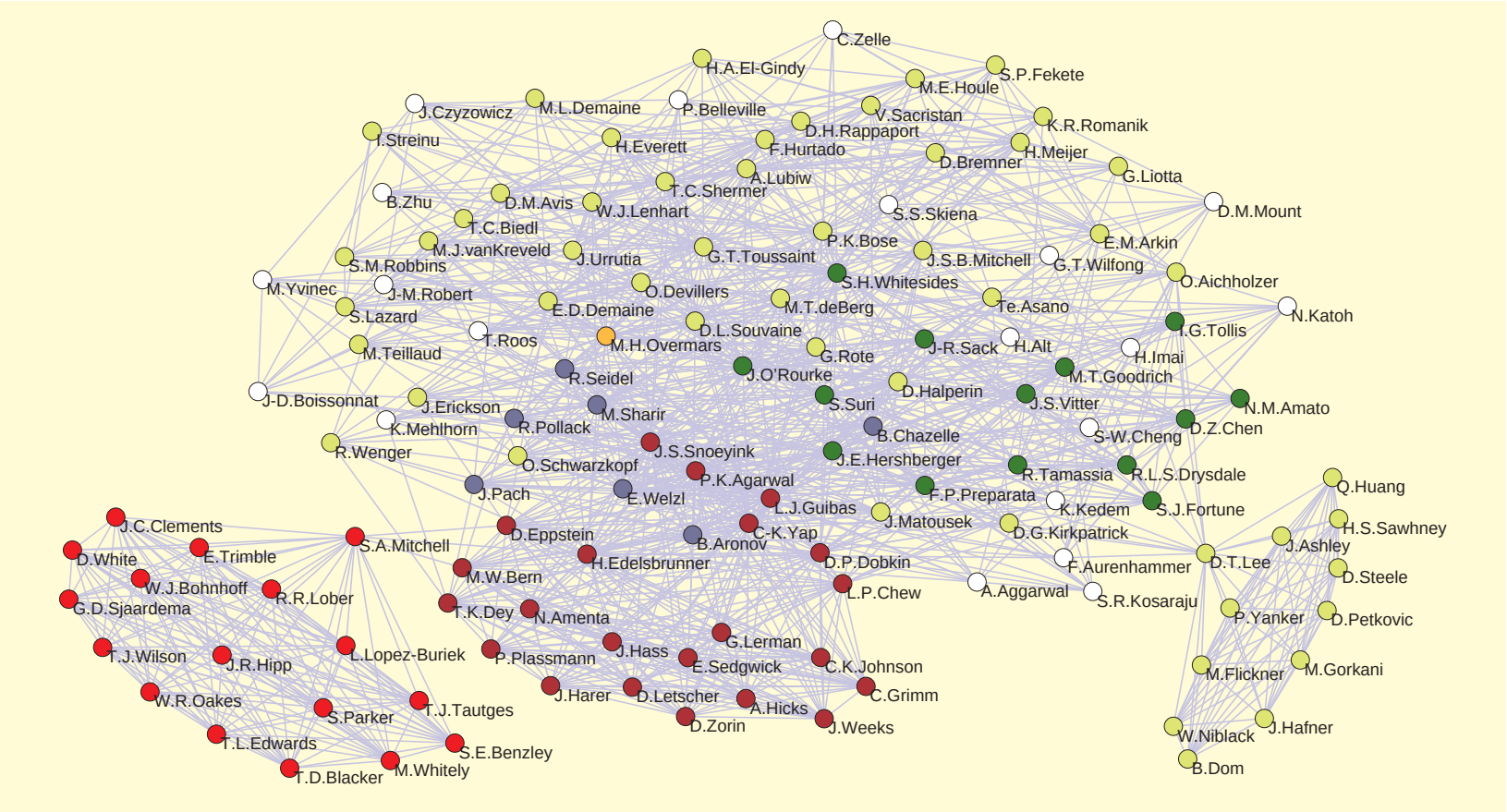
To omrežje je zahtavalo veliko 'čiščenja' zaradi težav z razpoznavanjem posameznih avtorjev. Tako na primer: Pankaj K. Agarwal, P. Agarwal, Pankaj Agarwal in P.K. Agarwal določajo isto osebo.

$n = 9072,\ m = 13567/22577 \rightarrow n' = 7343,\ m' = 11898.$

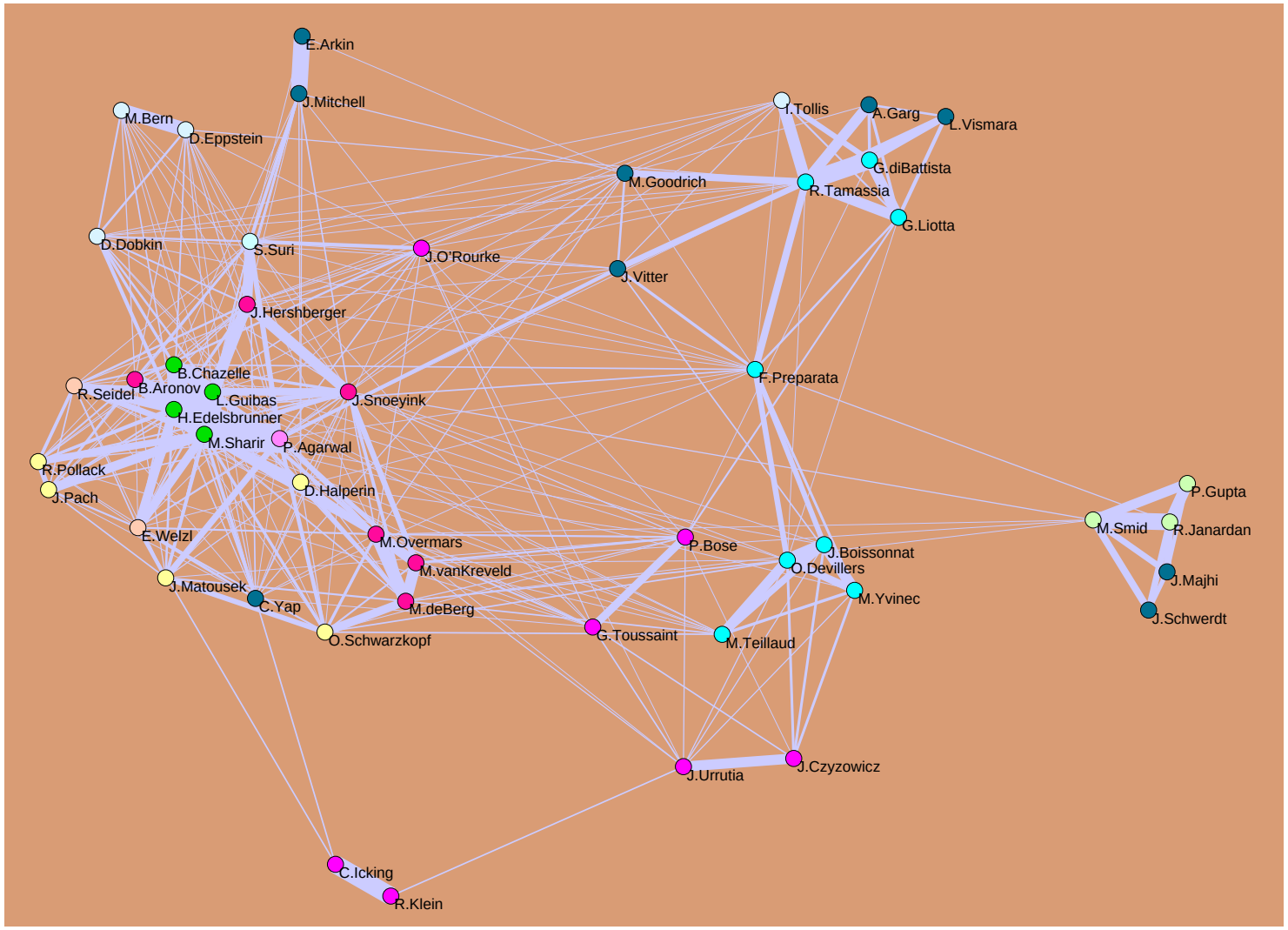
Sredice in posplošene sredice / ukazi Pajek

```
File/Network/Read  [Geom.net]
Network/Create Partition/k-Core/All
Info/Partition
Operations/Network+Partition/Extract Subnetwork [13-*]
Draw/Network+First Partition
Layout/Energy/Kamada-Kawai
Options/Values of lines/Similarities
Layout/Energy/Kamada-Kawai
Operations/Network+Partition/Extract Subnetwork [21]
Draw/Network
Layout/Energy/Kamada-Kawai
Options/Values of lines/Forget
Layout/Energy/Kamada-Kawai
[select Geom.net]
Network/Create Vector/Generalized Cores/Sum/All
Info/Vector
Vector/Make Partition/by Intervals/Selected Thresholds [45]
Info/Partition
Operations/Network+Partition/Extract Subnetwork [2]
Draw/Network
Options/Values of lines/Similarities
Layout/Energy/Fruchterman-Reingold
```

Navadne sredice redov 10–21 v Computational Geometry



p_S-sredica na ravni 46 v Computational Geometry



Aciklična omrežja

Omrežja sklicevanj

Rodovniki

Trojice

Motivi

Grafki

Analiza omrežij

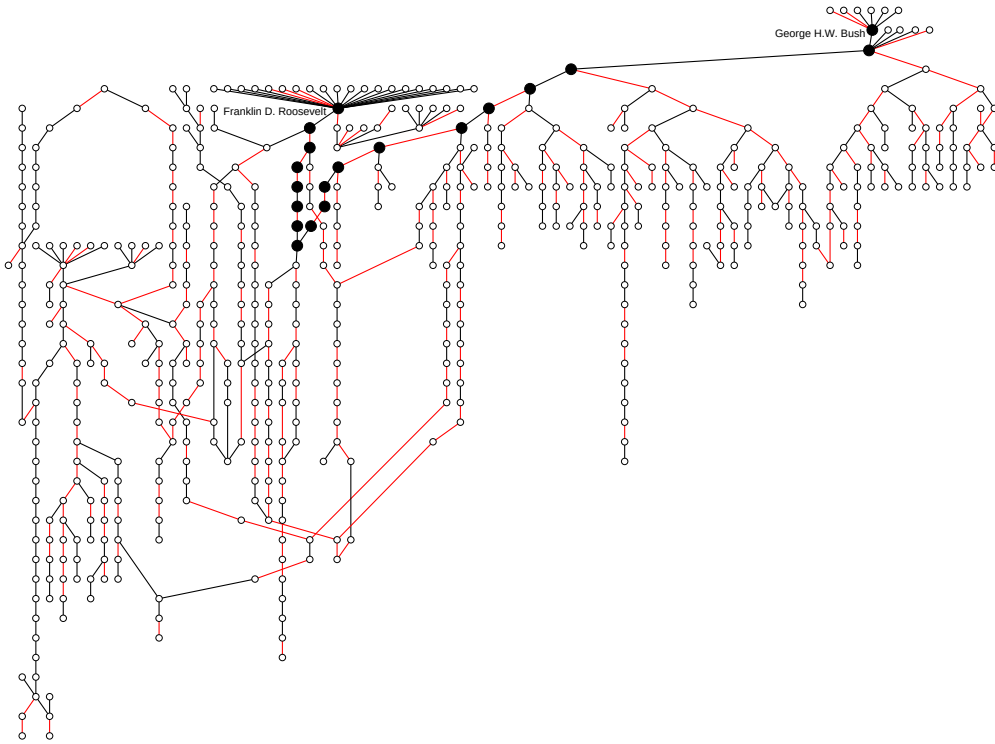
7. Aciklična omrežja in vzorci

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, maj 2024

Kazalo

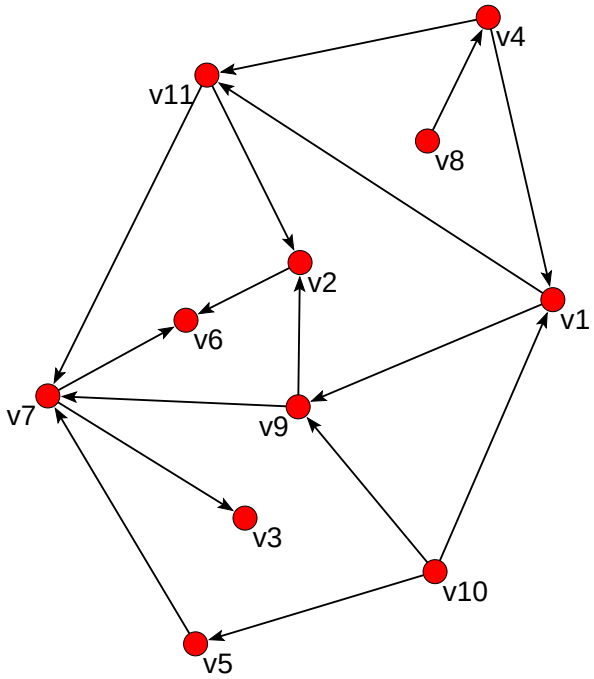
1 Aciklična omrežja
2 Omrežja sklicevanj
3 Rodovniki
4 Trojice
5 Motivi
6 Grafki



prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

6. maj 2024 ob 09 : 38/ april 2013

Aciklična omrežja



acyclic.paj

Omrežje $\mathcal{G} = (\mathcal{V}, \mathbf{R})$, $\mathbf{R} \subseteq \mathcal{V} \times \mathcal{V}$ je *aciklično*, če v njem ni nobenega (pravega) cikla.

$$\overline{\mathbf{R}} \cap I = \emptyset$$

$\overline{\mathbf{R}}$ je tranzitivna ovojnica relacije $\mathbf{R}$ – relacija *dosegljivosti*.
Včasih dopuščamo zanke $\overline{\mathbf{R} \setminus I} \cap I = \emptyset$.
Primeri acikličnih omrežij so: omrežja sklicevanj, rodovniki, projektna omrežja, ...

V dejanskih acikličnih omrežjih običajno obstaja lastnost $p : \mathcal{V} \to \mathbb{R}$ (najpogosteje čas), ki je usklajena s povezavami

$$(u, v) \in \mathbf{R} \Rightarrow p(u) < p(v)$$

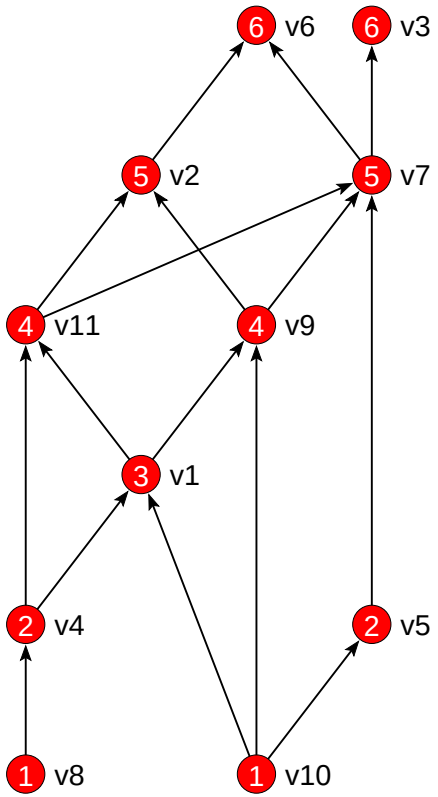
Osnovne lastnosti

Če je $\mathcal{G} = (\mathcal{V}, \mathbf{R})$ acikličen in $\mathcal{U} \subseteq \mathcal{V}$, je tudi $\mathcal{G}|\mathcal{U} = (\mathcal{U}, \mathbf{R}|\mathcal{U})$, $\mathbf{R}|\mathcal{U} = \mathbf{R} \cap \mathcal{U} \times \mathcal{U}$ acikličen.
Če je $\mathcal{G} = (\mathcal{V}, \mathbf{R})$ acikličen, je tudi $\mathcal{G}' = (\mathcal{V}, \mathbf{R}^{-1})$ acikličen. Dualnost.

Množica *začetkov* $\mathrm{Min}_{\mathbf{R}}(\mathcal{V}) = \{v : \neg\exists u \in \mathcal{V} : (u, v) \in \mathbf{R}\}$ in množica *koncev* $\mathrm{Max}_{\mathbf{R}}(\mathcal{V}) = \{v : \neg\exists u \in \mathcal{V} : (v, u) \in \mathbf{R}\}$ sta v končnem acikličnem omrežju neprazni.

Tranzitivna ovojnica $\overline{\mathbf{R}}$ aciklične relacije $\mathbf{R}$ je aciklična.
Relacija Q je *ogrodje* relacije $\mathbf{R}$ ntk. je $Q \subseteq \mathbf{R}$, $\overline{Q} = \overline{\mathbf{R}}$ in je relacija Q minimalna – iz nje ne moremo odstraniti nobene povezave, ne da bi 'pokvarili' drugo enakost.
Za splošne relacije (grafe) lahko obstaja več ogrodij; za aciklične pa je enolično določeno $Q = \mathbf{R} \setminus \mathbf{R} * \overline{\mathbf{R}}$.

Globina

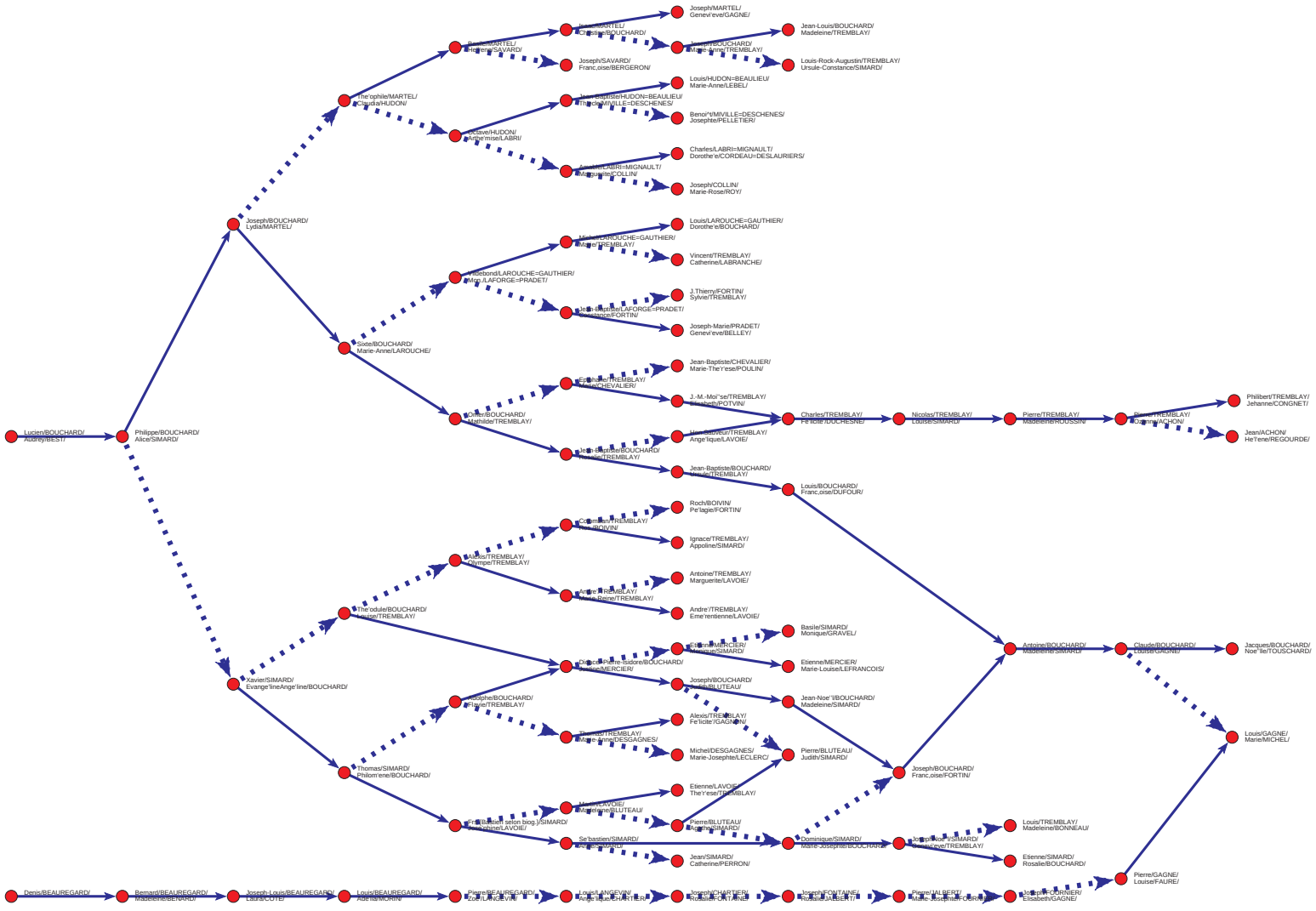


Preslikava $h : \mathcal{V} \to \mathbb{N}^+$ je *globina*, če so vse razlike na najdaljši poti in vrednost v začetku enake 1.

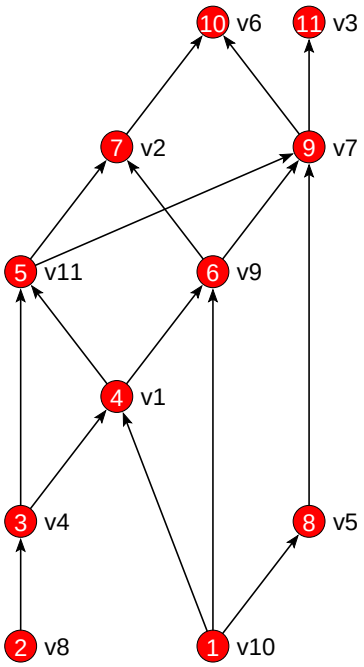
$\mathcal{U} \leftarrow \mathcal{V}$; $k \leftarrow 0$
while $\mathcal{U} \neq \emptyset$ **do**
  $\mathcal{T} \leftarrow \mathrm{Min}_R(\mathcal{U})$; $k \leftarrow k + 1$
  for $v \in \mathcal{T}$ **do** $h(v) \leftarrow k$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{T}$

Risanje po ravneh. Macro `Layers`.
Druge globine. Algoritem Sugiyama.

Prikaz parnega grafa Bouchardovega rodovnika



Pravilna oštevilčenja

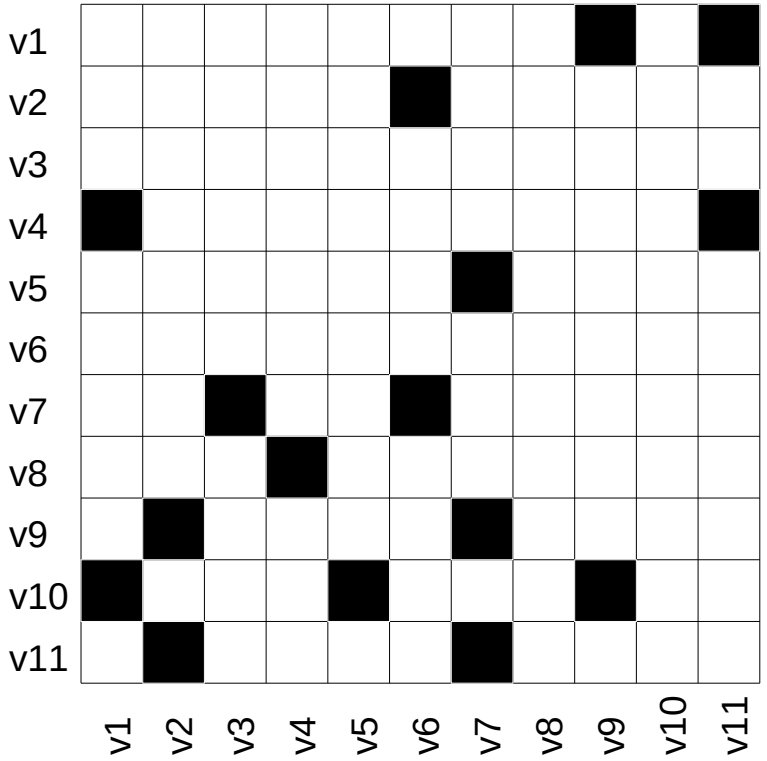


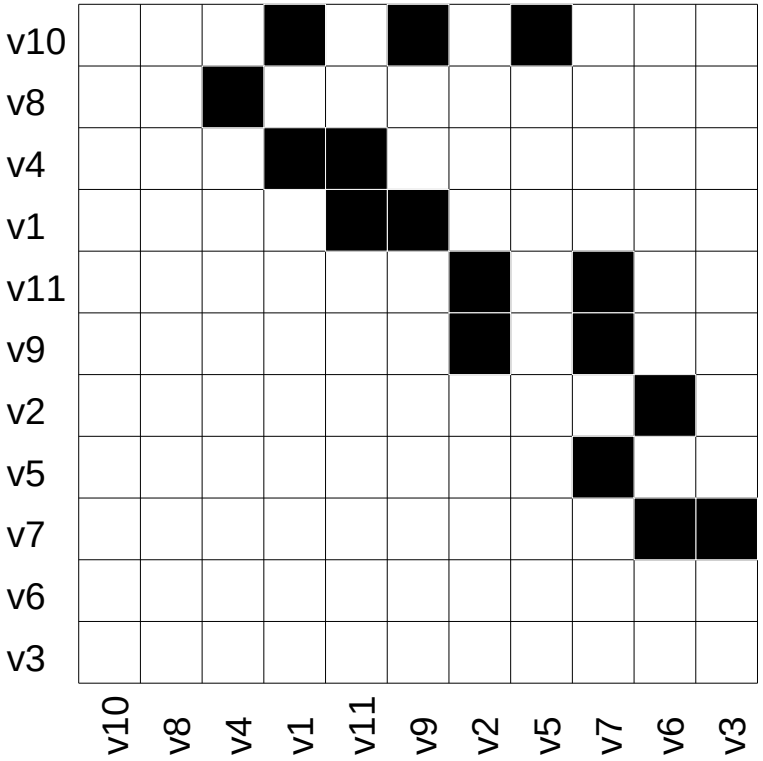
Injektivna, z relacijo **R** usklajena preslikava $h : \mathcal{V} \to 1..|\mathcal{V}|$ je *pravilno oštevilčenje*.
'Topološko urejanje'

$\mathcal{U} \leftarrow \mathcal{V}$; $k \leftarrow 0$
while $\mathcal{U} \neq \emptyset$ **do**
  izberi $v \in \mathrm{Min}_{\mathbf{R}}(\mathcal{U})$; $k \leftarrow k + 1$
  $h(v) \leftarrow k$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{v\}$

Matrični prikaz acikličnega omrežja glede na pravilno oštevilčenje ima ničelni spodnji trikotnik.

... Pravilna oštevilčenja






```
read or select  [Acyclic.paj]
Network/Acyclic Natwork/Depth Partition/Acyclic
Partition/Make Permutation
File/Network/Export as Matrix to EPS/Using Permutation [acy.eps]
```

Pravilna oštevilčenja in izračun vrednosti

Naj bo funkcija $f : \mathcal{V} \to \mathbb{R}$ definirana na sosedih takole:

- $f(v)$ je znana za $v \in \mathrm{Min}_{\mathbf{R}}(\mathcal{V})$
- $f(v) = F(\{f(u) : u\mathbf{R}v\})$

Če vrednosti funkcije f računamo v vrstnem redu določenem z nekim pravilnim oštevilčenjem, dobimo vse vrednosti v enem prehodu – saj so za vsak $v \in \mathcal{V}$ vrednosti, ki jih potrebujemo pri izračunu $f(v)$ že določene.

Pravilna oštevilčenja – primer CPM

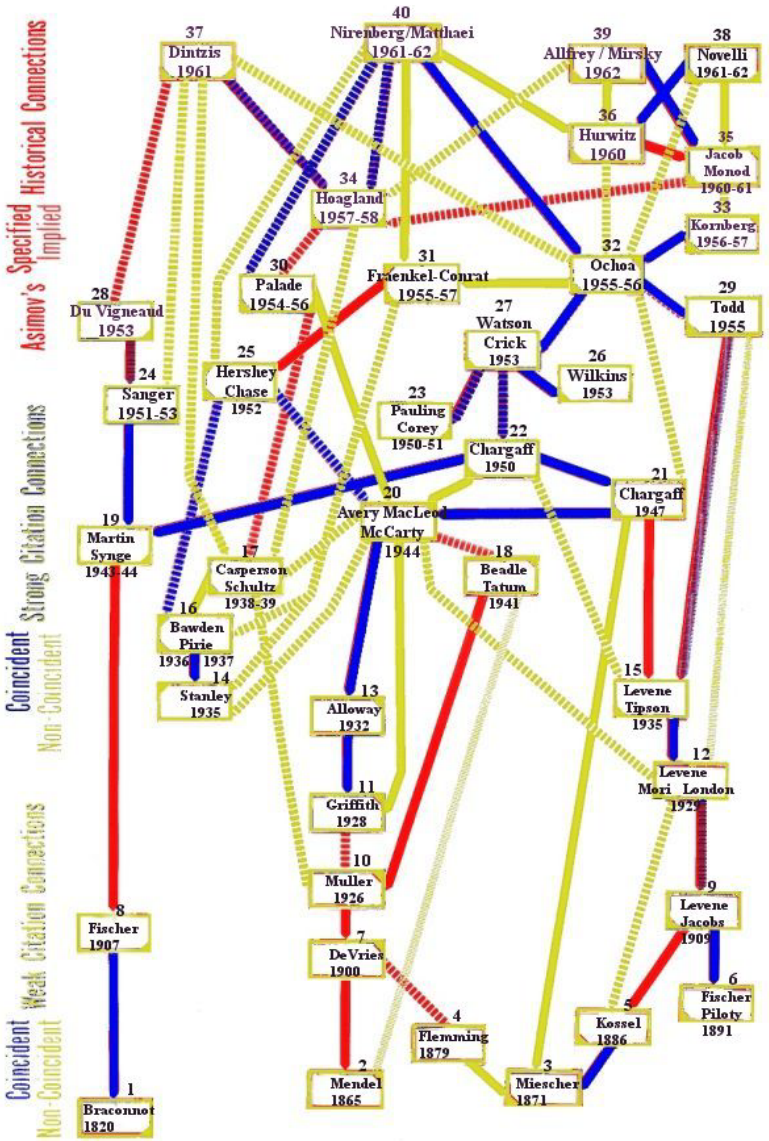
CPM (Critical Path Method): Projekt je sestavljen iz posameznih opravil. Vozlišča predstavljajo stanja projekta, povezave pa opravila. Za vsako opravilo (u, v) poznamo čas njegovega trajanja $t(u, v)$. Neko opravilo se lahko začne izvajati šele, ko so vsa opravila, ki se končajo v njegovem začetku zaključena. Projektno omrežje je aciklično. Zanima nas, koliko najmanj časa je potrebno za izvedbo projekta.
Naj bo $T(v)$ čas najzgodnejšega zaključka vseh opravil v stanju v.

$$T(v) = 0, \qquad v \in \mathrm{Min}_{\mathbf{R}}(\mathcal{V})$$

$$T(v) = \max_{u:u\mathbf{R}v}(T(u) + t(u, v))$$

`Network/Acyclic Network/Critical Path Method - CPM`

Omrežja sklicevanj



Analiza omrežij sklicevanj se je pričela leta 1964 s člankom Garfield et al. Leta 1989 sta Hummon in Doreian predlagala tri mere pomembnosti – uteži na povezavah, ki omogočajo računalniško določitev (naj)pomembnejših delov omrežja sklicevanj. Za dve izmed njih obstajata zelo učinkovita postopka za njun izračun.

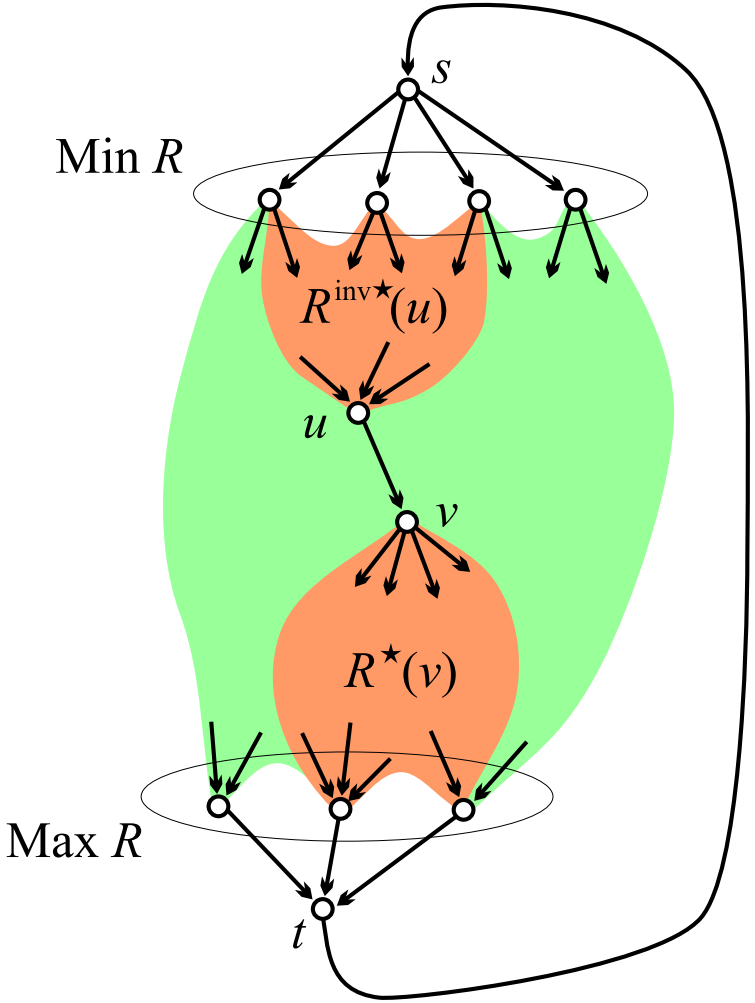
... omrežja sklicevanj

V dani množici enot/vozlišč $\mathcal{U}$ (članki, knjige, druga dela, itd.) vpeljemo *relacijo sklicevanja*/množico usmerjenih povezav $\mathbf{R} \subseteq \mathcal{U} \times \mathcal{U}$

$$u\mathbf{R}v \equiv v \text{ se sklicuje na } u$$

ki določa *omrežje sklicevanj* $\mathcal{N} = (\mathcal{U}, \mathbf{R})$.
Relacija sklicevanja je običajno *irrefleksivna* (ni zank) in (skoraj) *aciklična*. V nadaljevanju bomo privzeli, da ima ti dve lastnosti.
V dejanskih omrežjih sklicevanj se včasih pojavijo manjše (praviloma 2 ali 3 vozlišča) krepke komponente. Tako omrežje najenostavneje pretvorimo v aciklično tako, da skrčimo krepke komponente in odvržemo zanke, ki pri tem nastanejo. Obstajajo tudi druge možnosti.
Omrežje sklicevanj je koristno dopolniti v *standardno* obliko, tako da mu dodamo skupni *izvor* $s \notin \mathcal{U}$ in skupni *ponor* $t \notin \mathcal{U}$. Izvor s je neposredno povezan v vse minimalne elemente relacije $\mathbf{R}$; ponor t pa iz vseh maksimalnih elementov relacije $\mathbf{R}$. Dodamo še *povratno* povezavo (t, s).

Štetje poti – Search path count method



Metoda *štetja poti* – *search path count method* (SPC) temelji na števcih $n(u, v)$, ki štejejo število različnih poti iz izvora s v ponor t, ki gredo čez povezavo (u, v). Za izračun števcev $n(u, v)$ vpeljemo dve pomožni količini: $n^{-}(v)$ šteje število poti iz izvora s v vozlišče v, in $n^{+}(v)$ šteje število poti iz vozlišča v v ponor t.

Hitri postopek za izračun števila poti

Iz osnovnih načel kombinatorike izhaja

$$n(u,v) = n^-(u) \cdot n^+(v), \qquad (u,v) \in \mathbf{R}$$

kjer je

$$n^-(u) = \begin{cases} 1 & u = s \\ \sum_{v:v\mathbf{R}u} n^-(v) & sicer \end{cases}$$

in

$$n^+(u) = \begin{cases} 1 & u = t \\ \sum_{v:u\mathbf{R}v} n^+(v) & sicer \end{cases}$$

Ta zveza je osnova za hiter izračun števcev $n(u,v)$ – najprej vozlišča grafa topološko uredimo in nato uporabljamo v dobljenem vrstnem redu gornjo zvezo. Postopek ima časovno zahtevnost reda $O(m)$, $m = |\mathbf{R}|$. Topološka urejenost zagotavlja, da so v gornji zvezi vse količine že izračunane, ko jih potrebujemo.

Hummon in Doreian-ove uteži in SPC

Hummon in Doreian-ove uteži so takole določene:

- *search path link count* (SPLC) method: utež $w_l(u, v)$ je enaka številu "*vseh možnih poti po omrežju, ki imajo začetek v izvoru*" in gredo po povezavi $(u, v) \in \mathbf{R}$.
- *search path node pair* (SPNP) method: $w_p(u, v)$ "*upošteva vse povezanosti parov vozlišč s potmi, ki gredo po povezavi* $(u, v) \in \mathbf{R}$".

Uteži SPLC lahko določimo tako, da uporabimo postopek SPC na omrežju, ki ga dobimo, če v standardni obliki omrežja povežemo izvor s s povezavo še z vsemi neminimalnimi vozlišči iz $\mathcal{U}$; za uteži SPNP moramo temu omrežju dodati še povezave iz vseh nemaksimalnih vozlišč iz $\mathcal{U}$ v ponor t.

Vozliščne uteži

Količine uporabljene za določitev povezavnih uteži w lahko uporabimo tudi za določitev ustreznih vozliščnih vrednosti t

$$
\begin{aligned}
t_c(u) &= n^-(u) \cdot n^+(u) \\
t_l(u) &= n_l^-(u) \cdot n_l^+(u) \\
t_p(u) &= n_p^-(u) \cdot n_p^+(u)
\end{aligned}
$$

Te štejejo število poti izbrane vrste skozi vozlišče u.
V programu `Pajek` dobimo hkrati uteži w in lastnost t.

```
Network/Acyclic Network/Citation Weights/Search Path Count (SPC)
Network/Acyclic Network/Citation Weights/Search Path Link Count (SPLC)
Network/Acyclic Network/Citation Weights/Search Path Node Pair (SPNP)
```

Lastnosti uteži SPC

Vrednosti števcev $n(u,v)$ določajo tok po omrežju sklicevanj – zanj velja *Kirchoffov vozliščni zakon*: v vsakem vozlišču u standardnega omrežja sklicevanj velja *vstopni tok* = *izstopni* tok:

$$\sum_{v:v\mathbf{R}u} n(v,u) = \sum_{v:u\mathbf{R}v} n(u,v) = n^-(u)\cdot n^+(u)$$

Utež $n(t,s)$ je enaka celotnemu toku skozi omrežje in ponuja naravni način normalizacije uteži

$$w(u,v) = \frac{n(u,v)}{n(t,s)} \quad \Rightarrow \quad 0 \le w(u,v) \le 1$$

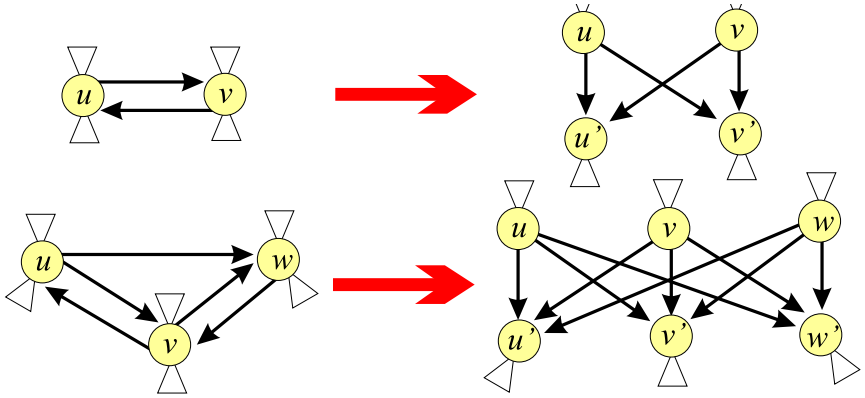
in, če je C minimalni povezavni prerez, velja še $\sum_{(u,v)\in C} w(u,v) = 1$. V velikih omrežjih lahko uteži postanejo zelo velike. Na to je potrebno biti pazljiv pri programiranju algoritma.

Omrežja sklicevanj s cikli

Če v omrežju obstaja cikel, obstaja med nekaterimi vozlišči neskončno število sprehodov. Nastali problem lahko poskušamo rešiti na več načinov: z vpeljavo 'staranja', ki zagotovi, da celotna utež sprehodov gre proti neki končni vrednosti; ali omejimo definicijo uteži na neko končno podmnožico sprehodov – npr. poti ali najkrajše poti. Toda to ustvari nova vprašanja: Kolikšen naj bo faktor 'staranja'? Ali je mogoče učinkovito prešteti (najkrajše) poti?
Druga možnost je, ker so omrežja sklicevanj skoraj aciklična, da jih pretvorimo v aciklična:

- skrčimo vsako ciklično skupino (netrivialno krepko komponento) v vozlišče, ali
- razklenemo cikle z odstranitvijo nekaj povezav, ali
- z uporabo ustreznih transformacij (glej naslednjo prosojnico).

Transformacija Preprint



Transformacija *preprint* temelji na naslednji zamisli: Vsako vozlišče (delo) iz krepke komponente se podvoji s svojo predobjavo (preprint). Dela znotraj komponente se sklicujejo na predobjave.

Velike krepke komponente v dejanskih omrežjih sklicevanj običajno pomenijo napako v podatkih. Seveda je to odvisno od načina izgradnje omrežja. Tako, na primer ima omrežje HEP – High Energy Particle Physics z **arXiv** veliko večjih krepkih komponent, ker so kot ena enota obravnavane vse različice istega članka. Tudi v tem omrežju bi si lahko pomagali s transformacijo 'preprint'.

Omrežje SOM

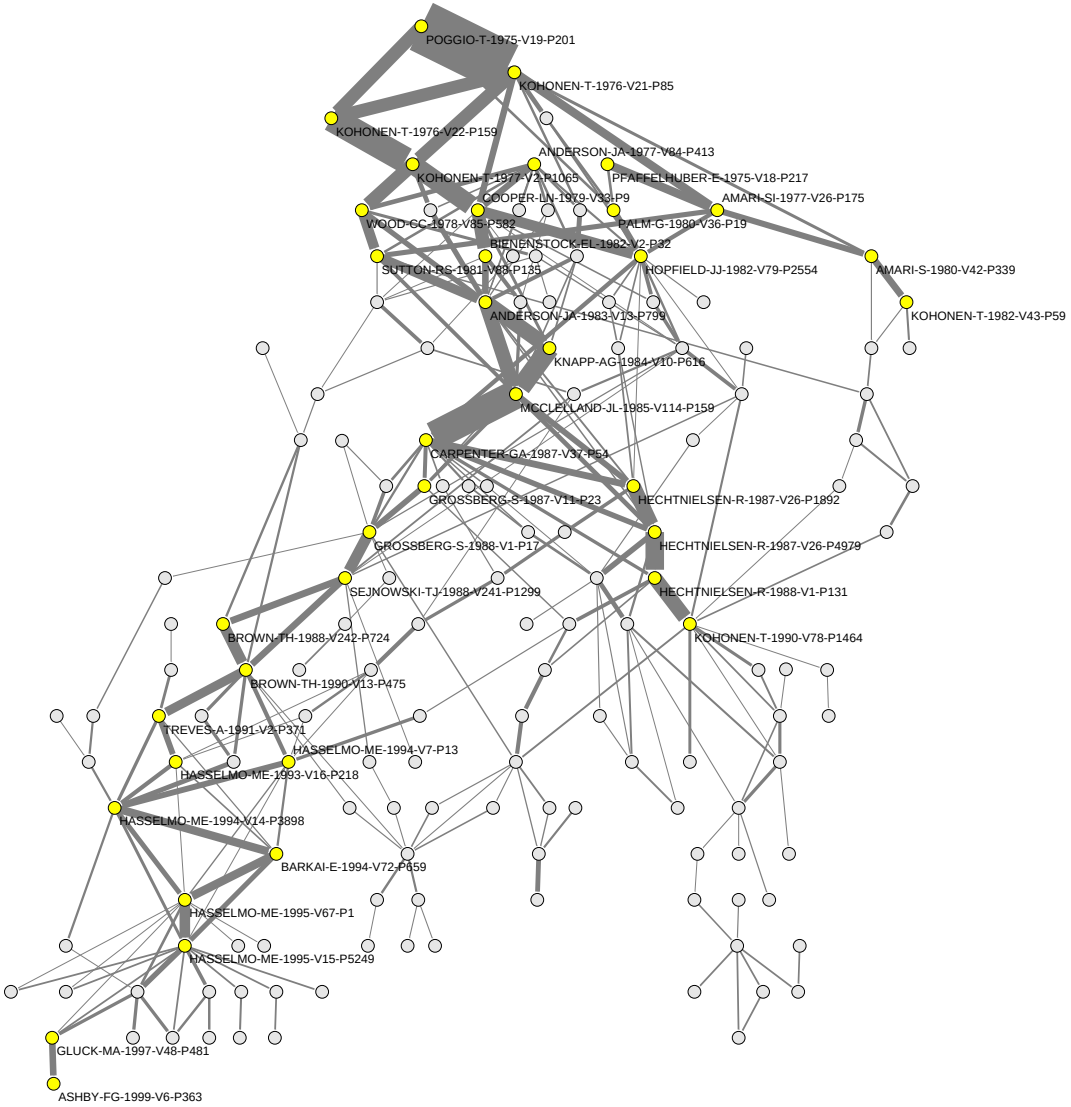
Poglejmo si omrežje sklicevanj za področje ($n = 4470$, $m = 12731$) SOM (*self-organizing maps*). Določimo uteži SPC.

Pri pregledu porazdelitve vrednosti uteži smo izbrali prag 0.007 in določili pripadajoči povezavni izerez. Odstranimo še vse male komponente ($k = 5$). Ostane ena sama komponenta. Narišemo jo po plasteh in ročno popravimo sliko..

Na sliki označimo samo pomembnejša dela – krajišča povezav z utežjo vsaj 0.05..

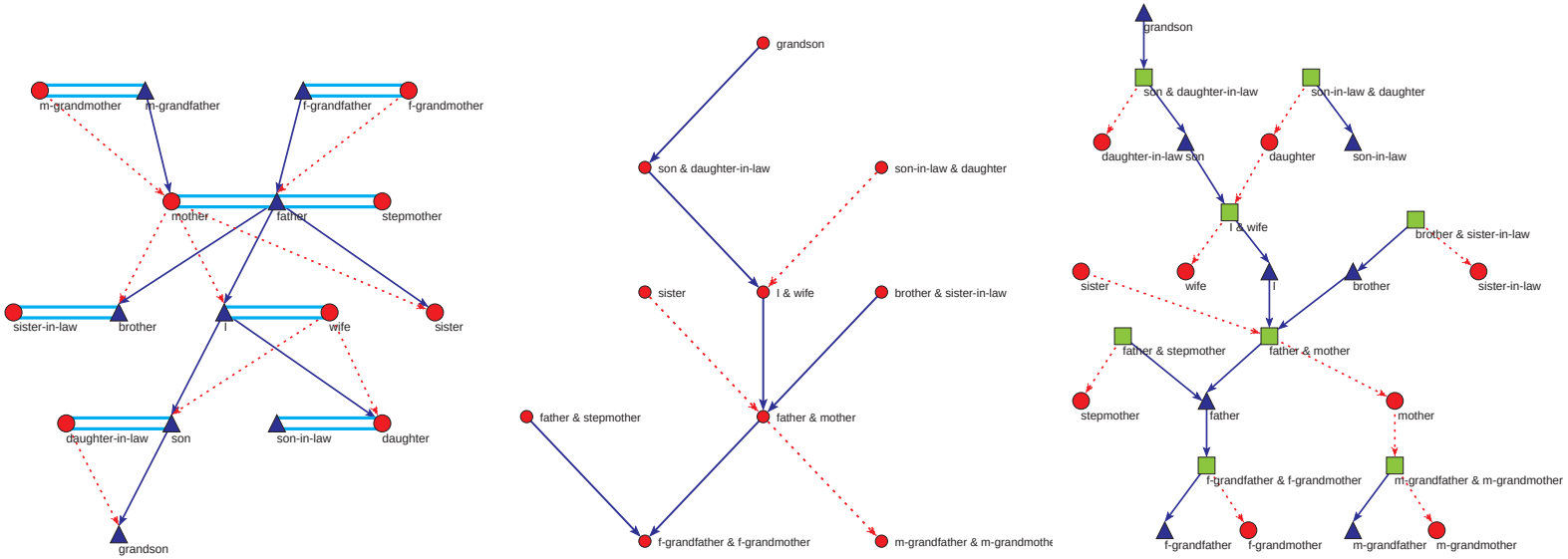
S slike vidimo, da v razvoju področja SOM obstajata vsaj dve veji razvoja.

Omrežje SOM – povezavni izrez 0.007



Rodovniki

Naslednji primer acikličnih omrežij so rodovniki. V poglavju 'Omrežja vsepovsod' smo že opisali naslednje tri omrežne predstavitve rodovnikov:



Orejev graf, parni graph, in dvodelni parni graph

Mormoni, Škofja Loka, ameriški predsedniki; računalništvo, matematika.

Primerjava predstavitev rodovnikov

Parni grafi imajo več prednosti (White et al., 1999):

- v parnem grafu je manj vozlišč in povezav kot v Orejevem grafu;
- parni grafi so usmerjeni aciklični grafi;
- vsaka sklenjena veriga v parnem grafu pomeni *poroko med sorodniki*. Obstajata dve vrsti teh porok: *poroke med krvnimi sorodniki*: npr. poroka med bratrancem in sestrično. *poroke med nekrvnimi sorodniki*: npr. brata se poročita s sestrama iz druge družine.
- parni grafi so veliko prikladnejši za analize.

Dvodelni parni grafi natančneje opisujejo rodovnik – npr. omogočajo razlikovati dve poroki enega izmed bratov od enkratnih porok vsakega izmed njiju; omogočajo razkriti poroke med polbrati in polsestami.

Rodovniki so redka omrežja

Rodovnik je *navaden*, če ima vsaka oseba največ dva starša.
Rodovniki so *redka* omrežja – število povezav je istega reda kot število vozlišč.
Za *navaden Orejev rodovnik* velja ($\mathcal{V}$ – vozlišča, $\mathcal{A}$ – usmerjene povezave, $\mathcal{E}$ – neusmerjene povezave):

$$|\mathcal{A}| \leq 2|\mathcal{V}|, \quad |\mathcal{E}| \leq \frac{1}{2}|\mathcal{V}|, \quad |\mathcal{L}| = |\mathcal{A}| + |\mathcal{E}| \leq \frac{5}{2}|\mathcal{V}|$$

Parni grafi so skoraj drevesa – odstopanja od dreves nastanejo zaradi porok med sorodniki ($\mathcal{V}_p$, $\mathcal{A}_p$ – vozlišča in povezave parnega grafa):

$$|\mathcal{V}_p| = |\mathcal{V}| - |\mathcal{E}| + n_{mult}, \quad |\mathcal{V}| \geq |\mathcal{V}_p| \geq \frac{1}{2}|\mathcal{V}|, \quad |\mathcal{A}_p| \leq 2|\mathcal{V}_p|$$

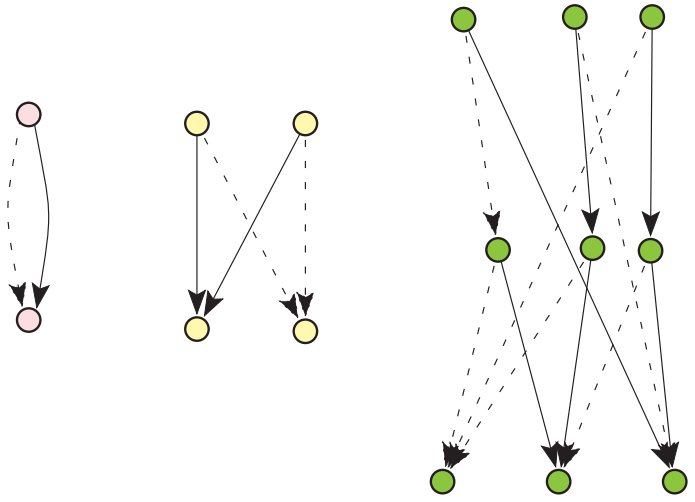
Za dvodelni parni graf pa velja:

$$|\mathcal{V}| \leq |\mathcal{V}_b| \leq \frac{3}{2}|\mathcal{V}|, \quad |\mathcal{A}_b| \leq 2|\mathcal{V}| + n_{mult}$$

Število vozlišč in povezav v Orejevih in parnih grafih

data	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}\rvert$	$\frac{\lvert\mathcal{L}\rvert}{\lvert\mathcal{V}\rvert}$	$\lvert\mathcal{V}_i\rvert$	n_{mult}	$\lvert\mathcal{V}_p\rvert$	$\lvert\mathcal{A}_p\rvert$	$\frac{\lvert\mathcal{A}_p\rvert}{\lvert\mathcal{V}_p\rvert}$
Drame	29606	8256	41814	1.69	13937	843	22193	21862	0.99
Hawlina	7405	2406	9908	1.66	2808	215	5214	5306	1.02
Marcus	702	215	919	1.62	292	20	507	496	0.98
Mazol	2532	856	3347	1.66	894	74	1750	1794	1.03
President	2145	978	2223	1.49	282	93	1260	1222	0.97
RoyalE	17774	7382	25822	1.87	4441	1431	11823	15063	1.27
Loka	47956	14154	68052	1.71	21074	1426	35228	36192	1.03
Silba	6427	2217	9627	1.84	2263	270	4480	5281	1.18
Ragusa	5999	2002	9315	1.89	2347	379	4376	5336	1.22
Tur	1269	407	1987	1.89	549	94	956	1114	1.17
Royal92	3010	1138	3724	1.62	1003	269	2141	2259	1.06
Little	25968	8778	34640	1.67	8412				1.01
Mumma	34224	11334	45565	1.66	11556				1.00
Tilltson	42559	12796	54043	1.57	16967				1.00

Prepletenost



Naj bo n število vozlišč v parnem grafu, m število povezav, k število šibkih komponent, in M število koncev (maksimalnih vozlišč – vozlišč z izhodno stopnjo 0, $M \geq 1$).

Prepletenost rodovnika imenujemo število:

$$RI = \frac{k + m - n}{k + n - 2M}$$

Za grafe brez povezav postavimo $RI = 0$.

Velja $0 \leq RI \leq 1$. $RI = 0$ natanko takrat, ko je graf gozd. Obstajajo poljubno veliki rodovniki z $RI = 1$.

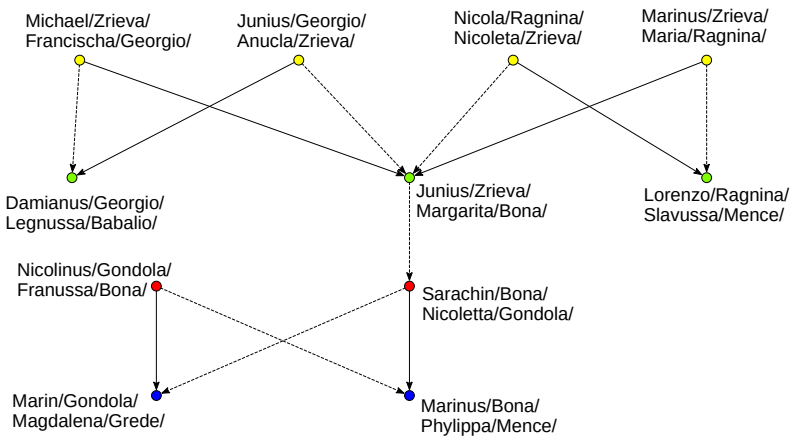
Iskanje vzorcev

Če se izbrani *vorec* določen z danim (majhnim) grafom ne pojavlja pogosto v redkem grafu, lahko njegove pojavitve razmeroma učinkovito poiščemo s preprostim pregledom vseh možnosti. Iskanje lahko pospešimo tako, da upoštevamo dodatne lastnosti vzorca:

- vozlišča vzorca in omrežja se morajo ujemati tudi v vrednosti izbrane lastnosti (npr. vrsta atoma v molekuli);
- ujemati se morajo uteži na povezavah (npr. moške/ženske povezave v *parnih grafih*);
- prvo vozlišče iz vzorca pripada dani skupini vozlišč.

Poroke med sorodniki v Ragusi

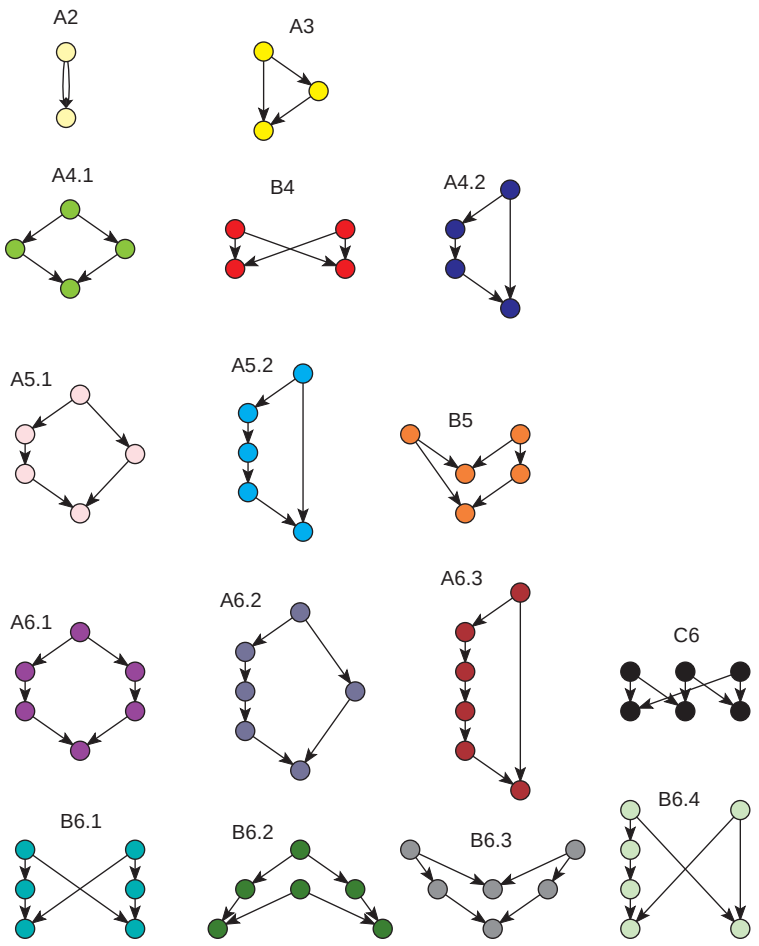
Iskanje vzorcev smo uporabili npr. pri analizi organskih molekul (iskanje ogljikovih obročev) in pri analizi porok med sorodniki v rodovnikih.



Slika prikazuje tri povezane skupine porok med sorodniki v rodovniku dubrovniškega (Ragusa) plemstva. Rodovnik je predstavljen kot parni graf. Polne povezave predstavljajo odnos _ *je sin od* _; pikčaste povezave pa odnos _ *je hči od* _. V vseh treh skupinah sta se brat in sestra iz ene družine poročila s sestro in bratom iz druge družine.

`Networks/Fragment (First in Second)/Find`

Poroke med sorodniki (parni grafi na 2 do 6 vozliščih)



Na sliki so prikazani vsi mogoči vzorci porok med sorodniki na 2 do 6 vozliščih v parnih grafih. Oznake imajo naslednji pomen:

- prvi znak – število začetkov: A – en, B – dva, C – tri.
- drugi znak: število vozlišč v vzorcu (2, 3, 4, 5, ali 6).
- tretji znak: različica, če sta prva znaka enaka.

V vsakem vzorcu je število začetkov enako številu koncev. Poroke med krvnimi sorodniki imajo en začetek in en konec – vse imajo oznako A.

Primerjava rodovnikov

Rodovnike lahko primerjamo, tako da primerjamo porazdelitve vzorcev v njih. Za primer smo vzeli naslednje rodovnike: `Loka.ged` (Škofja Loka), `Silba.ged` (otok Silba, Hrvaška), `Ragusa.ged` (dubrovniško plemstvo med 12. in 16. stoletjem, Dremelj et al., 2002) `Turcs.ged` (turški nomadi, White et al., 1999), `RoyalE.ged` (evropske kraljevske rodbine).

Vidimo:
generacijski preskok za več kot eno generacijo je (skoraj) neverjeten – vzorci A4.2, A5.2 in A6.3 se ne pojavljajo v rodovnikih, vzorec A6.2 se pojavi 2 krat na Silbi, vzorec B6.4 se pojavi 5 krat v Ragusai in 3 krat pri turkih). Pri turkih je zelo veliko porok vrst A4.1 in A6.1.

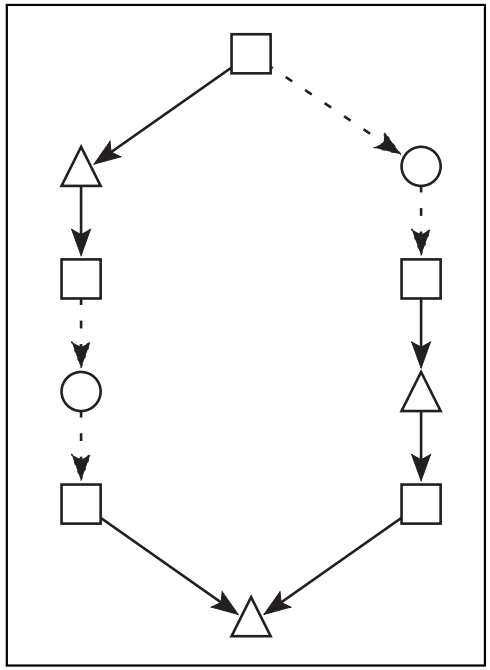
V vseh rodovnikih je število porok med nekrvnimi (vzorci B4, B5, C6, B6.1, B6.2, B6.3 in B6.4) sorodniki veliko večje od števila porok med krvnimi sorodniki. Razlogi za poroke med sorodniki so pogosto pogojeni s prizadevanji za ohranitev premoženja znotraj izbranih rodbin.

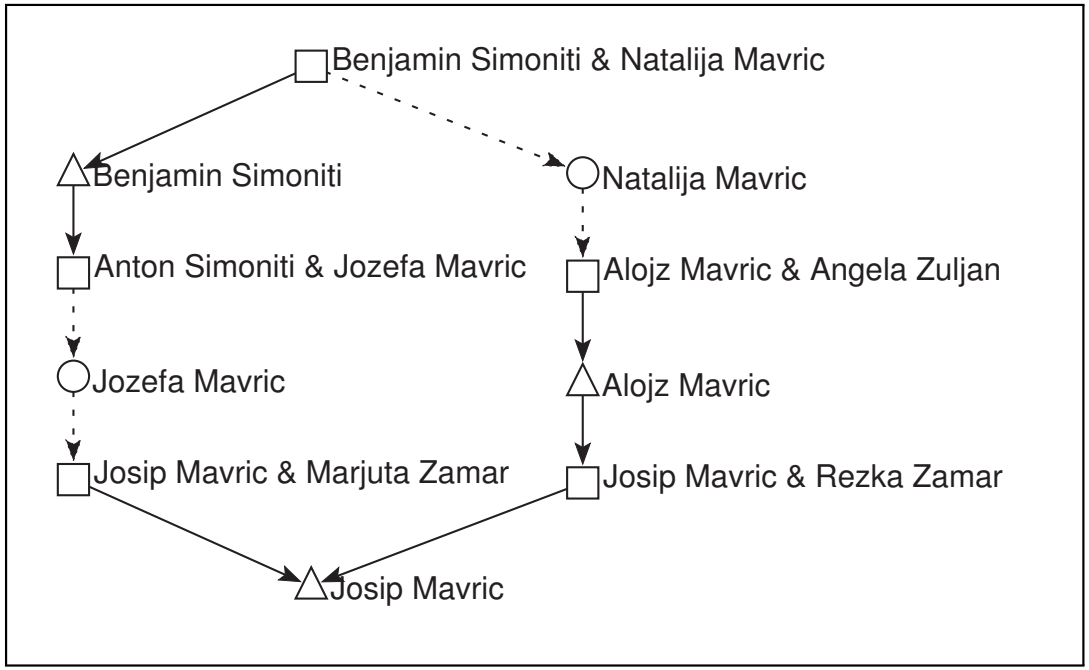
Normalizirane pogostosti vzorcev $\times$ 1000

pattern	Loka	Silba	Ragusa	Turcs	RoyalE
A2	0.07	0.00	0.00	0.00	0.00
A3	0.07	0.00	0.00	0.00	2.64
A4.1	0.85	2.26	1.50	**159.71**	18.45
B4	3.82	11.28	10.49	**98.28**	6.15
A4.2	0.00	0.00	0.00	0.00	0.00
A5.1	0.64	3.16	2.00	36.86	11.42
A5.2	0.00	0.00	0.00	0.00	0.00
B5	1.34	4.96	23.48	46.68	7.03
A6.1	1.98	12.63	1.00	**169.53**	11.42
A6.2	0.00	0.90	0.00	0.00	0.88
A6.3	0.00	0.00	0.00	0.00	0.00
C6	0.71	5.41	9.49	36.86	4.39
B6.1	0.00	0.45	1.00	0.00	0.00
B6.2	1.91	17.59	31.47	**130.22**	10.54
B6.3	3.32	13.53	40.96	**113.02**	11.42
B6.4	0.00	0.00	2.50	7.37	0.00
Sum	14.70	72.17	123.88	798.53	84.36

Zelo pogoste se poroke med sorodniki pri turkih; sledijo jim dubrovčani.

Dvodelni parni grafi: poroka med polbratrancem in polsestrično





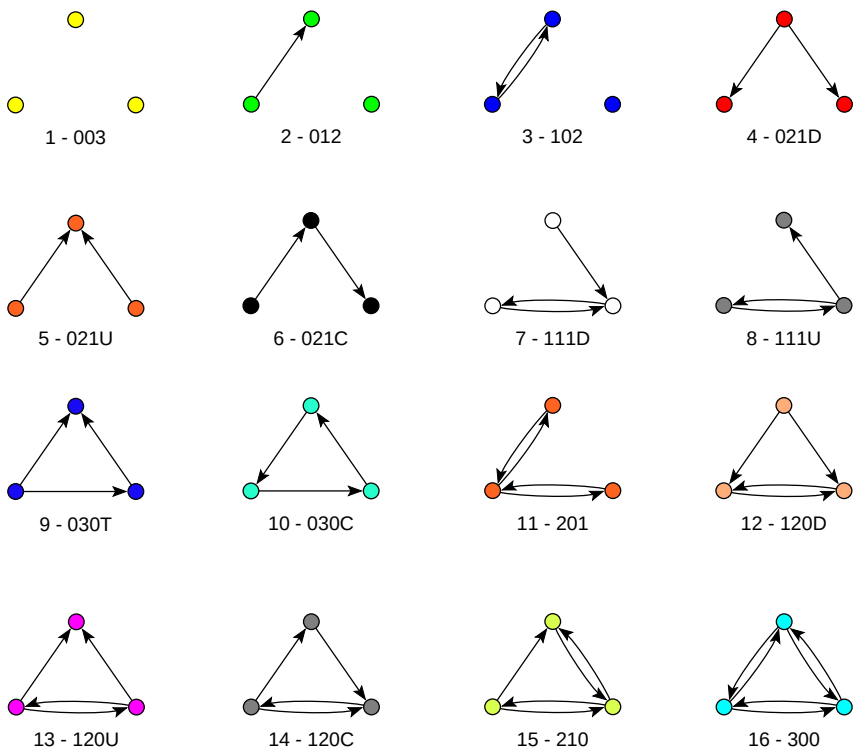
... rodovniki – druga vprašanja

Rodoslovce zanimajo še druga vprašanja:

- spremembe v pogostosti vzorcev skozi čas;
- posebnosti: velikokrat poročene osebe, osebe z veliko otrok;
- preverjanje sorodstvene povezanosti med osebama in iskanje najkrajše poti po rodovniku, če obstaja;
- določitev vse prednikov/potomcev izbrane osebe; določitev osebe, ki ima največ prednikov/potomcev;
- največja razlika v letih med možem in ženo; najstarejša/najmlajša oseba pri poroki; najstarejša/najmlajša oseba ob rojstvu otroka; ...
- določitev najdaljše moške/ženske verige;
- odkrivanje napak pri vnosu podatkov (preverjanje pravilnosti omrežja).

Na vsa ta vprašanja je mogoče odgovoriti z orodji programa `Pajek`.

Trojice



Naj bo $\mathcal{G} = (\mathcal{V}, \mathbf{R})$ enostaven usmerjen graf brez zank. *Trojica* je podgraf porojen z izbranimi tremi vozlišči.
Obstaja 16 neizomorfnih (vrst) trojic, ki so naprej razbite na tri skupine:

- the *ničelna* trojica 003;
- *dvočleni* trojici 012 in 102; ter
- *povezane* trojice: 111D, 201, 210, 300, 021D, 111U, 120D, 021U, 030T, 120U, 021C, 030C in 120C.

Trojiški spekter

Več lastnosti grafa je mogoče izraziti z uporabo *trojiškega spektra* – porazdelitve vseh njegovih trojic. Ta je tudi osnovna sestavina za modele omrežij p^*. Običajni postopek za določitev trojiškega spektra je reda $O(n^3)$; no, v večini velikih omrežij ga je mogoče določiti precej hitreje. Algoritem temelji na naslednjem opažanju: *v velikem in redkem omrežju je večina trojic ničelnih*. Označimo s T_1, T_2, T_3 zaporedoma število ničelnih, dvočlenih in povezanih trojic. Ker je celotno število trojic enako $T = \binom{n}{3}$ in so zgornje skupine razbitje množice vseh trojic, je zamisel postopka naslednja:

- preštej vse dvočlene T_2 in vse povezane T_3 trojice z njihovimi podvrstami;
- število ničelnih trojic izračunaj po zvezi $T_1 = T - T_2 - T_3$.

... Trojiški spekter

V izvedbi postopka moramo zagotoviti, da bo vsaka neničelna trojica šteta natanko enkrat pri prehodu množice povezav. Vozlišča trojice $\{v, u, w\}$ lahko v splošnem izberemo na 6 načinov (v, u, w), (v, w, u), (u, v, w), (u, w, v), (w, v, u), (w, u, v). Problem izomorfizma lahko rešimo z uvedbo *kanonskih* izbir, ki prispevajo k štetju; ostale, nekanonske, lahko preskočimo.

Celotna zahtevnost tega algoritma je reda $O(\hat{\Delta} m)$ in potemtakem, za grafe z majhno največjo stopnjo $\hat{\Delta} \ll n$, ker je $2m \leq n\hat{\Delta}$, reda $O(n)$.

Trojice

Triad:	BA	CL	RC	R2C	TR	HC	39+	p1	p2	p3	p4
003		✓	✓		✓	✓				✓	✓
012					✓	✓	✓			✓	✓
102	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
021D			✓	✓	✓	✓	✓				✓
021U			✓	✓	✓	✓	✓			✓	✓
021C									✓		✓
111D											✓
111U								✓	✓		
030T			✓	✓	✓	✓	✓		✓		✓
030C								✓	✓		✓
201											
120D			✓	✓	✓	✓	✓			✓	✓
120U			✓	✓	✓	✓	✓	✓	✓		✓
120C							✓		✓		
210						✓	✓		✓		
300	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Triad Micro-Models:

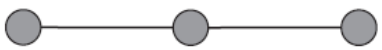
BA: Ballance (Cartwright and Harary, '56)
CL: Clustering Model (Davis. '67)
RC: Ranked Cluster (Davis & Leinhardt, '72)
R2C: Ranked 2-Clusters (Johnsen, '85)
TR: Transitivity (Davis and Leinhardt, '71)
HC: Hierarchical Cliques (Johnsen, '85)
39+: Model that fits D&L's 742 mats N :39-72
p1-p4: Johnsen, 1986. Process Agreement Models.

Moody

Matrično preštevanje vzorcev

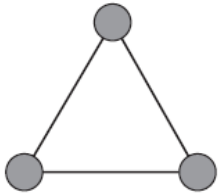
V knjigi Estrada: *A First Course in Network Theory* (2015) je podan dolg seznam obrazcev za preštevanje malih podgrafov.

$F1$



$$|F_1| = \frac{1}{2}\sum_i k_i(k_i-1)$$

$F2$

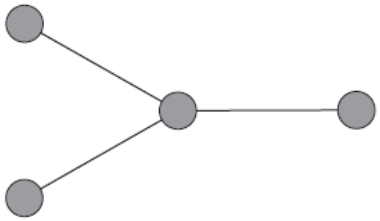


$$|F_2| = \frac{1}{6}\mathrm{tr}(A^3)$$

$F3$

$$|F_3| = \sum_{(i,j)\in E}(k_i-1)(k_j-1) - 3|F_2|$$

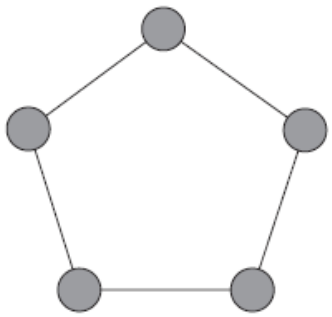
$F4$



$$|F_4| = \frac{1}{6}\sum_i k_i(k_i-1)(k_i-2)$$

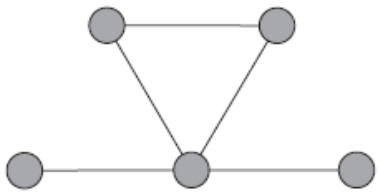
Fiziki stopnjo vozlišča v_i označujejo s k_i.

Matrično preštevanje vzorcev



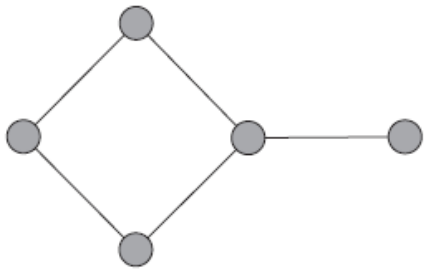
$F8$

$$|F_8| = \frac{1}{10}\left(\mathrm{tr}(A^5) - 30|F_2| - 10|F_6|\right)$$



$F9$

$$|F_9| = \frac{1}{2}\sum_{k_i \geq 4} t_i(k_i-2)(k_i-3)$$

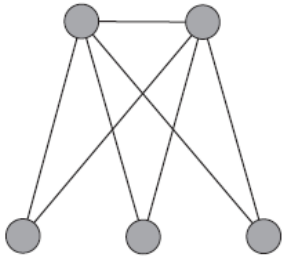


$F10$

$$|F_{10}| = \frac{1}{2}\sum_i (k_i-2) \times \sum_{i,j}\binom{(A^2)_{ij}}{2} - 2|F_7|$$

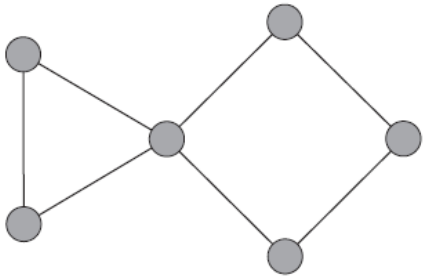
Matrično preštevanje vzorcev

$F17$



$$|F_{17}| = \sum_{(i,j)\in E} \binom{(A^2)_{ij}}{3}$$

$F18$



$$|F_{18}| = \sum_i t_i \cdot \sum_{i\neq j} \binom{(A^2)_{ij}}{2} - 6|F_7| - 2|F_{14}| - 6|F_{17}|$$

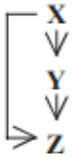
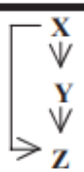
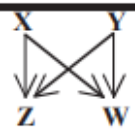
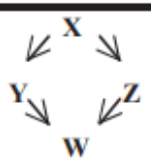
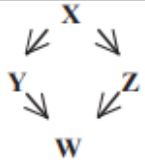
Motivi

Omrežni motivi so podomrežja, ki se v danem omrežju pojavljajo pogosteje kot je pričakovano. Domnevamo lahko, da odražajo lokalno organiziranost v omrežju, ki omogoča njegovo učinkovitejše delovanje.

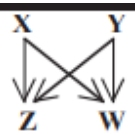
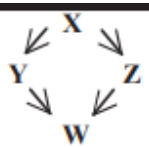
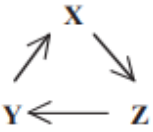
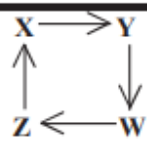
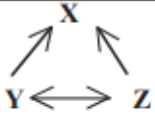
Milo, R, Shen-Orr, S, Itzkovitz, S, Kashtan, N, Chklovskii, D, Alon, U: Network Motifs: Simple Building Blocks of Complex Networks. Science, 298, October 2002, p. 824-827.

Wikipedia: Motifs

Motivi

Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)					**Feed-forward loop**			**Bi-fan**			
E. coli	424	519	40	7 ± 3	10	203	47 ± 12	13			
*S. cerevisiae**	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
C. elegans†	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs					**Three chain**			**Bi-parallel**			
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			

Motivi

Electronic circuits (forward logic chips)					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)					**Three-node feedback loop**			**Bi-fan**			**Four-node feedback loop**
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web					**Feedback with two mutual dyads**			**Fully connected triad**			**Uplinked mutual dyad**
nd.edu§	325,729	1.46e6	1.1e5	$2e3 \pm 1e2$	800	6.8e6	$5e4 \pm 4e2$	15,000	1.2e6	$1e4 \pm 2e2$	5000

R: igraph::motifs

Grafki

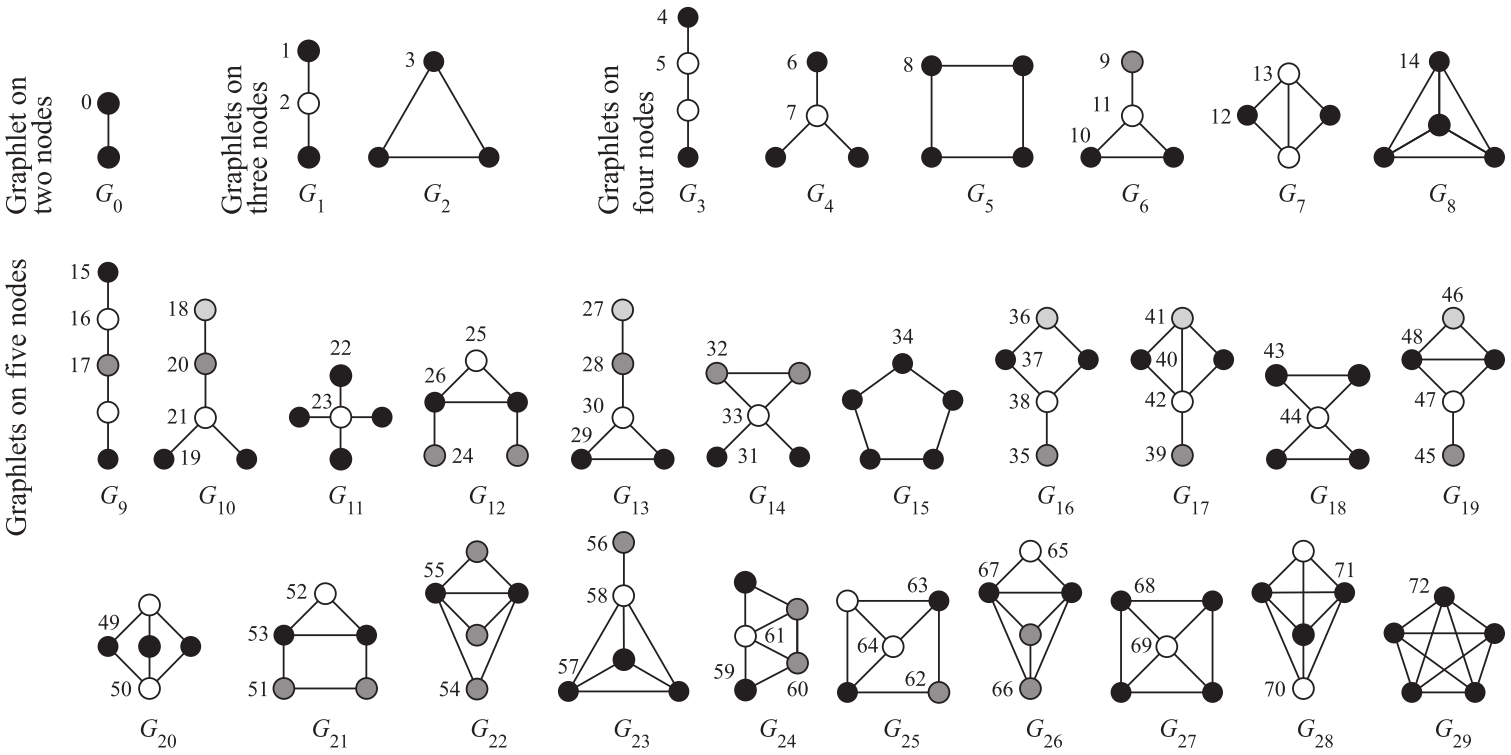
Grafki (graphlets) so majhni neizomorfni porojeni podgrafi danega omrežja. Grafki se razlikujejo od motivov, ker so porojeni podgrafi, motivi pa splošni podgrafi.

Grafke je vpeljala Pržulj Nataša: Biological network comparison using graphlet degree distribution. Bioinformatics, Volume 23, Issue 2, 15 January 2007, Pages e177–e183, PDF

Wikipedia: /Graphlets

iGraph: graphlet; ORCA

Grafki na 2–5 vozliščih z orbitami avtomorfizmov



Vozlišča iste barve pripadajo isti orbiti v grafku.

Dvovrstna omrežja

IMDB

Množenje omrežij

Evropski projekti

Bibliografska omrežja

Analiza omrežij

8. Dvovrstna omrežja in množenje omrežij

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, maj 2024

Kazalo

1 Dvovrstna omrežja
2 IMDB
3 Množenje omrežij
4 Evropski projekti
5 Bibliografska omrežja

Names of Participants of Group I	Code Numbers and Dates of Social Events Reported in *Old City Herald*													
	(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 3/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3
1. Mrs. Evelyn Jefferson	×	×	×	×	×	×		×	×					
2. Miss Laura Mandeville	×	×	×		×	×	×	×						
3. Miss Theresa Anderson		×	×	×	×	×	×	×	×					
4. Miss Brenda Rogers	×		×	×	×	×	×	×						
5. Miss Charlotte McDowd			×	×	×		×							
6. Miss Frances Anderson			×		×	×		×						
7. Miss Eleanor Nye					×	×	×	×						
8. Miss Pearl Oglethorpe						×		×	×					
9. Miss Ruth DeSand					×		×	×	×					
10. Miss Verne Sanderson							×	×	×			×		
11. Miss Myra Liddell								×	×	×		×		
12. Miss Katherine Rogers								×	×	×		×	×	×
13. Mrs. Sylvia Avondale							×	×	×	×		×	×	×
14. Mrs. Nora Fayette						×	×		×	×	×	×	×	×
15. Mrs. Helen Lloyd							×	×		×	×	×		
16. Mrs. Dorothy Murchison								×	×					
17. Mrs. Olivia Carleton									×		×			
18. Mrs. Flora Price									×		×			

prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

21. maj 2024 ob 01 : 39/ maj 2013

Dvovrstna omrežja

Dvovrstno omrežje je omrežje $\mathcal{N} = (\mathcal{U}, \mathcal{V}, \mathcal{A}, w)$, v katerem je množica vozlišč sestavljena iz dveh ločenih podmnožic $\mathcal{U}$ in $\mathcal{V}$ ter imajo povezave iz množice $\mathcal{A}$ svoj začetek v množici $\mathcal{U}$ in svoj konec v množici $\mathcal{V}$. $w : \mathcal{A} \to \mathbb{R}$ je utež povezav. Če utež ni podana, postavimo $w(u, v) = 1$ za vse povezave $(u, v) \in \mathcal{A}$. Množica povezav $\mathcal{A}$ določa relacijo $\mathcal{A} \subseteq \mathcal{U} \times \mathcal{V}$.
Dvovrstno omrežje lahko predstavimo tudi s pravokotno matriko $\mathbf{A} = [a_{uv}]_{\mathcal{U} \times \mathcal{V}}$.

$$a_{uv} = \begin{cases} w(u, v) & (u, v) \in \mathcal{A} \\ 0 & sicer \end{cases}$$

Primeri: (osebe, društva, leta članstva),
(kupci, dobrine, vrednost),
(poslanci, zakoni, glasoval za),
(osebe, časopisi, bere),
(članki, gesla, je opisan z), itd.

Pristopi k analizi dvovrstnih omrežij

Za analizo dvovrstnih omrežij lahko prilagodimo Kleinbergov postopek (hubs and authorities) z *lastnimi vektorji*. Vpeljemo vektorja pomembnosti $(\mathbf{x}, \mathbf{y})$ na $\mathcal{U} \cup \mathcal{V}$ določena z zvezama $\mathbf{y} = \mathbf{A}\mathbf{x}$ in $\mathbf{x} = \mathbf{A}^{\mathsf{T}}\mathbf{y}$.

`Network/2-Mode Network/Important Vertices`

Novejša pristopa sta: *dvovrstne sredice* in *4-obroči*. V naslednjem predavanju bomo spoznali še uporabo *razvrščanja* in *bločnih modelov* pri analizi dvovrstnih omrežij.

Internet Movie Database http://www.imdb.com/

Omrežje IMDB je bilo pripravljeno za 12th Annual Graph Drawing Contest, 2005.
Je dvovrstno in ima $1324748 = 428440 + 896308$ vozlišč ter 3792390 povezav.

Dvovrstne sredice

Podmnožica vozlišč $C \subseteq \mathcal{V}$ je (p, q)-*sredica* dvovrstnega omrežja $\mathcal{N} = (\mathcal{V}_1, \mathcal{V}_2; \mathcal{L})$, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ natanko takrat, ko

a. v s C porojenem podomrežju $\mathcal{K} = (C_1, C_2; \mathcal{L}(C))$, $C_1 = C \cap \mathcal{V}_1$, $C_2 = C \cap \mathcal{V}_2$ velja $\forall v \in C_1 : \deg_K(v) \geq p$ in $\forall v \in C_2 : \deg_K(v) \geq q$;

b. C je maksimalna podmnožica $\mathcal{V}$, ki zadošča pogoju **a**.

Lastnosti dvovrstnih sredic:

- $C(0, 0) = \mathcal{V}$
- $\mathcal{K}(p, q)$ ni vedno povezano
- $(p_1 \leq p_2) \wedge (q_1 \leq q_2) \Rightarrow C(p_1, q_1) \subseteq C(p_2, q_2)$
- $\mathcal{C} = \{C(p, q) : p, q \in \mathbb{N}\}$. Če so vse neprazne podmnožice iz $\mathcal{C}$ različne, je mreža.

Postopek za določitev dvovrstnih sredic

Za določitev (p, q)-sredice lahko uporabimo postopek podoben postopku za določanje običajnih sredic:

repeat
iz množice $\mathcal{V}_1$ odstrani vsa vozlišča s trenutno stopnjo manjšo od p, i
iz množice $\mathcal{V}_2$ odstrani vsa vozlišča s trenutno stopnjo manjšo od
until nobeno vozlišče ni bilo odstranjeno;

Postopek je mogoče sprogramirati tako, da ima zahtevnost $O(m)$. Težava je v tem, da je lahko veliko dvovrstnih sredic. Kako izbrati zanimive? Pomagamo si lahko s tabelo značilnosti sredic $n_1 = |C_1(p, q)|$, $n_2 = |C_2(p, q)|$ in k – je število komponent v $\mathcal{K}(p, q)$:

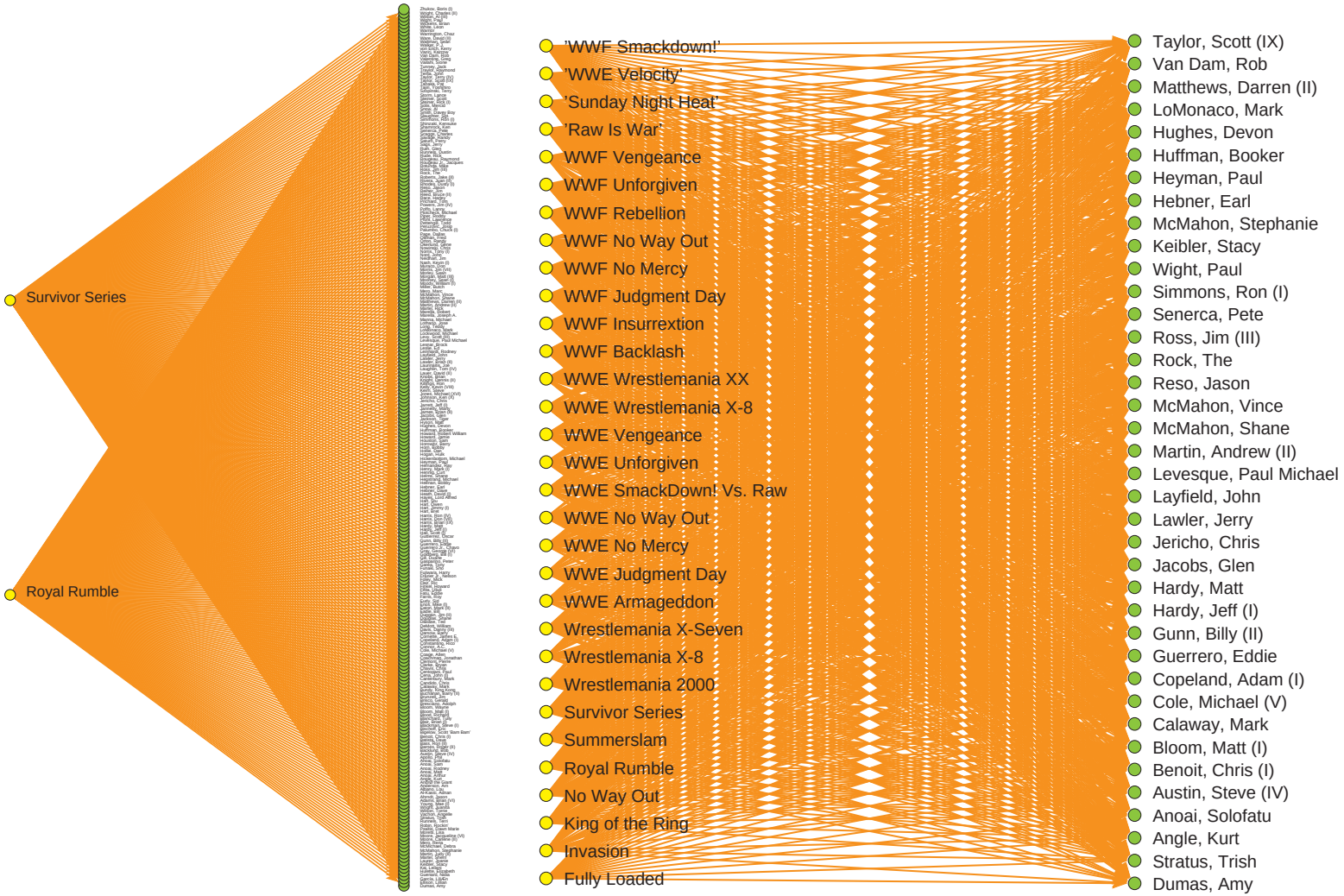
- $n_1 + n_2 \leq$ izbrano zgornjo mejo
- večji skoki z $C(p-1, q)$ ali $C(p, q-1)$ na $C(p, q)$.

`Network/2-Mode Network/Core/2-Mode Border`

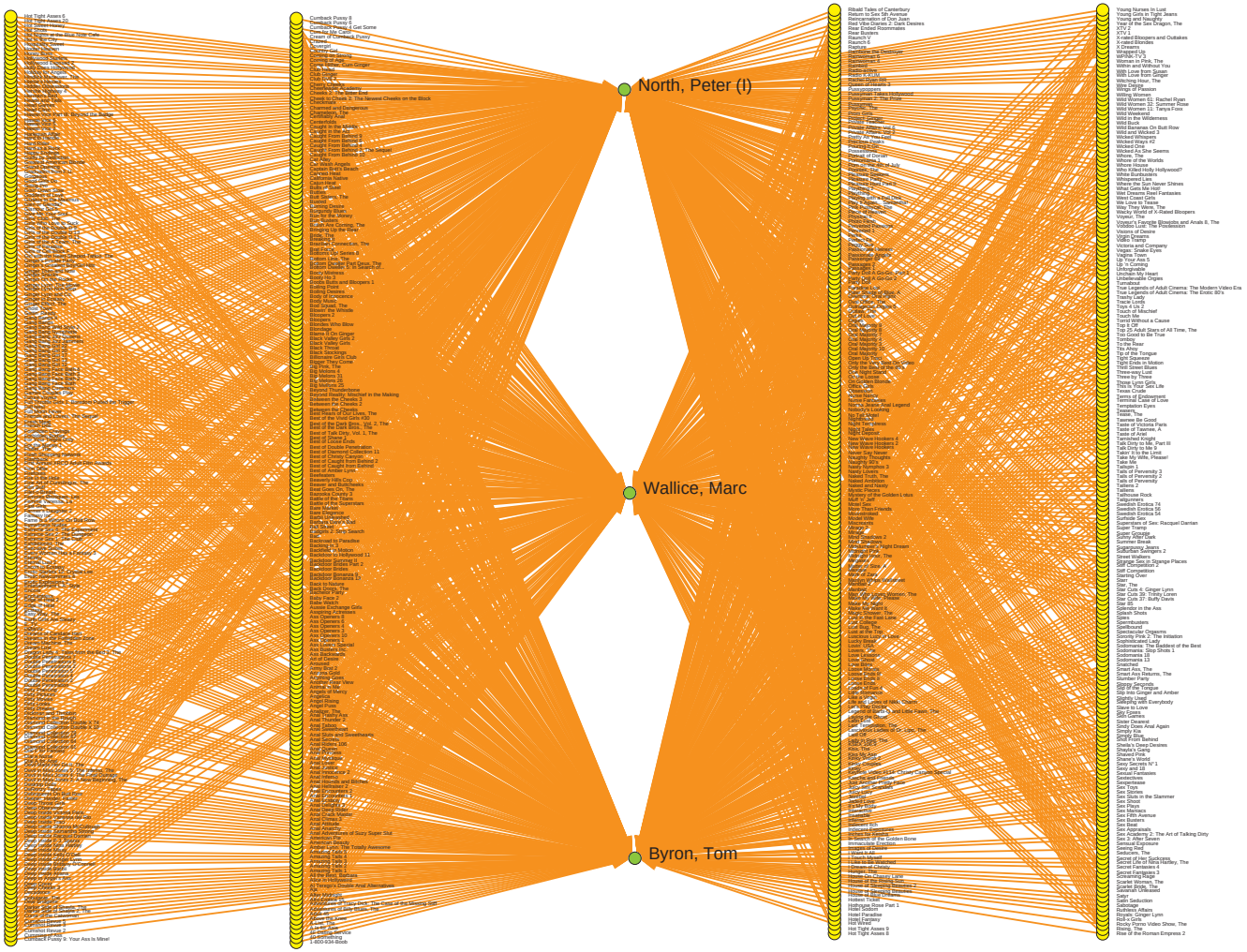
Tabela $(p, q : n_1, n_2)$ za Internet Movie Database

1	1590:	1590	1	22	24:	1854	1153	43	14:	29	83
2	516:	788	3	23	23:	47	56	44	14:	29	83
3	212:	1705	18	24	23:	34	39	45	13:	30	95
4	151:	4330	154	25	22:	42	53	46	13:	29	94
5	131:	4282	209	26	22:	31	38	47	12:	29	101
6	115:	3635	223	27	22:	31	38	48	12:	28	100
7	101:	3224	244	28	20:	36	53	49	12:	26	95
8	88:	2860	263	29	20:	35	52	50	11:	27	111
9	77:	3467	393	30	19:	35	59	51	11:	26	110
10	69:	3150	428	31	19:	35	59	52	11:	16	79
11	63:	2442	382	32	19:	34	57	53	10:	35	162
12	56:	2479	454	33	18:	34	62	54	10:	35	162
13	50:	3330	716	34	18:	34	62	55	10:	34	162
14	46:	2460	596	35	18:	33	61	56	10:	34	162
15	42:	2663	739	36	17:	33	65	57	9:	35	187
16	39:	2173	678	37	16:	33	75	58	9:	33	180
17	35:	2791	995	38	16:	30	73	59	9:	33	180
18	32:	2684	1080	39	16:	29	70	60	9:	32	178
19	30:	2395	1063	40	15:	29	77	61	9:	31	177
20	28:	2216	1087	41	15:	28	76	62	9:	31	177
21	26:	1988	1087	42	15:	28	76	63	8:	31	202

(247,2)-sredica in (27,22)-sredica

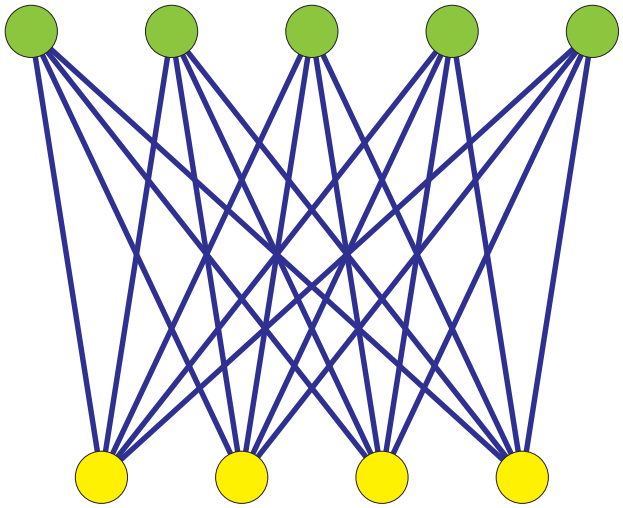


(2,516)-Hard core



4-obroči in analiza dvovrstnih omrežij

V dvovrstnih omrežjih ni 3-obročev. Najkrajše sklenjene verige imajo dolžino 4 – so 4-obroči. Najgostejše podstrukture v dvovrstnih omrežjih so polni dvodelni podgrafi $K_{p,q}$. Ti vsebujejo veliko 4-obročev.



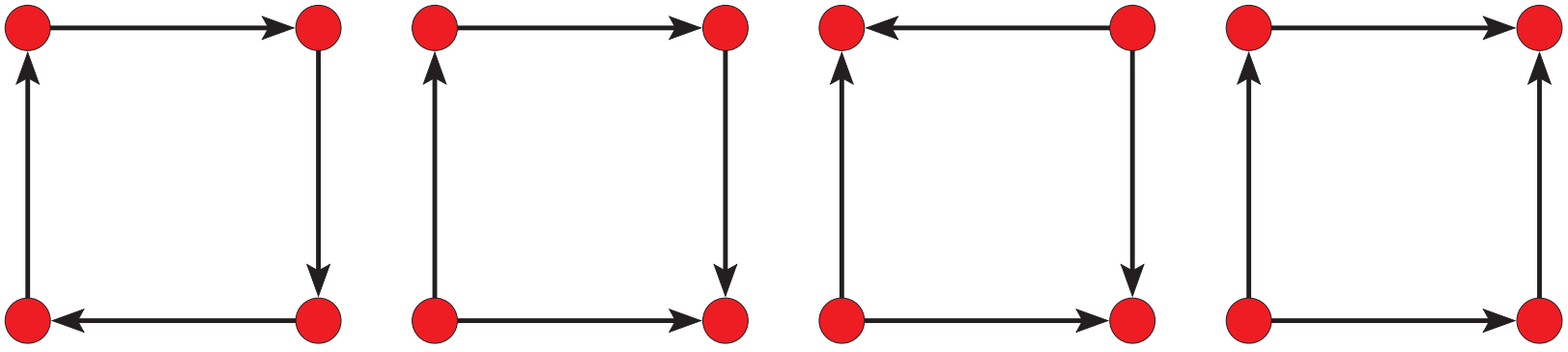
$$w_4(K_{p,q}) = (p-1)(q-1)$$

Določanje 4-obročnih uteži je bilo vključeno v program Pajek avgusta 2005.

`Network/Create New Network/with Ring Counts.../4-Rings/Undirected`

4-obroči v usmerjenih omrežjih

Obstajajo 4 vrste usmerjenih 4-obročev:



ciklični tranzitivni rodovniški karo

Za tranzitivne obroče `Pajek` določi še posebno utež, ki pove, kolikokrat je povezava *bližnjica* v tranzitivnem 4-obroču.

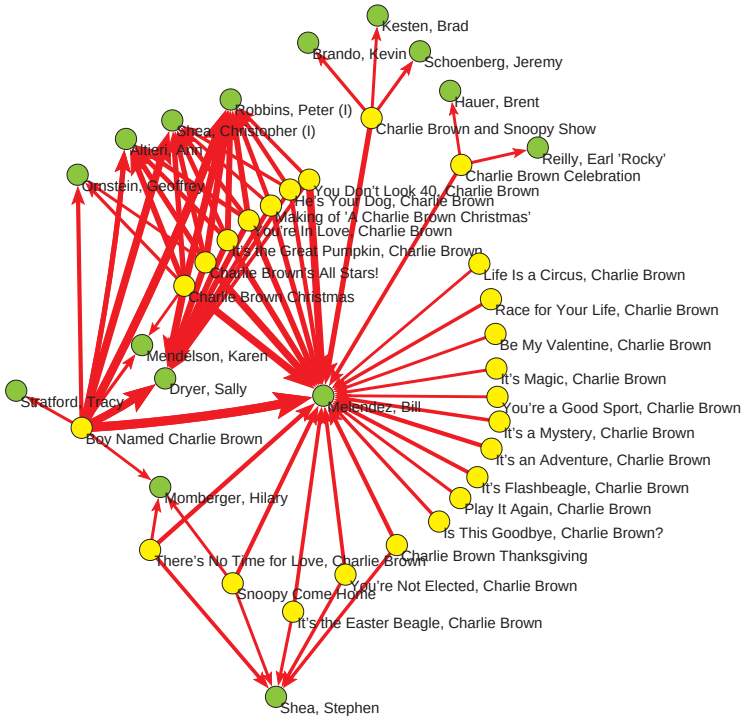
`Network/Create New Network/with Ring Counts.../4-Rings/Directed`

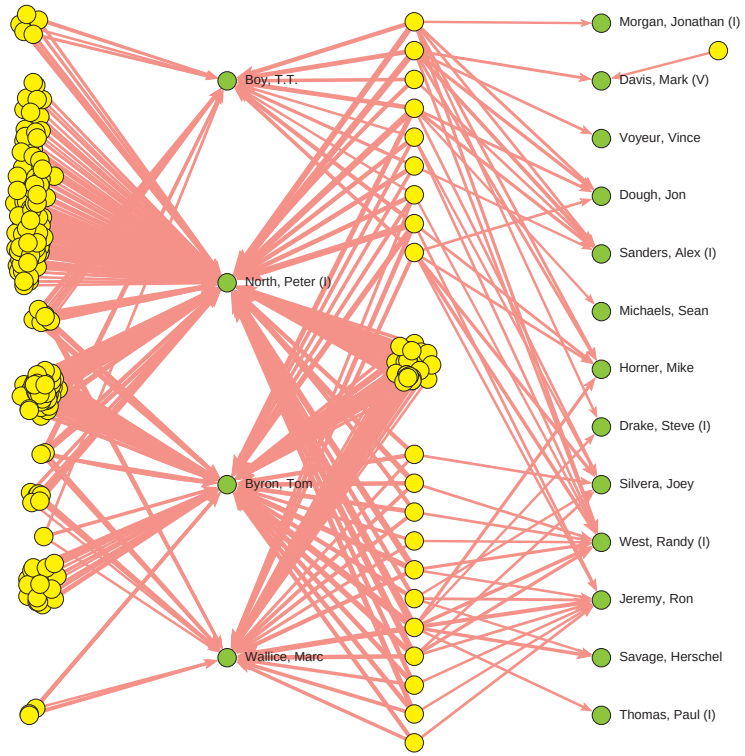
Enostavni povezavni otoki v IMDB za w_4

Dobili smo 12465 enostavnih povezavnih otokov na 56086 vozliščih. Tu je porazdelitev njihovih velikosti.

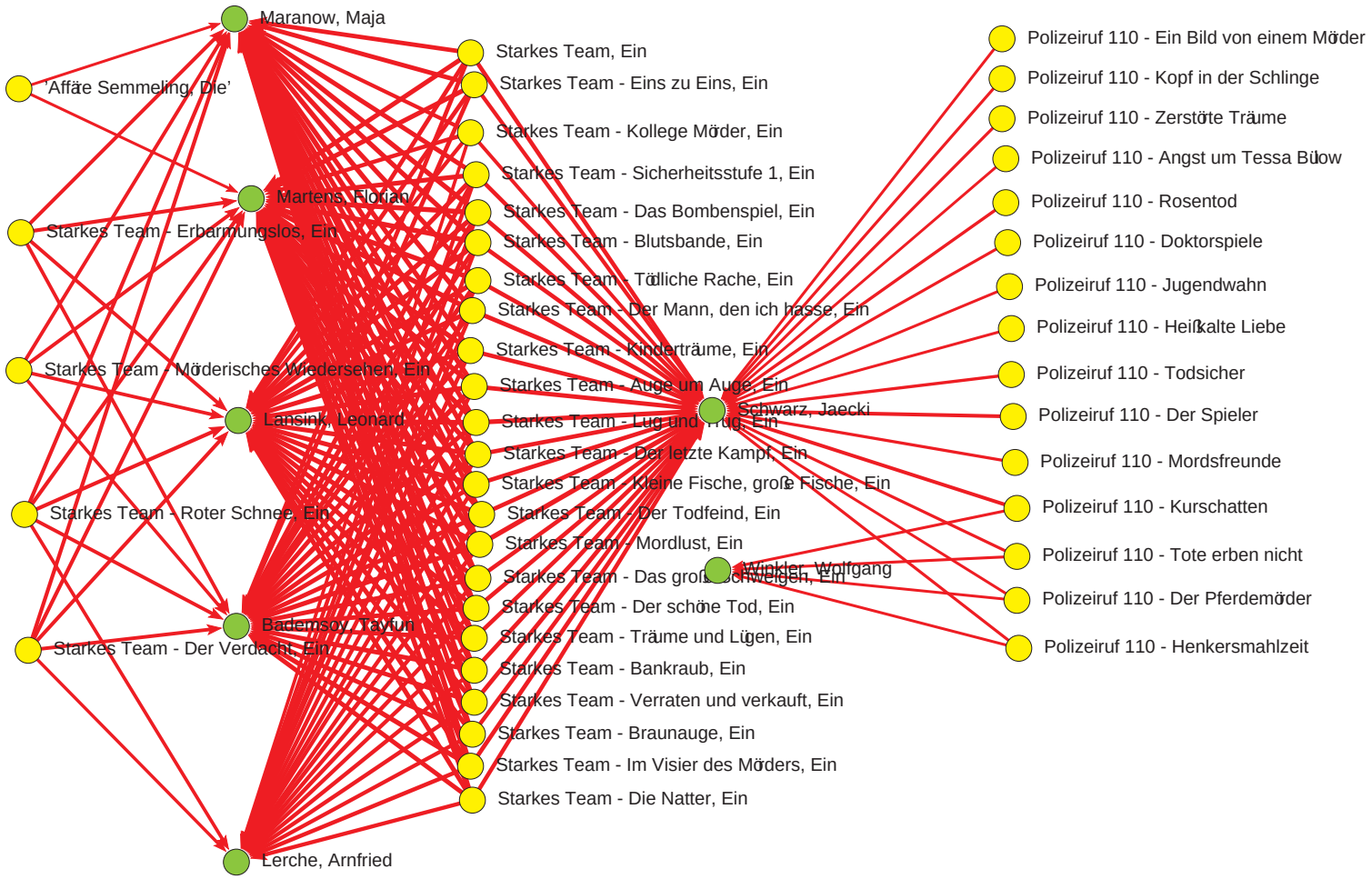
Size	Freq	Size	Freq	Size	Freq	Size	Freq
2	5512	20	19	38	4	59	2
3	1978	21	18	39	3	61	1
4	1639	22	15	40	2	64	1
5	968	23	9	42	2	67	1
6	666	24	13	43	3	70	1
7	394	25	12	45	3	73	1
8	257	26	6	46	4	76	1
9	209	27	6	47	5	82	1
10	148	28	5	48	1	86	1
11	118	29	6	49	2	106	1
12	87	30	3	50	2	122	1
13	55	31	6	51	1	135	1
14	62	32	5	52	2	144	1
15	46	33	3	53	1	163	1
16	39	34	1	54	2	269	1
17	27	35	5	55	1	301	1
18	28	36	4	57	1	332	2
19	29	37	7	58	1	673	1

Primer: Otoka za w_4 / Charlie Brown in Adult



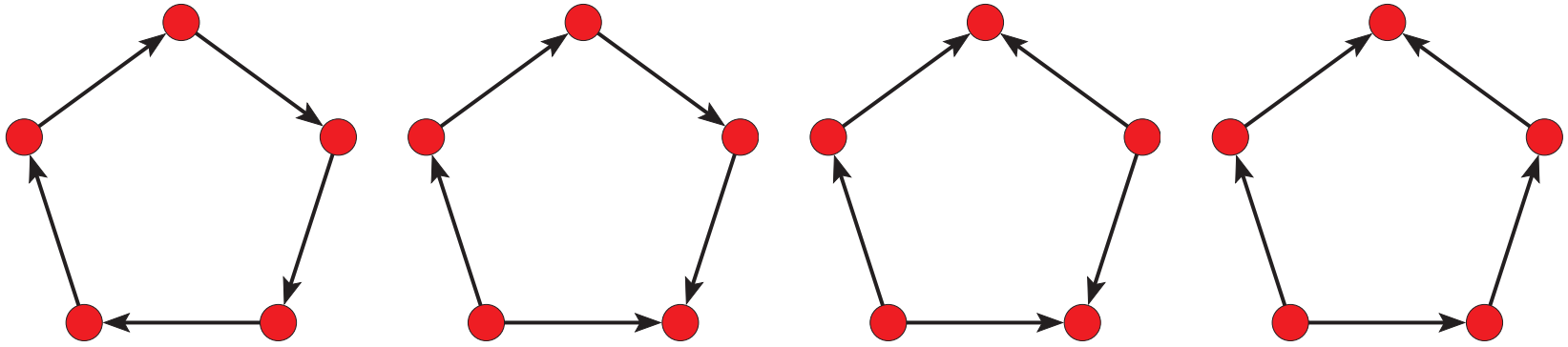


Primer: Otok za w_4 / Polizeiruf 110 in Starkes Team



5-obroči

Najbrž bodo v `Pajek` vključene tudi uteži w_5. Zanimivo je, da tudi v tem primeru obstajajo le 4 vrste usmerjenih 5-obročev.



ciklični tranzitivni ???? ????

Množenje omrežij

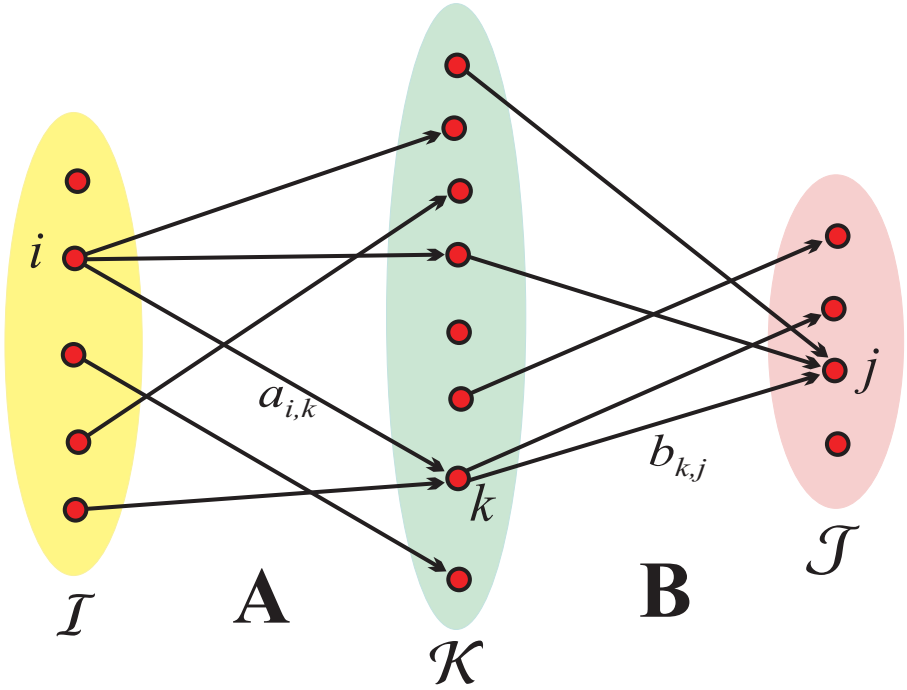
Naj bo $\mathcal{N} = (\mathcal{I}, \mathcal{J}, \mathcal{E}, w)$ enostavno dvovrstno omrežje nad množicama vozlišč $\mathcal{I}$ in $\mathcal{J}$ ter množico povezav $\mathcal{E}$, ki imajo krajišča v obeh množicah $\mathcal{I}$ in $\mathcal{J}$. $w : \mathcal{E} \to \mathbb{R}$ (ali kak drug polkolobar) je utež. Omrežju priredimo matriko omrežja $\mathbf{W} = [w_{i,j}]$ z vrednostmi: $w_{i,j} = w(i,j)$ za $(i,j) \in \mathcal{E}$ in $w_{i,j} = 0$ sicer.

Imejmo usklajeni omrežji $\mathcal{N}_A = (\mathcal{I}, \mathcal{K}, \mathcal{E}_A, w_A)$ in $\mathcal{N}_B = (\mathcal{K}, \mathcal{J}, \mathcal{E}_B, w_B)$ s pripadajočima matrikama $\mathbf{A}_{\mathcal{I}\times\mathcal{K}}$ in $\mathbf{B}_{\mathcal{K}\times\mathcal{J}}$. *Produkt omrežij* $\mathcal{N}_A$ in $\mathcal{N}_B$ imenujemo omrežje $\mathcal{N}_C = (\mathcal{I}, \mathcal{J}, \mathcal{E}_C, w_C)$, kjer so $\mathcal{E}_C = \{(i,j) : i \in \mathcal{I}, j \in \mathcal{J}, c_{i,j} \neq 0\}$ in $w_C(i,j) = c_{i,j}$ za $(i,j) \in \mathcal{E}_C$. Matrika produkta $\mathbf{C} = [c_{i,j}]_{\mathcal{I}\times\mathcal{J}} = \mathbf{A} * \mathbf{B}$ je določena na običajen način

$$c_{i,j} = \sum_{k \in \mathcal{K}} a_{i,k} \cdot b_{k,j}$$

V primeru, ko so $\mathcal{I} = \mathcal{K} = \mathcal{J}$ imamo opravka z navadnimi enovrstnimi omrežji (s kvadratnimi matrikami).

Množenje omrežij



$$c_{i,j} = \sum_{k \in N_A(i) \cap N_B^-(j)} a_{i,k} \cdot b_{k,j}$$

Če so vse uteži v omrežjih $\mathcal{N}_A$ in $\mathcal{N}_B$ enake 1, vrednost $c_{i,j}$ šteje na koliko načinov lahko pridemo iz vozlišča $i \in \mathcal{I}$ v vozlišče $j \in \mathcal{J}$ čez neko vozlišče iz $\mathcal{K}$.

Običajno množenje matrik

Običajni postopek za izračun produkta matrik **C**

$$c_{i,j} = \sum_{k \in \mathcal{K}} a_{i,k} \cdot b_{k,j}$$

je naslednji

for i **in** $\mathcal{I}$ **do**
  for j **in** $\mathcal{J}$ **do begin**
    $s := 0;$
    for k **in** $\mathcal{K}$ **do** $s := s + a_{i,k} * b_{k,j};$
    $c_{i,j} := s;$
  end;

Njegova zahtevnost je reda $O(|\mathcal{I}| \cdot |\mathcal{K}| \cdot |\mathcal{J}|)$ – in je zato prepočasno za uporabo na velikih omrežjih.

Hitro množenje redkih matrik

Za velika redka omrežja lahko izračunamo produkt veliko hitreje, če upoštevamo le neničelne vrednosti:

for k **in** $\mathcal{K}$ **do**
  for i **in** $N_A^-(k)$ **do**
    for j **in** $N_B(k)$ **do**
      if $\exists c_{i,j}$ **then** $c_{i,j} := c_{i,j} + a_{i,k} * b_{k,j}$
      else new $c_{i,j} := a_{i,k} * b_{k,j}$

$N_A^-(k)$: sosedi vozlišča k v omrežju $\mathcal{N}_A$
$N_B(k)$: sosedi vozlišča k v omrežju $\mathcal{N}_B$

V splošnem je množenje velikih redkih omrežij nevarna operacija, ker se lahko izid 'razpoči' – ni redek.

`Networks/Multiply Networks`

Zahtevnost hitrega množenja

Iz postopka množenja omrežij vidimo, da vsako vmesno vozlišče $k \in \mathcal{K}$ doda produktu polno dvovrstno podomrežje $K_{N_A^-(k),N_B(k)}$ (ali, v primeru $\mathcal{I} = \mathcal{J}$, polno podomrežje $K_{N(k)}$). Če sta obe stopnji $\deg_A(k) = |N_A^-(k)|$ in $\deg_B(k) = |N_B(k)|$ veliki, potem je že izračun tega podomrežja kvadratičen (prostor in čas) – izid se razpoči.

Enostavno je videti, da če ima vsaj eno od redkih omrežij $\mathcal{N}_A$ in $\mathcal{N}_B$ majhno največjo stopnjo Δ na vmesni množici $\mathcal{K}$, potem je tudi produktno omrežje $\mathcal{N}_C$ redko.

Pokazali bomo močnejše trditev: Če za redki omrežji $\mathcal{N}_A$ in $\mathcal{N}_B$ obstaja v $\mathcal{K}$ le nekaj vozlišč velike stopnje in ni nobeno tako, da bi imelo veliko stopnjo v obeh omrežjih, potem je tudi produktno omrežje $\mathcal{N}_C$ redko.

Podrobnejša analiza zahtevnosti

Bodita **A** in **B** matriki omrežij $\mathcal{N}_A = (\mathcal{I}, \mathcal{K}, \mathcal{E}_A, w_A)$ in $\mathcal{N}_B = (\mathcal{K}, \mathcal{J}, \mathcal{E}_B, w_B)$. Privzemimo, da lahko telo zank izračunamo v konstantnem času c. Označimo $d_{min}(k) = \min(\deg_A(k), \deg_B(k))$, $\Delta_{min} = \max_{k \in \mathcal{K}} d_{min}(k)$, $d_{max}(k) = \max(\deg_A(k), \deg_B(k))$, $\mathcal{K}(d) = \{k \in \mathcal{K} : d_{max}(k) \geq d\}$, $d^* = \mathrm{argmin}_d(|\mathcal{K}(d)| \leq d)$ in $\mathcal{K}^* = \mathcal{K}(d^*)$. Velja $|\mathcal{K}^*| \leq d^*$. Tedaj je zahtevnost izračuna produkta enaka

$$C = \sum_{k \in \mathcal{K}} \sum_{i \in N_A(k)} \sum_{j \in N_B(k)} c = c \cdot \sum_{k \in \mathcal{K}} \deg_A(k) \cdot \deg_B(k) = c \cdot \sum_{k \in \mathcal{K}} d_{min}(k) \cdot d_{max}(k)$$

$$= c \cdot \Big(\sum_{k \in \mathcal{K}^*} d_{min}(k) \cdot d_{max}(k) + \sum_{k \in \mathcal{K} \setminus \mathcal{K}^*} d_{min}(k) \cdot d_{max}(k) \Big)$$

$$\leq c \cdot \Big(\Delta_{min} \cdot \sum_{k \in \mathcal{K}^*} d_{max}(k) + d^* \cdot \sum_{k \in \mathcal{K} \setminus \mathcal{K}^*} d_{min}(k) \Big)$$

$$\leq c \cdot d^* \cdot (\Delta_{min} \cdot \max(|\mathcal{I}|, |\mathcal{J}|) + \min(|\mathcal{E}_A|, |\mathcal{E}_B|))$$

Če sta za omrežji $\mathcal{N}_A$ in $\mathcal{N}_B$ količini Δ_{min} in d^* majhni, tedaj je tudi produktno omrežje $\mathcal{N}_C$ redko.

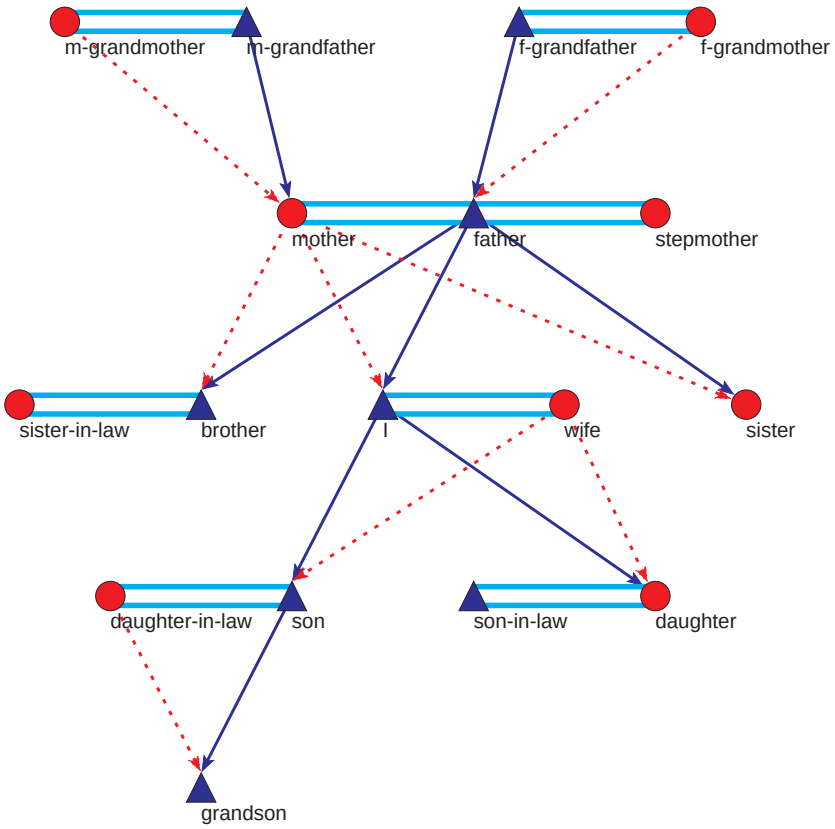
Primer: Sorodstvene vezi

V antropologiji štejejo za osnovne sorodstvene relacije:

Kin Type	English Type	Slovensko
P	Parent	starši
F	Father	oče
M	Mother	mati
C	Child	otrok
D	Daughter	hči
S	Son	sin
G	Sibling	brat|sestra
Z	Sister	sestra
B	Brother	brat
E	Spouse	zakonec
W	Wife	žena
H	Husband	mož

Rodovniki so običajno opisani v obliki GEDCOM. Primeri family, Bouchards.

Orejev graf



V Orejevem grafu je vsaka oseba predstavljena z vozliščem, poroke, odnos

_ is a spouse of _ ,

je predstavljen z neusmerjeno povezavo, odnosa

_ is a mother of _

in

_ is a father of _

pa z usmerjeno povezavo s starša na otroka.

Izračuni sorodstvenih vezi

Ko prebere rodovnik kot Orejev graf, `Pajek` ustvari tri relacije:

F: _ **is a father of** _
M: _ **is a mother of** _
E: _ **is a spouse of** _

Za nadaljnje potrebe moramo ustvariti še dve (diagonalni) relaciji, ki nam omogočata razlikovanje med moškimi in ženskami:

L: _ **is a male** _ / 1-moški, 0-ženska
J: _ **is a female** _ / 1-ženska, 0-moški

$$\mathbf{F} \cap \mathbf{M} = \emptyset, \qquad \mathbf{L} \cup \mathbf{J} \subseteq \mathbf{I}, \qquad \mathbf{L} \cap \mathbf{J} = \emptyset$$

Izpeljane sorodstvene relacije

Druge osnovne sorodstvene relacije lahko določimo z uporabo makrojev, ki temeljijo na naslednjih zvezah:

_ **is a parent of** _	P	$=$	$F \cup M$
_ **is a child of** _	C	$=$	P^T
_ **is a son of** _	S	$=$	$L * C$
_ **is a daughter of** _	D	$=$	$J * C$
_ **is a husband of** _	H	$=$	$L * E$
_ **is a wife of** _	W	$=$	$J * E$
_ **is a sibling of** _	G	$=$	$((F^T * F) \cap (M^T * M)) \setminus I$
_ **is a brother of** _	B	$=$	$L * G$
_ **is a sister of** _	Z	$=$	$J * G$
_ **is an uncle of** _	U	$=$	$B * P$
_ **is an aunt of** _	A	$=$	$Z * P$
_ **is a semi-sibling of** _	G_e	$=$	$(P^T * P) \setminus I$

Z njihovo uporabo lahko določimo še vrsto drugih relacij:

_ **is a grand mother of** _	M_2	$=$	$M * P$
_ **is a niece of** _	Ni	$=$	$D * G$

Razmerja med velikostmi sorodstvenih relacij v rodovnikih

Kin Type	Turks	Ragusa	Loka	Silba	Royal
P-Parent	1.000	1.000	1.000	1.000	1.000
F-Father	0.514	0.532	0.504	0.519	0.540
M-Mother	0.486	0.468	0.496	0.481	0.460
C-Child	1.000	1.000	1.000	1.000	1.000
D-Daughter	0.431	0.384	0.480	0.469	0.427
S-Son	0.569	0.616	0.520	0.531	0.573
G-Sibling	1.250	0.943	1.019	0.811	0.767
Z-Sister	1.135	0.746	0.983	0.760	0.707
B-Brother	1.366	1.140	1.055	0.861	0.828
E-Spouse	0.205	0.215	0.208	0.230	0.306
H-Husband	0.205	0.215	0.208	0.230	0.306
W-Wife	0.205	0.215	0.208	0.230	0.306
U-Uncle	1.920	1.789	1.200	1.181	0.927
A-Aunt	1.750	1.143	1.190	1.097	0.798
Ge-Semi-sibling	1.473	1.155	1.128	0.932	0.905
n	1269	5999	47956	6427	3010
mE = Spouse	407	2002	14154	2217	1138
mA = Parent	1987	9315	68052	9627	3724

Omrežja iz podatkovnih tabel in baz

RuthDELmain.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Ident	Num	File	ORGANISATION ORI	ORG	Org	Contact Name	Street	ZIP	Project	City	Country	coun	EU	Region
2	1	1480	613.html	3D PLUS SA	3D F	3D I	LIGNIER, Olivier	641 RU	78530	IST-2001-3440	Buc	FRANCE	20	2	ÎLE DE FF
3	2	1481	613.html	3D PLUS SA	3D PLUS		LIGNIER, Olivier	641 Ru	78530	IST-2001-3440	Buc	FRANCE	20	2	ÎLE DE FF
4	3	4001	924.html	3D VISION	3D V	3D '	MARIAT, Jacques	Savoie	73375	502909	Le Bo	FRANCE	20	2	CENTRE-I
5	4	1648	160.html	3D Web Technologies	3D WEB		DENNISON, Andrew		M31 4XL	BMH4989519	Carrir	UNITED KI	60	2	NORTH W
6	5	1406	442.html	3E	3E		PALMERS, Geert	Eredier	1000	NNE5/51/1999	Bruxe	BELGIQUE	8	2	REG.BRU
7	6	1007	884.html	4M2C PATRIC SALOI	4M2C PA		N/A	CRANA	12157	507255	Berlir	DEUTSCH	15	2	BERLIN E
8	7	7914	991.html	5T S.c.r.l.	5T S.C.R		N/A	C.so B	10126	Road2/506716	Torinc	ITALIA	26	2	NORD OV
9	8	6880	588.html	A & C 2000 S.R.L.	A &	SAI	CARLUCCI, Renz	VIALE	148	IST-2001-3454	Roma	ITALIA	26	2	LAZIO Ro
10	9	6881	588.html	A & C 2000 S.R.L.	A & C 2C		CARLUCCI, Renz	Viale C	148	IST-2001-3454	Roma	ITALIA	26	2	LAZIO Ro
11	10	1647	176.html	A. BENETTI MACCHII	A. BENE		Federico BENETT	Via Pro	54033	BRST985466	Carra	ITALIA	26	2	CENTRO
12	11	6605	984.html	A. Mickiewicz Univers	A. M	Inst	PATKOWSKI, Ad	Ul. H. \	61-712	502235	Pozn	POLSKA	45	2	
13	12	6571	135.html	A.BRITO - INDUSTRIA	A.BRITO		VIEIRA DE BRITC	5109,E	4350-119	BRST985263	Porto	PORTUGA	46	2	CONTINEI
14	13	1813	409.html	A.L. DIGITAL LIMITED	A.L.	A.L.	LAURIE, Ben	VOYSE	W4 4GB	IST-2000-2633	Chisv	UNITED KI	60	2	SOUTH E.
15	14	1814	409.html	A.L. Digital Limited	A.L.	DIG	LAURIE, Ben	Voysey	W4 4GB	IST-2000-2633	Chisv	UNITED KI	60	2	SOUTH E.
16	15	1885	960.html	A.P. MOLLER-MAER	A.P.	TEC	DRAGSTED, Jorr	Esplan	1098	506676	Kope	DANMARK	14	2	København
17	16	6731	537.html	A.S.M. S.A.	A.S.M.	S	MOYA GARCIA,	Carrete	43206	IST-2000-3008	Reus	ESPAÑA	19	2	ESTE CA
18	17	8150	232.html	AABO AKADEMI UNI	AAB	COI	NYBACKA-WILLI	14-18B	20500	ERK5-CT-1999	Turku	SUOMI/FIN	53	2	MANNER-
19	18	8152	662.html	AABO AKADEMI UNI	AAB	DEF	BJORKSTRAND,	3,Tykis	20521	EVK1-CT-2002	Turku	SUOMI/FIN	53	2	
20	19	8148	959.html	AABO AKADEMI UNI	AAB	Dep	HUPA, Mikko	Domky	20500	502679	Turku	SUOMI/FIN	53	2	MANNER-
21	20	8151	233.html	AABO AKADEMI UNI	AAB	DEF	NYBACKA-WILLI	Lemmi	20500	ERK6-CT-1999	Turku	SUOMI/FIN	53	2	MANNER-
22	21	125	116.html	AACHEN UNIVERSIT	AAC	GIE	E. NEUSSL	Intzest	52072	BRPR980663	Aach	DEUTSCH	15	2	NORDRHE
23	22	123	104.html	AACHEN UNIVERSIT	AAC	GIE	MEISER, Lukas	Intzest	52072	BRPR980695	Aach	DEUTSCH	15	2	NORDRHE
24	23	155	364.html	AACHEN UNIVERSIT	AAC	INS	RAUHUT, Burkha	18,Eilfs	52062	G1RD-CT-2000	Aach	DEUTSCH	15	2	NORDRHE

Podatkovna tabela ali baza $\mathcal{T}$ je sestavljena iz množice *zapisov* $\mathcal{T} = \{T_k : k \in \mathcal{K}\}$, kjer je $\mathcal{K}$ množica *ključev*. Posamezni zapis ima obliko $T_k = (k, q_1(k), q_2(k), \ldots, q_r(k))$ kjer je $q_i(k)$ vrednost *lastnosti* $\mathbf{q}_i$ za ključ k.

Zapis z Web of Science

```
PT J
AU Dipple, H
   Evans, B
TI The Leicestershire Huntington's disease support group: a social network
   analysis
SO HEALTH & SOCIAL CARE IN THE COMMUNITY
LA English
DT Article
C1 Rehabil Serv, Troon Way Business Ctr, Leicester LE4 9HA, Leics, England.
RP Dipple, H, Rehabil Serv, Troon Way Business Ctr, Sandringham
   Suite,Humberstone Lane, Leicester LE4 9HA, Leics, England.
CR BORGATTI SP, 1992, UCINET 4 VERSION 1 0
   FOLSTEIN S, 1989, HUNTINGTONS DIS DISO
   SCOTT J, 1991, SOCIAL NETWORK ANAL
NR 3
TC 3
PU BLACKWELL SCIENCE LTD
PI OXFORD
PA P O BOX 88, OSNEY MEAD, OXFORD OX2 ONE, OXON, ENGLAND
SN 0966-0410
J9 HEALTH SOC CARE COMMUNITY
JI Health Soc. Care Community
PD JUL
PY 1998
VL 6
IS 4
BP 286
EP 289
PG 4
SC Public, Environmental & Occupational Health; Social Work
GA 105UP
UT ISI:000075092200008
ER
```

WoS2Pajek

Zapisi BiBTEX

```
@Article{int:Mizuno1,
  author =        "S. Mizuno",
  title =         "An ${O(n^{3}L)}$ algorithm using a sequence for
                  linear complementarity problems",
  journal =       "Journal of the Operations Research Society of Japan",
  volume =        "33",
  year =          "1990",
  pages =         "66--75",
}
@InCollection{int:Vorst1,
  author =        "{J. G. G. van de} Vorst",
  title =         "An attempt to use parallel computing in large scale
                  optimisation",
  booktitle =     "Logistics, Where Ends Have to Meet~: Proceedings of
                  the Shell Conference on Logistics in Apeldoorn, The
                  Netherlands, November 1988",
  editor =        "{C. F. H. van} Rijn",
  year =          "1989",
  pages =         "112--119",
  publisher =     "Pergamon Press",
  address =       "Oxford, United Kingdom",
}
```

Bib2Pajek.py

...Omrežja iz podatkovnih tabel

Naj ima lastnost $\mathbf{q}$ zalogo vrednosti $\mathcal{Q}$. Če je ta končna (to lahko vselej dosežemo z razbitjem na razrede), lahko lastnosti $\mathbf{q}$ priredimo dvovrstno omrežje $\mathcal{K} \times \mathbf{q} = (\mathcal{K}, \mathcal{Q}, \mathcal{E}, w)$ določeno z usmerjenimi povezavami $(k, v) \in \mathcal{E}$ ntk. $q(k) = v$, z utežmi $w(k, v) = 1$.
txt2pajek (Jürgen Pfeffer)

Dvovrstno omrežje $\mathbf{q}_i \times \mathbf{q}_j = (\mathcal{Q}_i, \mathcal{Q}_j, \mathcal{E}, w)$ lahko definiramo tudi za lastnosti $\mathbf{q}_i$ in $\mathbf{q}_j$, kjer je $(u, v) \in \mathcal{E}$ natanko takrat, ko
$\exists k \in \mathcal{K} : (q_i(k) = u \wedge q_j(k) = v)$, in je
$w(u, v) = \mathrm{card}(\{k \in \mathcal{K} : (q_i(k) = u \wedge q_j(k) = v)\})$.

Naj bo še $[\mathbf{q}_i \times \mathbf{q}_j]^T = \mathbf{q}_j \times \mathbf{q}_i$.
Potem velja $\mathbf{q}_i \times \mathbf{q}_j = [\mathcal{K} \times \mathbf{q}_i]^T * [\mathcal{K} \times \mathbf{q}_j] = [\mathbf{q}_i \times \mathcal{K}] * [\mathcal{K} \times \mathbf{q}_j]$.

Par lastnosti $\mathbf{q}_i$ in $\mathbf{q}_j$ lahko združimo tudi glede na neko tretjo lastnost $\mathbf{q}_s$: dobimo dvovrstno omrežje $[\mathbf{q}_i \times \mathbf{q}_j]/\mathbf{q}_s = [\mathbf{q}_i \times \mathbf{q}_s] * [\mathbf{q}_s \times \mathbf{q}_j]$.

Dvovrstna omrežja iz podatkovnih baz

Za podatke iz Web of Science (Knowledge) lahko ustvarimo ustrezna omrežja s programom `WoS2Pajek`:

- omrežje sklicevanj **Ci**: dela $\times$ dela;
- omrežje avtorstev **WA**: dela $\times$ avtorji; za dela brez polnih opisov je znan le prvi avtor;
- omrežje gesel (ključnih besed) **WK**: dela $\times$ gesla (samo za dela s polnimi opisi);
- omrežje revij **WJ**: dela $\times$ revije;
- razbitje del glede na leto objave;
- razbitje del – polni opis (1) / samo ime ISI (0);

Podobni programi obstajajo tudi za druge vire: Scopus, BibTEX, Zentralblatt Math, Google Scholar, DBLP, IMDB, itd.

Evropski projekti na temo simulacij

Za srečanje **The Age of Simulation** januarja 2006 na Ars Electronica v Linzu smo skupaj s sodelavci podjetja FAS z Dunaja analizirali podatke o evropskih projektih na temo simulacij. Podatke so zbrali sodelavci FASa s spletišča projektov in jih uredili v obliki velike tabele v Excelu. Posamezni zapis sestavljajo različni podatki o posamezni sodelujoči ustanovi na posameznem projektu. Tabelo smo najprej shranili v obliki CSV in nato s programom Text2Pajek iz nje ustvarili tri dvovrstna omrežja:

- `project.net` – idents $\times$ projects $=$ **P**
- `country.net` – idents $\times$ countries $=$ **C**
- `institution.net` – idents $\times$ institutions $=$ **U**

Velikosti posameznih množic so naslednje:
$|\text{idents}| = 8869$, $|\text{projects}| = 933$, $|\text{institutions}| = 3438$, $|\text{countries}| = 60$.

Evropski projekti – izpeljana omrežij

Ker imajo vsa tri omrežja skupno množico $\mathcal{K}$ =idents, lahko na prej opisani način iz njih z množenjem pridobimo različna *izpeljana* omrežja:

- `ProjInst.net` – projects $\times$ institutions $\mathbf{W} = \mathbf{P}^T \star \mathbf{U}$
- `Countries.net` – countries $\times$ countries $\mathbf{S} = \mathbf{C}^T \star \mathbf{C}$
- `Institutions.net` – institutions $\times$ institutions $\mathbf{Q} = \mathbf{W}^T \star \mathbf{W}$
- ...

Obstajata posebna ukaza za $\mathbf{A}^T * \mathbf{A}$ in $\mathbf{A} * \mathbf{A}^T$

```
Network/2-Mode Network/2-Mode to 1-Mode/Rows
Network/2-Mode Network/2-Mode to 1-Mode/Columns
```

Evropski projekti – izločeni projekti

Za 27 zapisov v tabeli podatki niso popolni. Običajno problem rešimo tako, da ustvarimo novo tabelo, iz katere te zapise izločimo. No, mogoča je tudi druga pot: ustvarimo skupino C_D nepopolnih zapisov in iz nje matriko $\mathbf{D}$ – idents $\times$ idents. Matrika $\mathbf{D}$ je diagonalna matrika z vrednostjo 1 za popolne zapise in vrednostjo 0 za zapise iz C_D. Z matriko $\mathbf{D}$ lahko na primer določimo omrežje `ProjInst.net` iz celotne tabele kot $\mathbf{W} = \mathbf{P}^T \star \mathbf{D} \star \mathbf{U}$ – nepopolni zapisi ne prispevajo k omrežju.

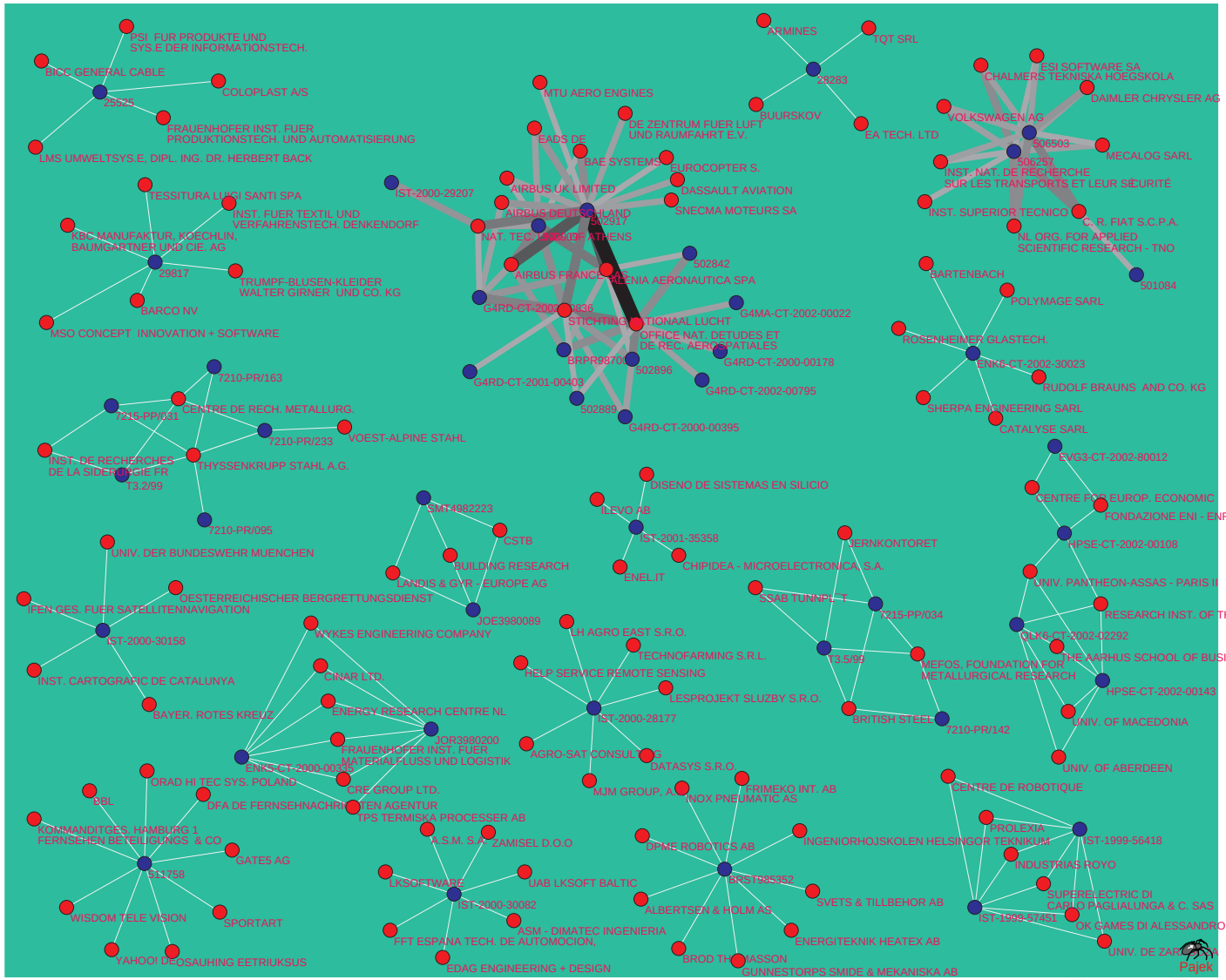
Analiza omrežja `ProjInst.net`

Za določitev pomembnih delov omrežja `ProjInst.net` smo najprej določili omrežje 4-obročnih uteži in na tem omrežju določili povezavne otoke:

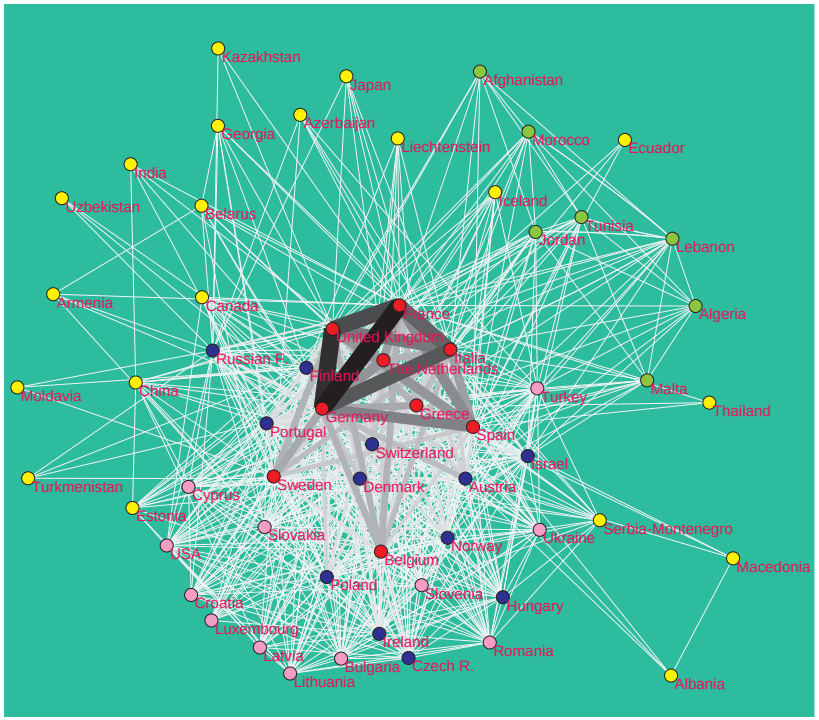
```
Network/Create New Network/With Ring Counts .../4-Rings/Undirected
Network/Create Partition/Islands/Line Weights[Simple] [2,200]
```

Dobili smo 101 otok, 18 med njimi ima velikost vsaj 5 (vozlišč). Najpomembnejša otoka sestavljajo letalske ustanove in avtomobilske ustanove.
V zapisu nekaterih oznak vozlišč smo uporabili `\n`, ki oznako prelomi.
Za analizo bi lahko uporabili tudi $(p, q)-$sredice.

Analiza omrežja ProjInst.net



Analiza omrežja Countries.net



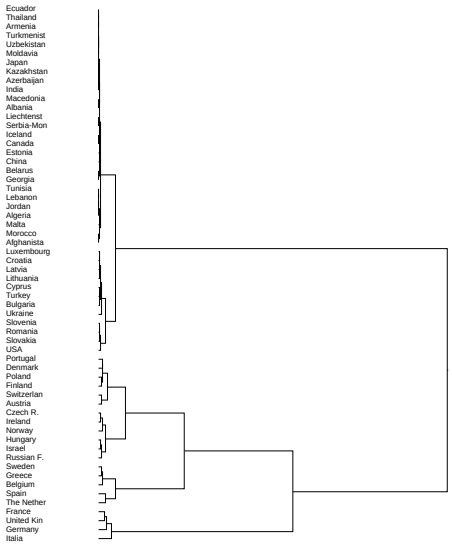
Omrežje Countries.net ima le 60 vozlišč, je pa gosto. Uteži povezav predstavljajo število projektov, pri katerih hkrati sodelujeta krajiščni državi. Za preglednejšo sliko moramo povezave urediti glede na uteži:

`Network/Create New Network/`
`Transform/Sort lines/`
`Line values/Ascending`

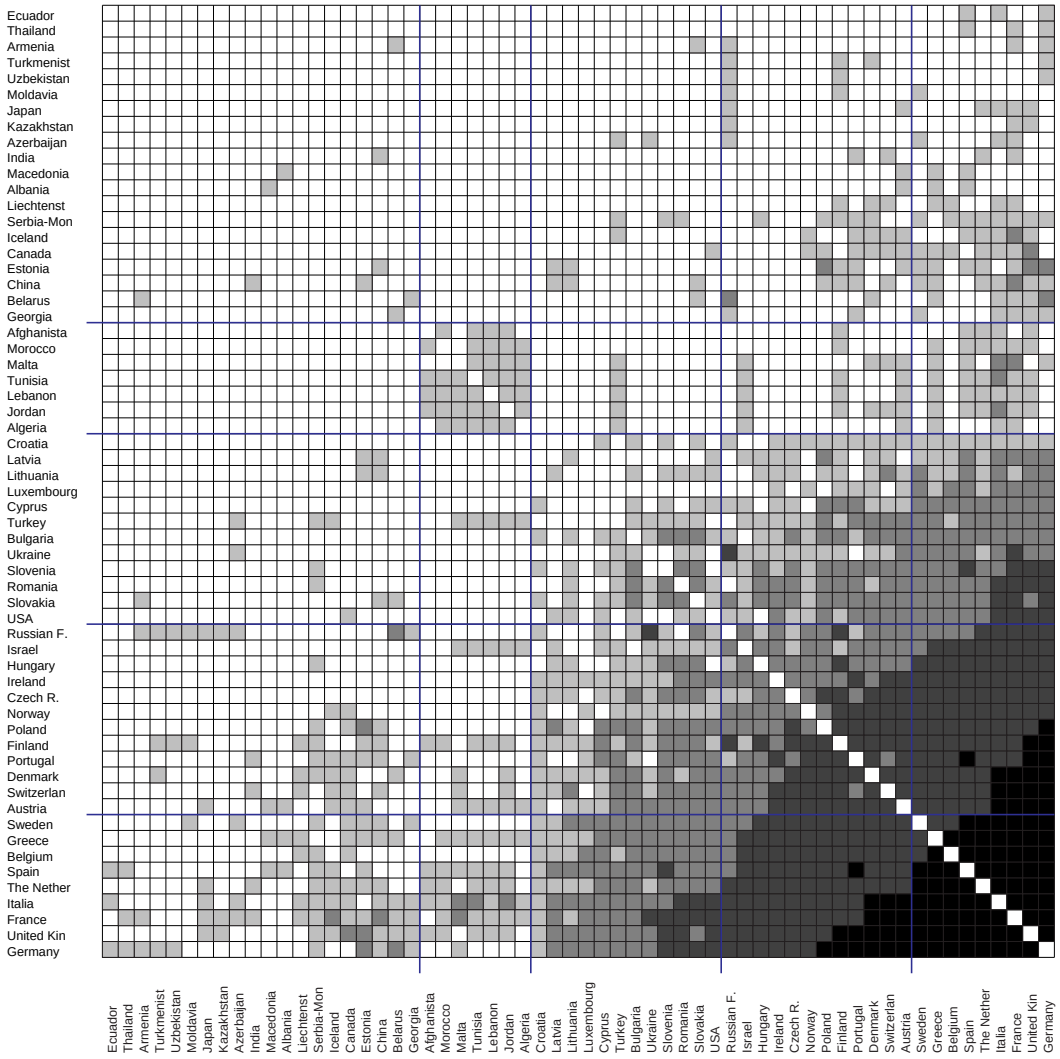
Še preglednejši vpogled v sodelovanje med državami dobimo z matričnim prikazom. Ustrezno urejenost dobimo iz hierarhične razvrstitve dobljene z Wardovim postopkom uporabljenim nad različnostjo d_5. Uteži so prekodirane glede na prage (2,10,50).
Urejenost lahko še izboljšamo s preurejanjem poddreves v hierarhiji. Kakor vidimo na sliki, dobimo značilno (večslojno) zgradbo središče – obrobje.

Analiza omrežja Countries.net

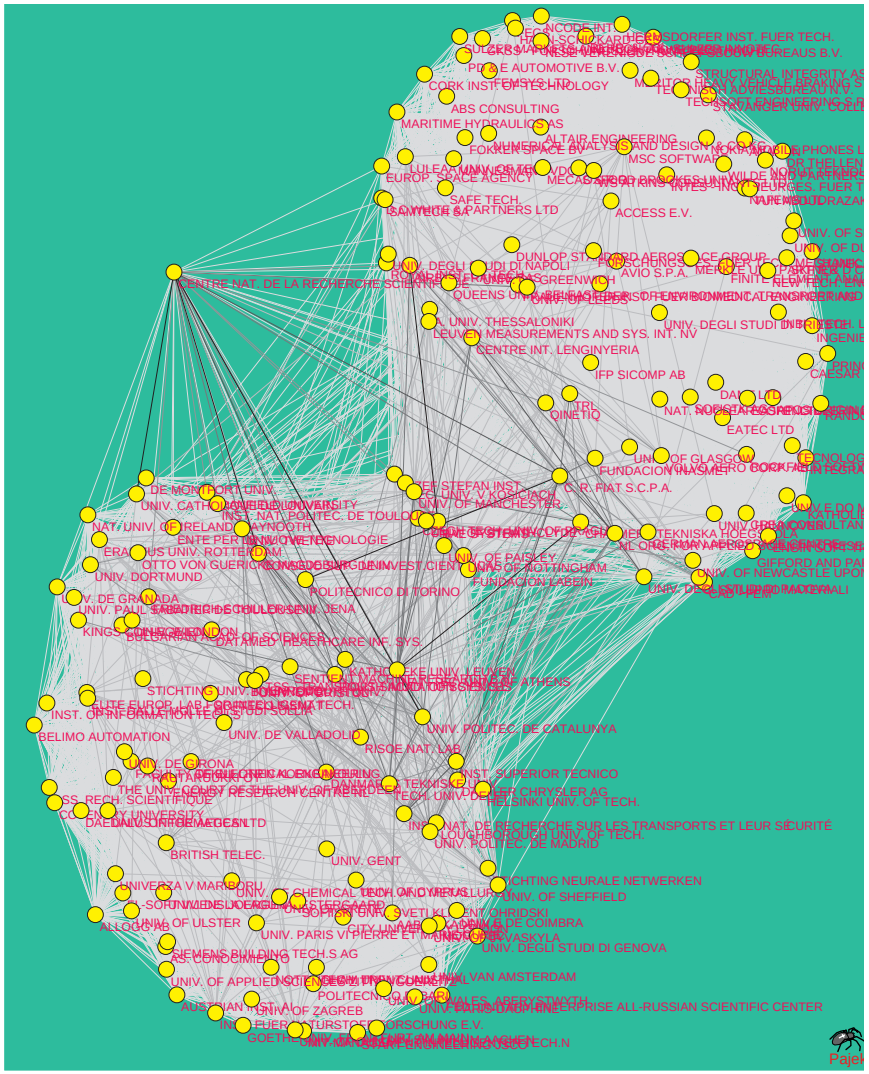
Pajek - Ward [0.00,4785.14]



Pajek - shadow [0.00,4.00]

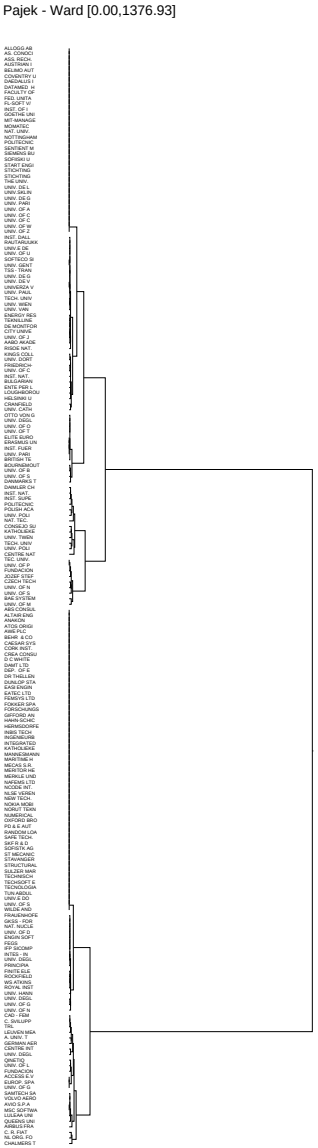


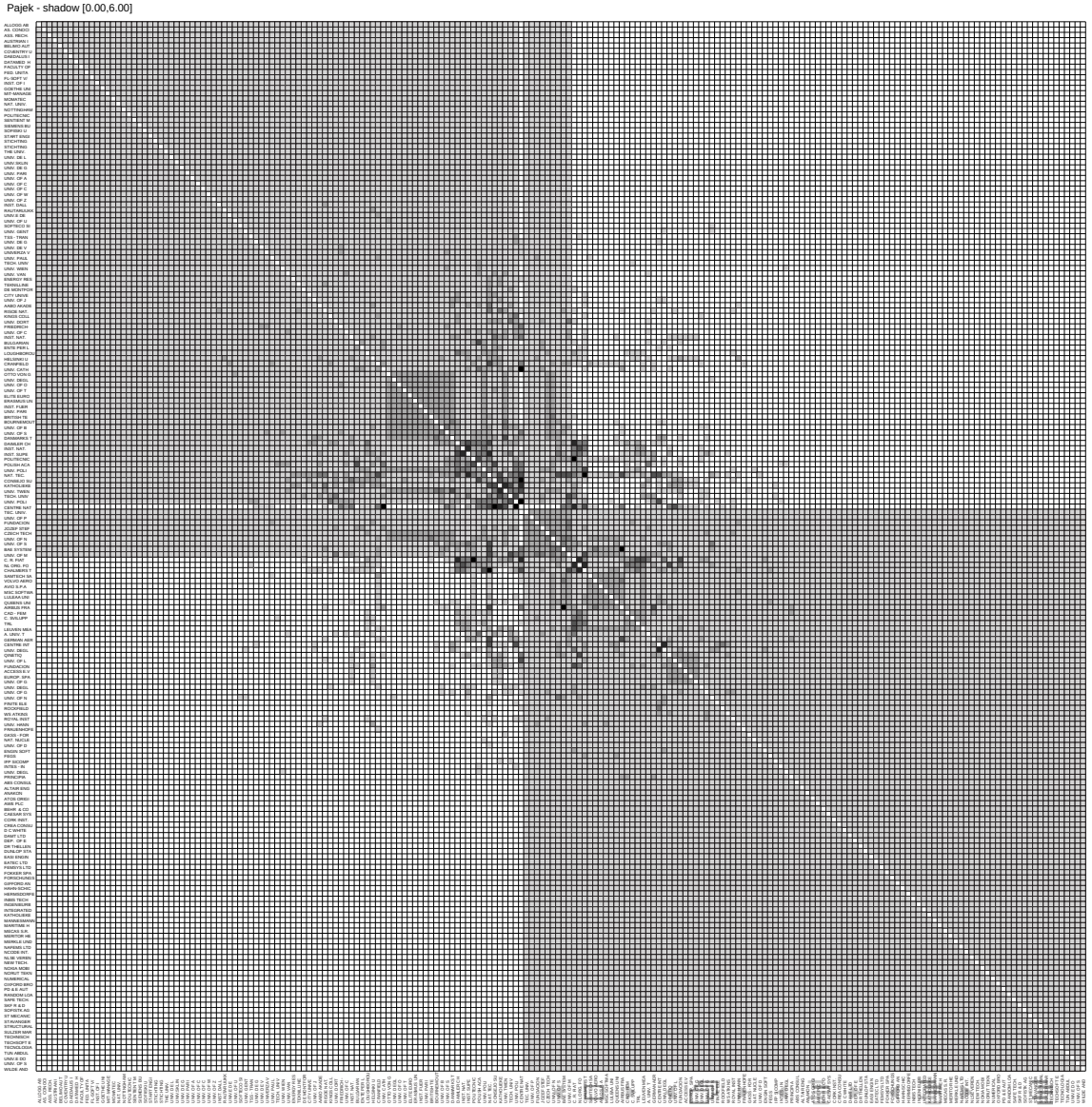
Analiza omrežja Institutions.net



Za določitev najpomembnejših ustanov smo najprej določili sredice vrste p_S in za tako določen vektor določili vozliščne otoke. V bistvu smo dobili en sam velik otok prikazan na sliki. Ta je sestavljen iz dveh večjih skupin povezanih čez posredniške ustanove (med njimi je tudi Inštitut Jožef Stefan). Posebej odstopajoče vozlišče je Centre national de la recherche scientifique (CNRS). Še bolje je medsebojna povezanost med ustanovami razvidna iz matričnega prikaza.

Analiza omrežja Institutions.net





Pretvorba dvovrstnih omrežij na enovrstna

Z uporabo množenja omrežij lahko dvovrstno omrežje $\mathcal{N} = (\mathcal{I}, \mathcal{J}, \mathcal{E}, w)$ pretvorimo v dve enovrstni omrežji $\mathcal{N}_1 = \mathcal{N} * \mathcal{N}^T$ in/ali $\mathcal{N}_2 = \mathcal{N}^T * \mathcal{N}$. Naj bo $\mathbf{A}$ matrika omrežja $\mathcal{N}$ in $\mathbf{B}$ matrika omrežja $\mathcal{N}_1$. Tedaj velja $\mathbf{B} = \mathbf{A}\mathbf{A}^T$, $b_{uv} = \sum_{z \in \mathcal{J}} a_{uz} \cdot a^T_{zv} = \sum_{z \in \mathcal{J}} a_{uz} \cdot a_{vz}$. Očitno velja $b_{uv} = b_{vu}$ – matrika $\mathbf{B}$ je simetrična.

Tudi omrežju $\mathcal{N}_2$ pripadajoča matrika $\mathbf{C} = \mathbf{A}^T\mathbf{A}$ je simetrična.

Enovrstni omrežji $\mathcal{N}_1$ in $\mathcal{N}_2$ lahko analiziramo z običajnimi postopki analize enovrstnih omrežij. Osnovna težava je, da sta lahko eno ali celo obe preveliki.

Normalizacije

Normalizacije naj bi omogočile hiter pregled enovrstnih omrežij, ki jih dobimo iz dvovrstnih.

V enovrstnih omrežjih dobljenih iz velikih dvovrstnih omrežij so pogosto razlike v utežeh zelo velike. Zato ni mogoče primerjati vozlišč glede na te vrednosti. Pred primerjavo jih moramo normalizirati – poskrbeti za primerljivost uteži.

Obstaja več načinov, kako lahko to naredimo. Nekaj izmed njih je prikazanih v tabeli na naslednji prosojnici. Uporabimo jih lahko tudi na drugih enovrstnih uteženih omrežjih.

V primeru omrežij brez zank postavimo za neusmerjena omrežja diagonalne vrednosti na vsoto izvendiagonalnih elementov v pripadajoči vrstici (ali stolpcu) $w_{vv} = \sum_{u \neq v} w_{vu}$; za usmerjena omrežja pa neko srednjo vrednost iz vrstične in stolpčne vsote – npr. $w_{vv} = \frac{1}{2}(\sum_{u \neq v} w_{vu} + \sum_{u \neq v} w_{uv})$. Običajno privzamemo, da omrežje nima osamljenih vozlišč.

... Normalizacije

$$\mathrm{Geo}_{uv} = \frac{w_{uv}}{\sqrt{w_{uu} w_{vv}}} \qquad \mathrm{GeoDeg}_{uv} = \frac{w_{uv}}{\sqrt{\deg_u \deg_v}}$$

$$\mathrm{Input}_{uv} = \frac{w_{uv}}{w_{vv}} \qquad \mathrm{Output}_{uv} = \frac{w_{uv}}{w_{uu}}$$

$$\mathrm{Min}_{uv} = \frac{w_{uv}}{\min(w_{uu}, w_{vv})} \qquad \mathrm{Max}_{uv} = \frac{w_{uv}}{\max(w_{uu}, w_{vv})}$$

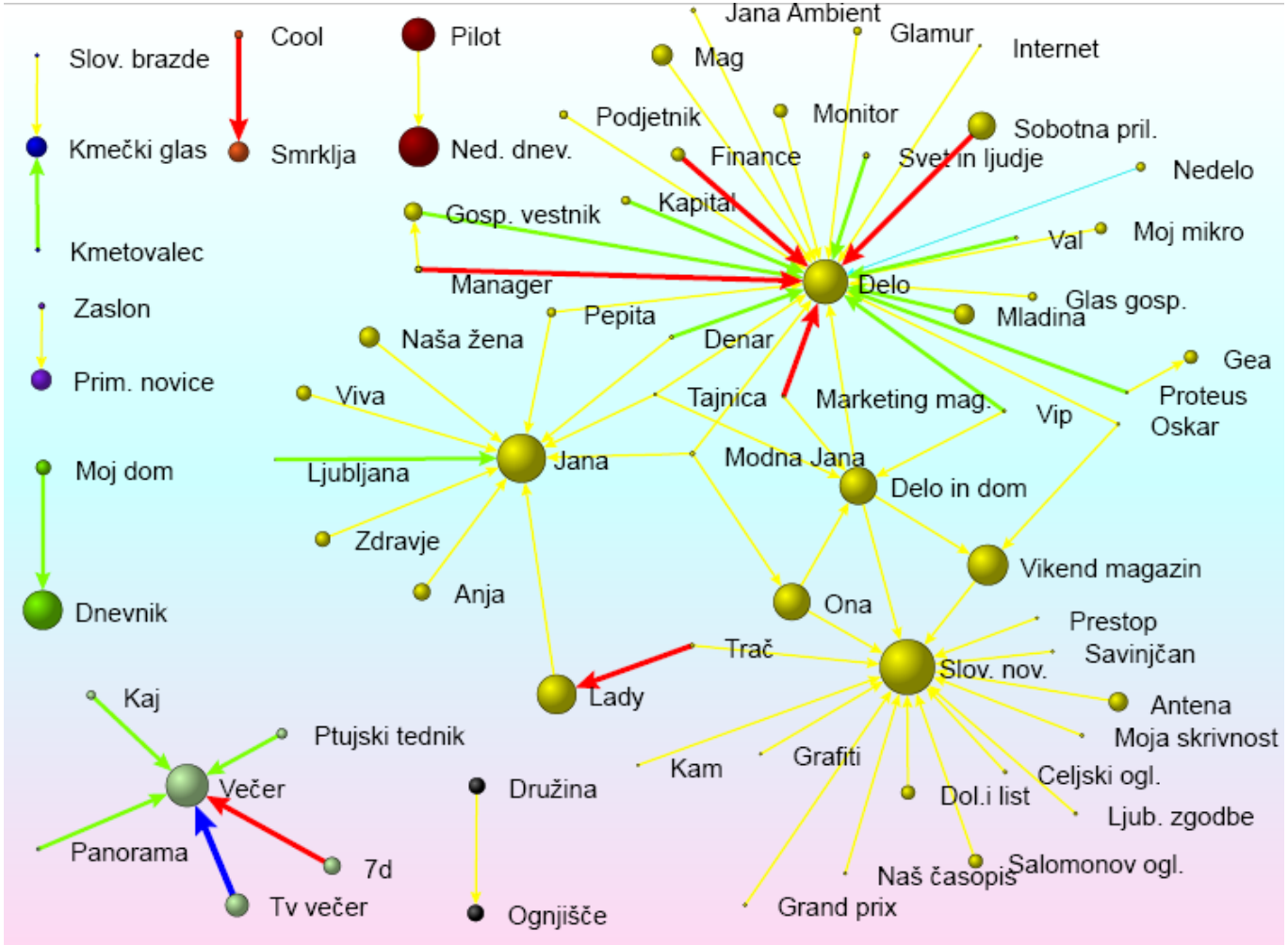
$$\mathrm{MinDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{uu}} & w_{uu} \leq w_{vv} \\ 0 & \textit{sicer} \end{cases} \qquad \mathrm{MaxDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{vv}} & w_{uu} \leq w_{vv} \\ 0 & \textit{sicer} \end{cases}$$

Normalizirano omrežje analiziramo z uporabo povezavnih prerezov ali otokov.

`Network/2-Mode Network/2-Mode to 1-Mode/Normalize 1-Mode/`

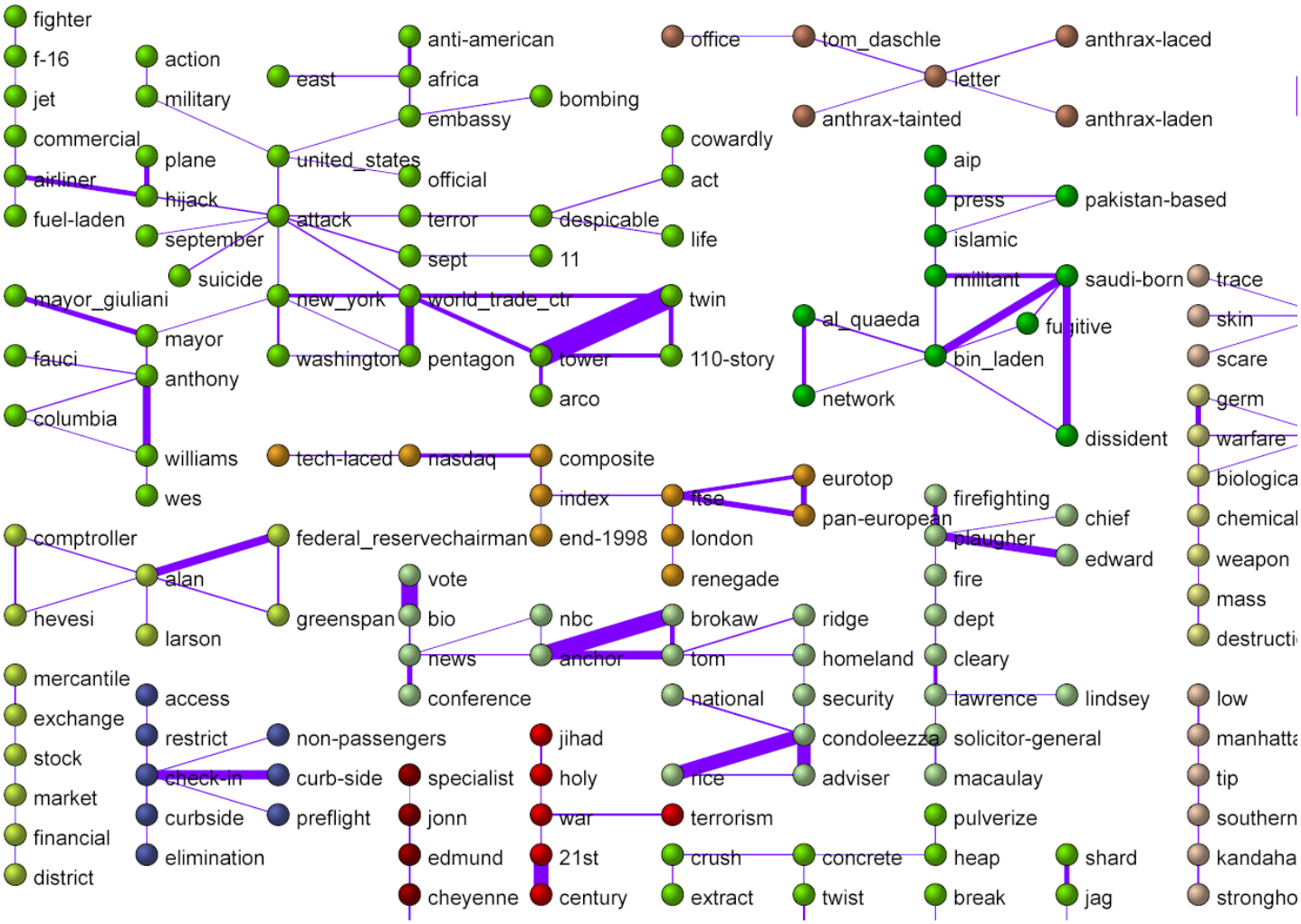
Reuters Terror News: GeoDeg, MaxDir, MinDir.

MinDir na omrežju Slovenski časopisi 2000

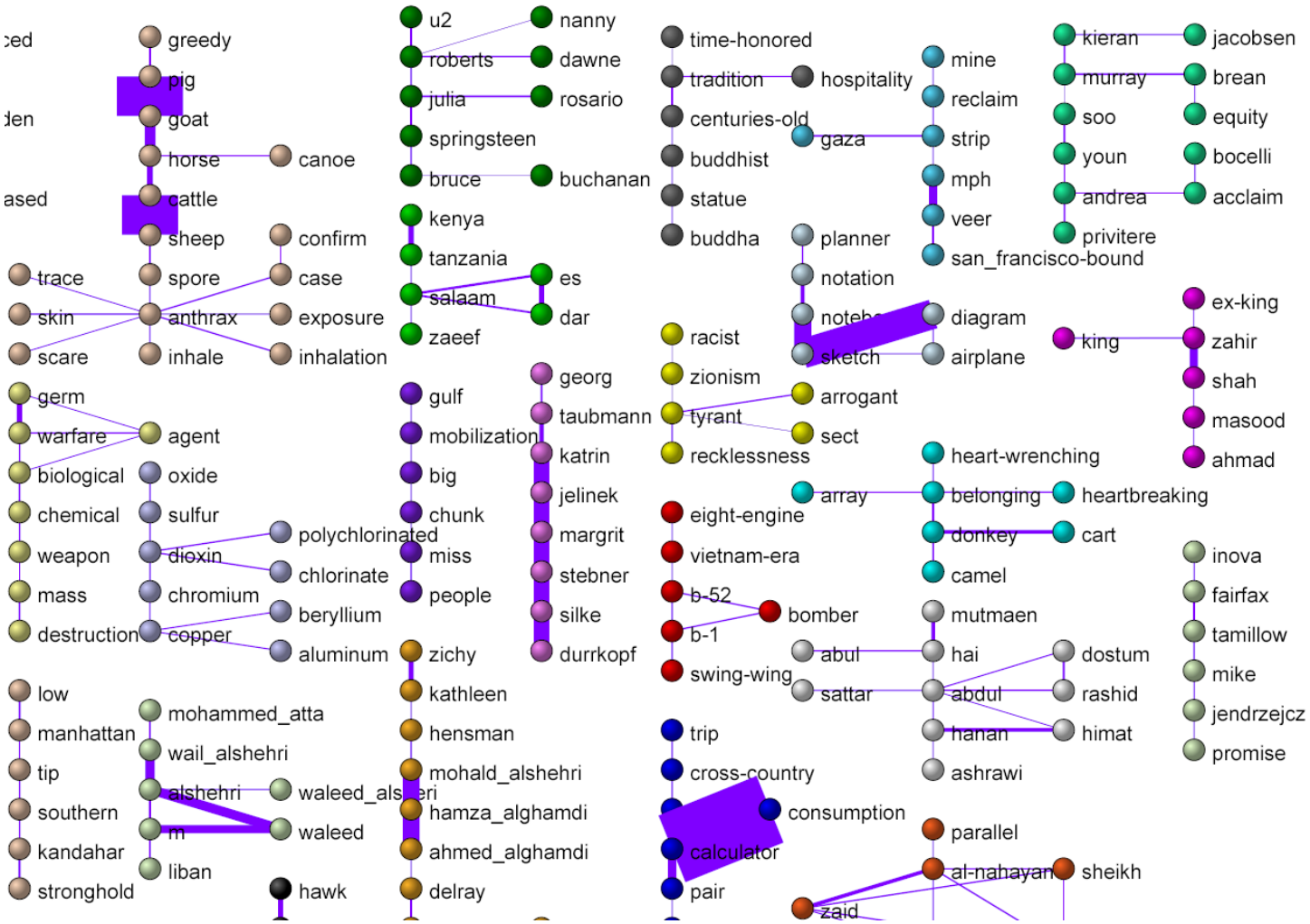


Čez 100000 oseb je bilo v letih 1999 in 2000 vprašanih, katere časopise berejo. Navedli so 124 različnih časopisov. (vir Cati)

GeoDeg na omrežju Reuters terror news



GeoDeg na omrežju Reuters terror news



Omrežje avtorstev

V dvovrstnem *omrežju avtorstev* **WA** na dela $\times$ avtorji imajo povezave (p, i) utež $wa_{pi} = 1$ in povedo, da je oseba i (so)avtor dela p.

$$\forall p \in W : \sum_{i \in A} wa_{pi} = \text{outdeg}_{WA}(p) = \ \# \text{ avtorjev dela } p$$

Označimo z **N** njegovo normalizirano različico

$$\forall p \in W : \sum_{i \in A} n_{pi} \in \{0, 1\}$$

ki jo dobimo iz **WA** z $n_{pi} = wa_{pi} / \max(1, \text{outdeg}_{WA}(p))$ ali pa po kakem drugem pravilu, ki določa delež posameznega avtorja.
$i \sim p \ \Leftrightarrow \ i \in N(p) \ \Leftrightarrow \ p \in N^{-}(i) \ \Leftrightarrow \ wa_{pi} > 0$

Nekaj transformacij omrežij

Graf omrežja $b(\mathcal{N})$ je omrežje, ki ga dobimo iz danega omrežja $\mathcal{N}$ tako, da vse njegove uteži postavimo na 1.

Transpozicija ali preusmeritev $\mathcal{N}^T$ ali $t(\mathcal{N})$ je omrežje, ki ga dobimo iz danega omrežja $\mathcal{N}$ tako, da obrnemo smeri vseh njegovih usmerjenih povezav. $\mathbf{AW} = \mathbf{WA}^T$, $\mathbf{KW} = \mathbf{WK}^T$, ...

(Izhodna) normalizacija $n(\mathcal{N})$ je omrežje, ki ga dobimo iz danega omrežja $\mathcal{N}$ tako, da utež vsake povezave a delimo z vsoto uteži vseh povezav, ki imajo isto začetno vozlišče kot povezava a. Za omrežja s konstantno utežjo velja

$$n(\mathbf{A}) = \mathrm{diag}\left(\frac{1}{\max(1, \mathrm{outdeg}_{WA}(i))}\right)_{i \in \mathcal{I}} * \mathbf{A}$$

$\mathbf{N} = n(\mathbf{WA})$, $\mathbf{WA} = b(\mathbf{N})$

Prvo omrežje sodelovanj

Co = **AW** $*$ **WA**

$$co_{ij} = \sum_{p \in W} wa_{pi} wa_{pj} = \sum_{p \in N^-(i) \cap N^-(j)} 1$$

co_{ij} = število del, ki sta jih avtorja i in j napisala skupaj

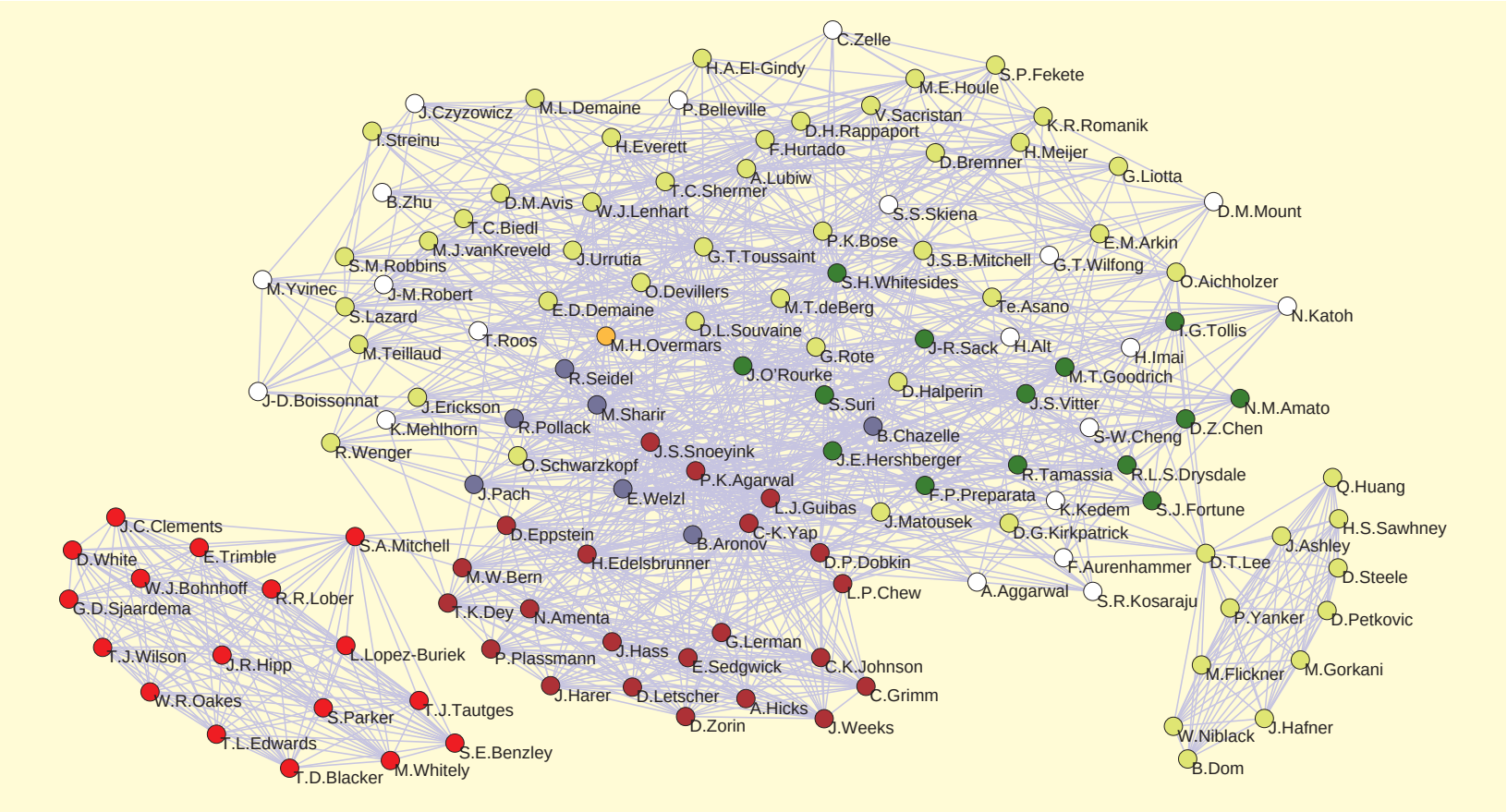
Velja: $co_{ij} = co_{ji}$.

Z uporabo uteži co_{ij} lahko izrazimo Saltonovo kosinusno podobnost imenovano tudi Ochiai-jev koeficient med avtorjema i in j

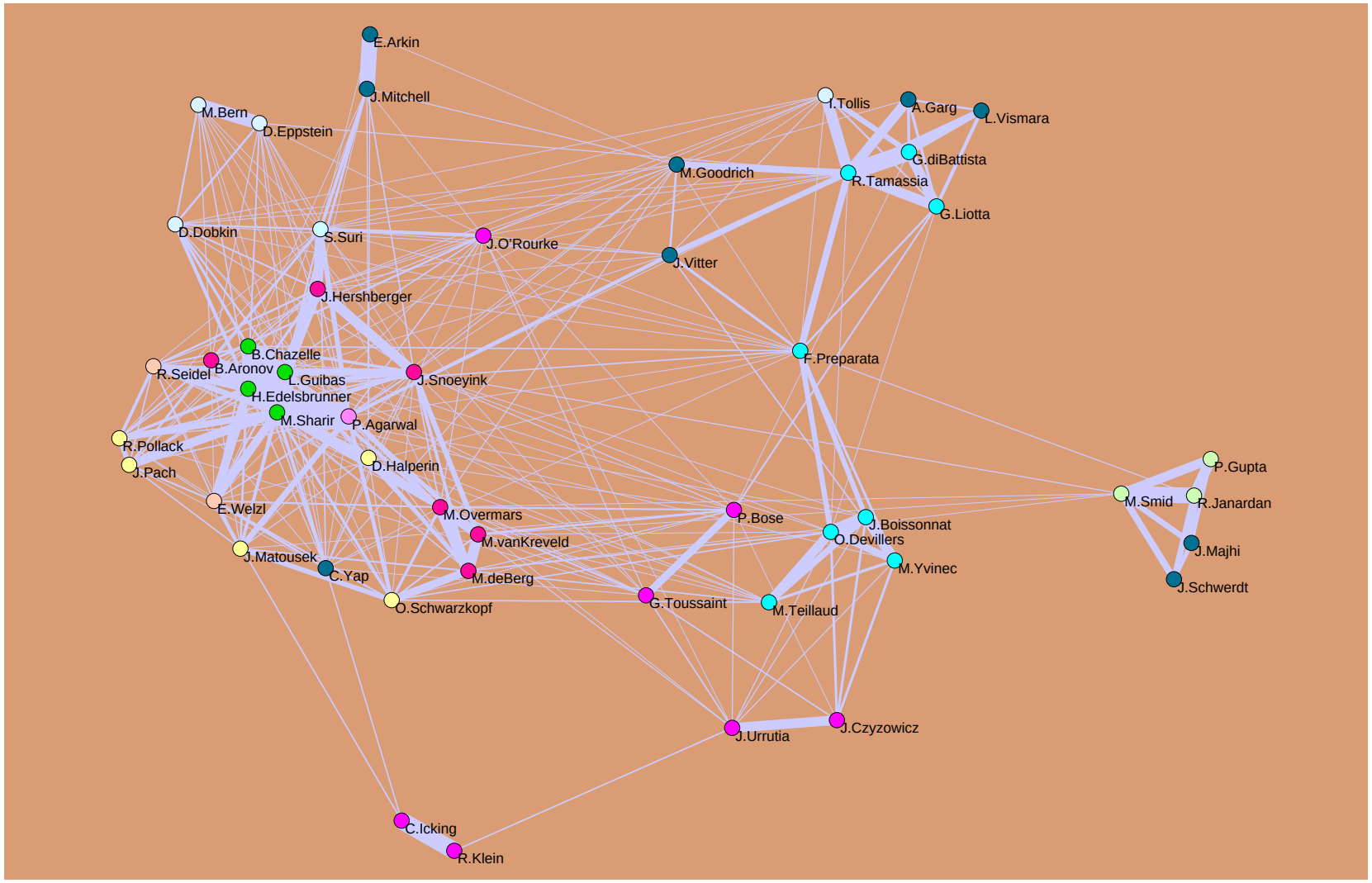
$$cos(i,j) = \frac{co_{ij}}{\sqrt{co_{ii} co_{jj}}}, \qquad \text{for } co_{ij} > 0$$

Težava: Kot smo že povedali, je omrežje **Co** lepljenka polnih podgrafov. Dela z velikim številom avtorjev ustvarijo velike polne podgrafe in zameglijo sliko sodelovanj.

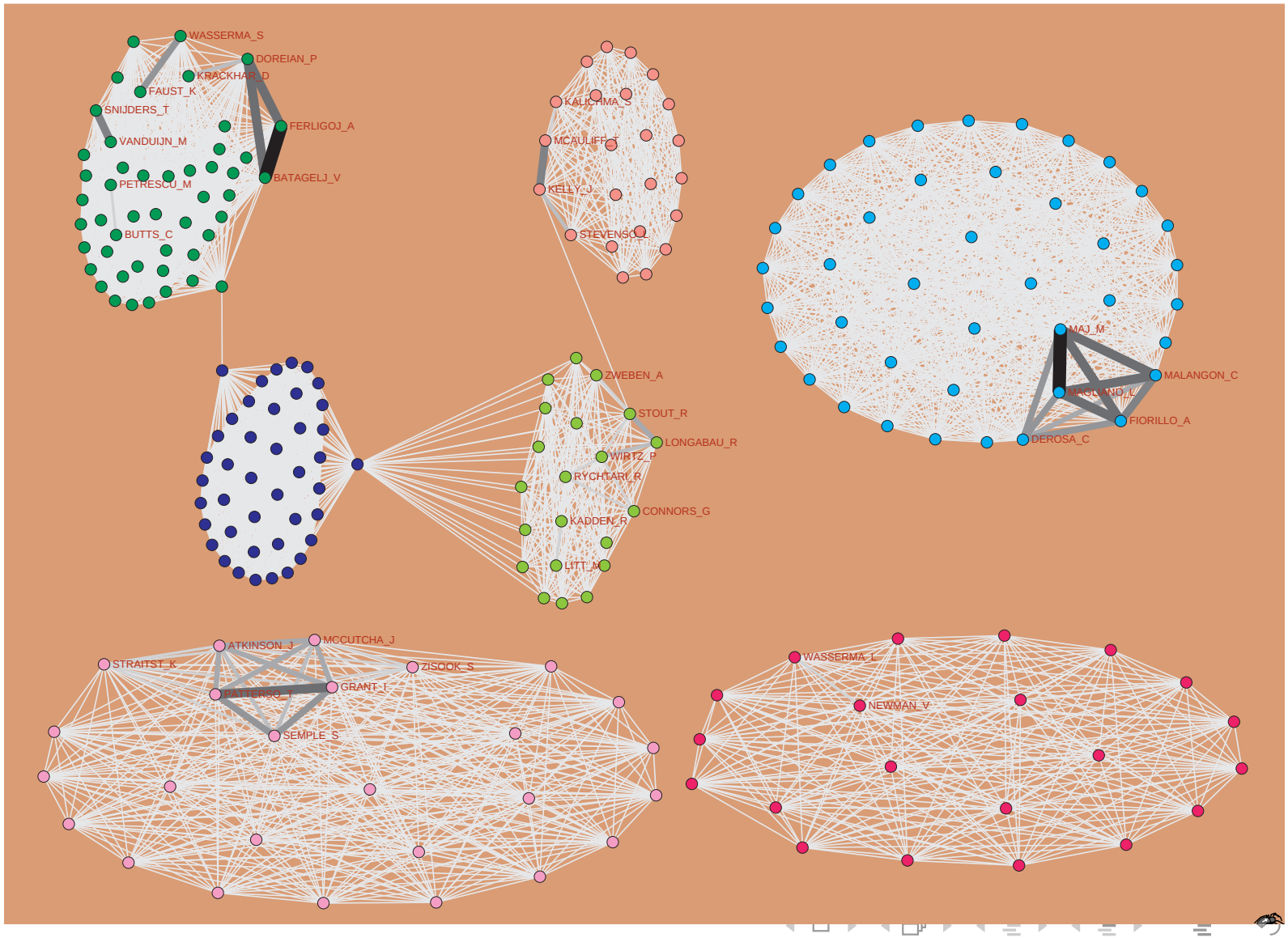
Sredice redov 10–21 v Computational Geometry



p_S-sredica za prag 46 v Computational Geometry



Sredice redov 20–47 v **Co**(SN5)



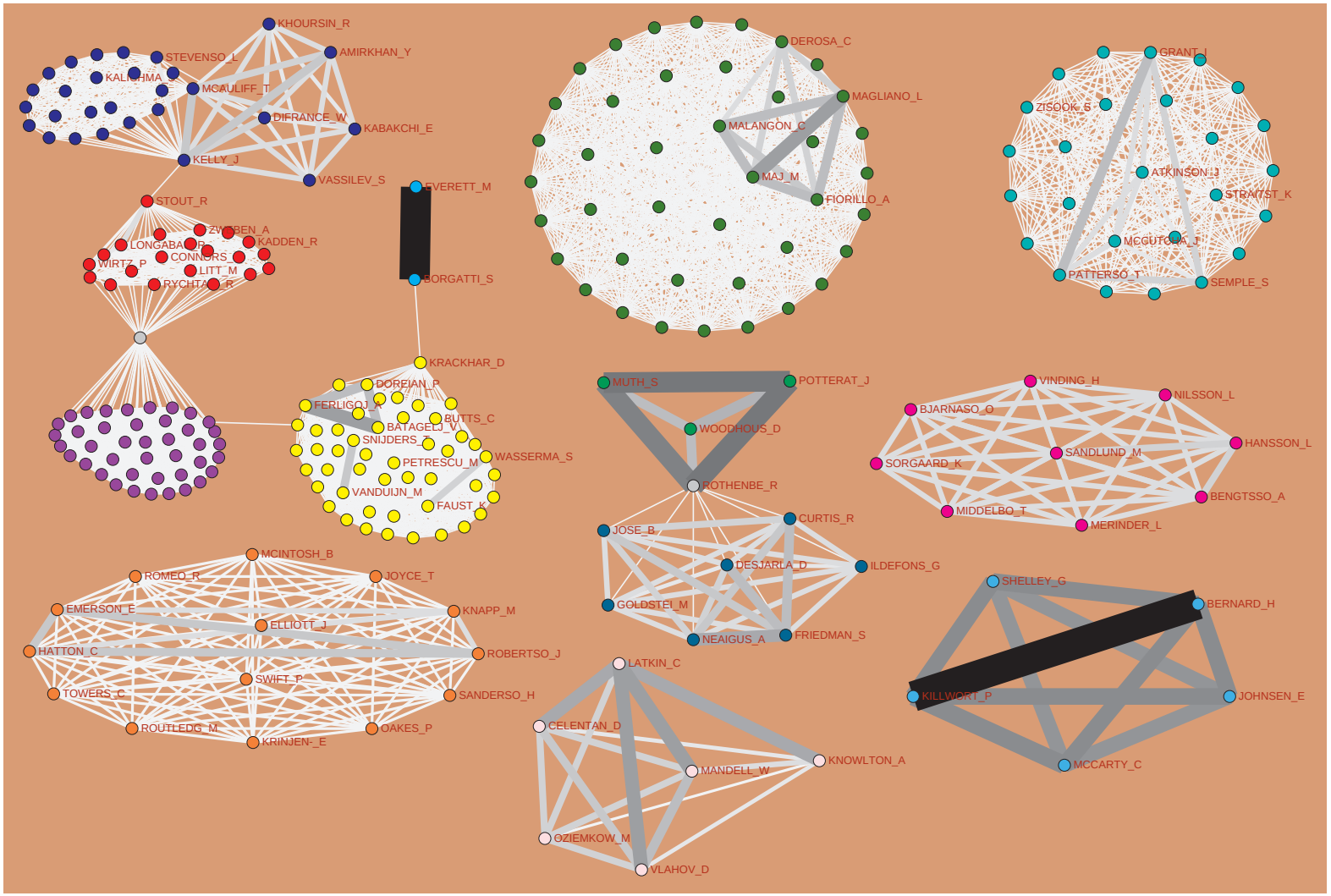
Članki glede na število avtorjev v SN5

outdeg	frequency	outdeg	frequency	paper
1	2637	12	8	
2	2143	13	4	
3	1333	14	3	
4	713	15	2	
5	396	21	1	Pierce et al. (2007)
6	206	22	1	Allen et al. (1998)
7	114	23	1	Kelly et al. (1997)
8	65	26	1	Semple et al. (1993)
9	43	41	1	Magliano et al. (2006)
10	24	42	1	Doll et al. (1992)
11	10	48	1	Snijders et al. (2007)

Snijders et al. (2007)

Snijders et al.(2007): Snijders, T.A.B., Robinson, T., Atkinson, A.C., Riani, M., Gormley, I.C., Murphy, T.B., Sweeting, T., Leslie, D.S., Longford, N.T., Kent, J.T., Lawrance, T., Airoldi, E.M., Besag, J., Blei, D., Fienberg, S.E., Breiger, R., Butts, C.T., Doreian, P., Batagelj, V., Ferligoj, A., Draper, D., van Duijn, M.A.J., Faust, K., Petrescu-Prahova, M., Forster, J.J., Gelman, A., Goodreau, S. M., Greenwood, P.E., Gruenberg, K., Francis, B., Hennig, C., Hoff, P.D., Hunter, D.R., Husmeier, D., Glasbey, C., Krackhardt, D., Kuha, J., Skrondal, A., Lawson, A., Liao, T. F., Mendes, B., Reinert, G., Richardson, S., Lewin, A., Titterington, D.M., Wasserman, S., Werhli, A.V. and Ghazal, P.. *Discussion on the paper by Handcock, Raftery and Tantrum*. Journal of the Royal Statistical Society: Series A - Statistics in Society, 170 (2007), pp. 322-354.

p_S-sredica za prag 20 v **Co**(SN5)



Drugo omrežje sodelovanj

$\mathbf{Cn} = \mathbf{AW} * \mathbf{N}$

$$cn_{ij} = \sum_{p \in W} wa_{pi} n_{pj} = \sum_{p \in N^-(i) \cap N^-(j)} n_{pj}$$

cn_{ij} = prispevek avtorja j k delom, ki jih napisal skupaj z avtorjem i.

Velja $\sum_{j \in A} \sum_{j \in A} wa_{pi} n_{pj} = \text{outdeg}_{WA}(p)$ in $\sum_{j \in A} cn_{ij} = \text{indeg}_{WA}(i)$

Če je $n_{pi} = \frac{wa_{pi}}{\text{outdeg}(p)}$, velja še $cn_{ij} = cn_{ji}$.

$cn_{ii} = \sum_{p \in N(i)} n_{pi}$ je prispevek avtorja i svojim delom.

Samozadostnost: $S_i = \frac{cn_{ii}}{\text{outdeg}_{WA}(i)}$

Sodelovalnost: $K_i = 1 - S_i$

$$\sum_{i \in A} \sum_{j \in A} cn_{ij} = \sum_{i \in A} \text{indeg}_{WA}(i) = m_{WA}$$

Za izračun tabele smo v programu `Pajek` pripravili makro.

"Najboljši" avtorji na področju Social Networks

i	author	cn_{ii}	total	K_i	i	author	cn_{ii}	total	K_i
1	Burt,R	43.83	53	0.173	26	Latkin,C	10.14	37	0.726
2	Newman,M	36.77	60	0.387	27	Morris,M	9.98	20	0.501
3	Doreian,P	34.44	47	0.267	28	Rothenberg,R	9.82	28	0.649
4	Bonacich,P	30.17	41	0.264	29	Kadushin,C	9.75	11	0.114
5	Marsden,P	29.42	37	0.205	30	Faust,K	9.72	18	0.460
6	Wellman,B	26.87	41	0.345	31	Batagelj,V	9.69	20	0.516
7	Leydesdorf,L	24.37	35	0.304	32	Mizruchi,M	9.67	15	0.356
8	White,H	23.50	33	0.288	33	[Anon]	9.00	9	0.000
9	Friedkin,N	20.00	23	0.130	34	Johnson,J	8.89	21	0.577
10	Borgatti,S	19.20	41	0.532	35	Fararo,T	8.83	16	0.448
11	Everett,M	16.92	31	0.454	36	Lazega,E	8.50	12	0.292
12	Litwin,H	16.00	21	0.238	37	Knoke,D	8.33	11	0.242
13	Freeman,L	15.53	20	0.223	38	Ferligoj,A	8.19	19	0.569
14	Barabasi,A	14.99	35	0.572	39	Brewer,D	8.03	11	0.270
15	Snijders,T	14.99	30	0.500	40	Klovdahl,A	7.96	17	0.532
16	Valente,T	14.80	34	0.565	41	Hammer,M	7.92	10	0.208
17	Breiger,R	14.44	20	0.278	42	White,D	7.83	15	0.478
18	Skvoretz,J	14.43	27	0.466	43	Holme,P	7.42	14	0.470
19	Krackhardt,D	13.65	25	0.454	44	Boyd,J	7.37	13	0.433
20	Carley,K	12.93	28	0.538	45	Kilduff,M	7.25	16	0.547
21	Pattison,P	12.10	27	0.552	46	Small,H	7.00	7	0.000
22	Wasserman,S	11.72	26	0.549	47	Iacobucci,D	7.00	12	0.417
23	Berkman,L	11.21	30	0.626	48	Pappi,F	6.83	10	0.317
24	Moody,J	10.83	15	0.278	49	Chen,C	6.78	12	0.435
25	Scott,J	10.47	15	0.302	50	Seidman,S	6.75	9	0.250

Tretje omrežje sodelovanj

$\mathbf{Ct} = \mathbf{N}^T * \mathbf{N}$

ct_{ij} = skupni prispevek sodelovanja avtorjev i in j delom.

Velja $\quad ct_{ij} = ct_{ji} \quad$ in

$$\sum_{i \in A} \sum_{j \in A} n_{pi} n_{pj} = 1$$

Skupni prispevek polnega podomrežja, ki pripada posameznemu članku p, je enak 1.

$\sum_{j \in A} ct_{ij} = \sum_{p \in W} n_{pi}$ = skupni prispevek avtorja i delom iz W.

$$\sum_{i \in A} \sum_{j \in A} ct_{ij} = |W^+_{\mathbf{WA}}|$$

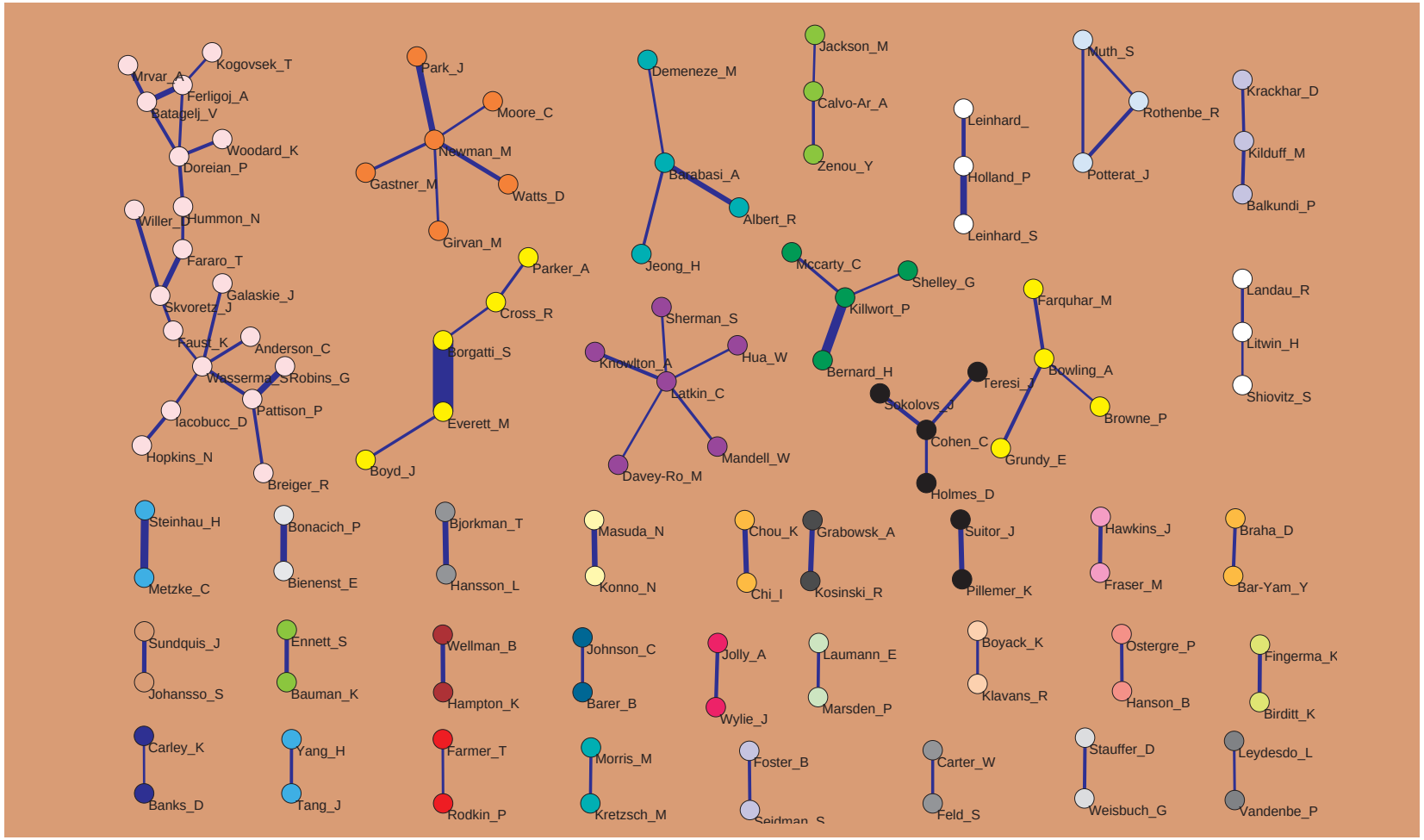
Prispevek avtorja i: $\quad ac_i = \dfrac{|A|}{|W^+_{\mathbf{WA}}|} \sum_{p \in W} n_{pi} \quad$ in $\quad \sum_{i \in A} ac_i = |A|$.

Torej je povprečni avtorjev prispevek ac enak 1.

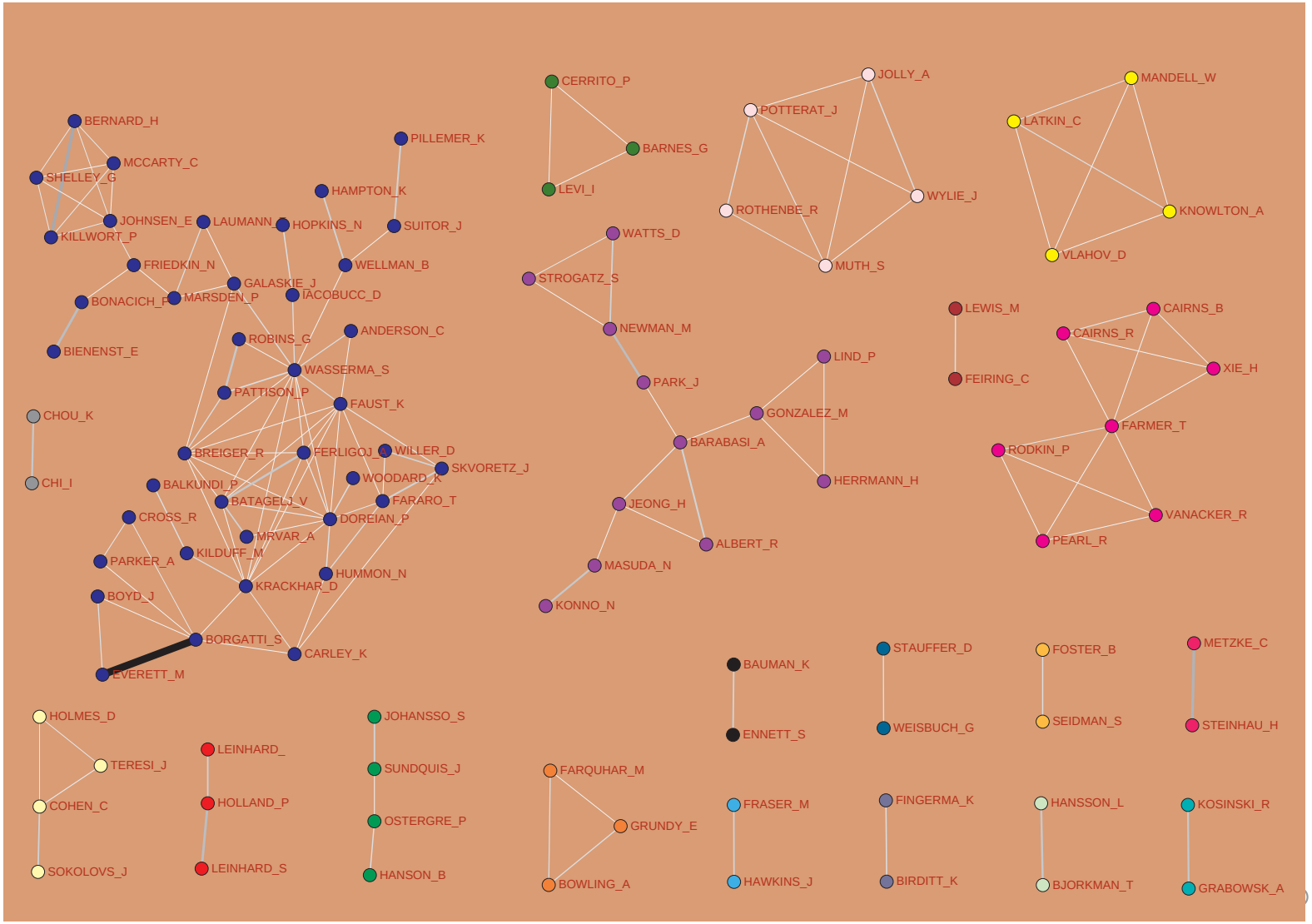
Vsa tri omrežja sodelovanj imajo isti graf: $b(\mathbf{Co}) = b(\mathbf{Cn}) = b(\mathbf{Ct})$.

Komponente **Ct**(SN5) za izrez pri pragu 0.5

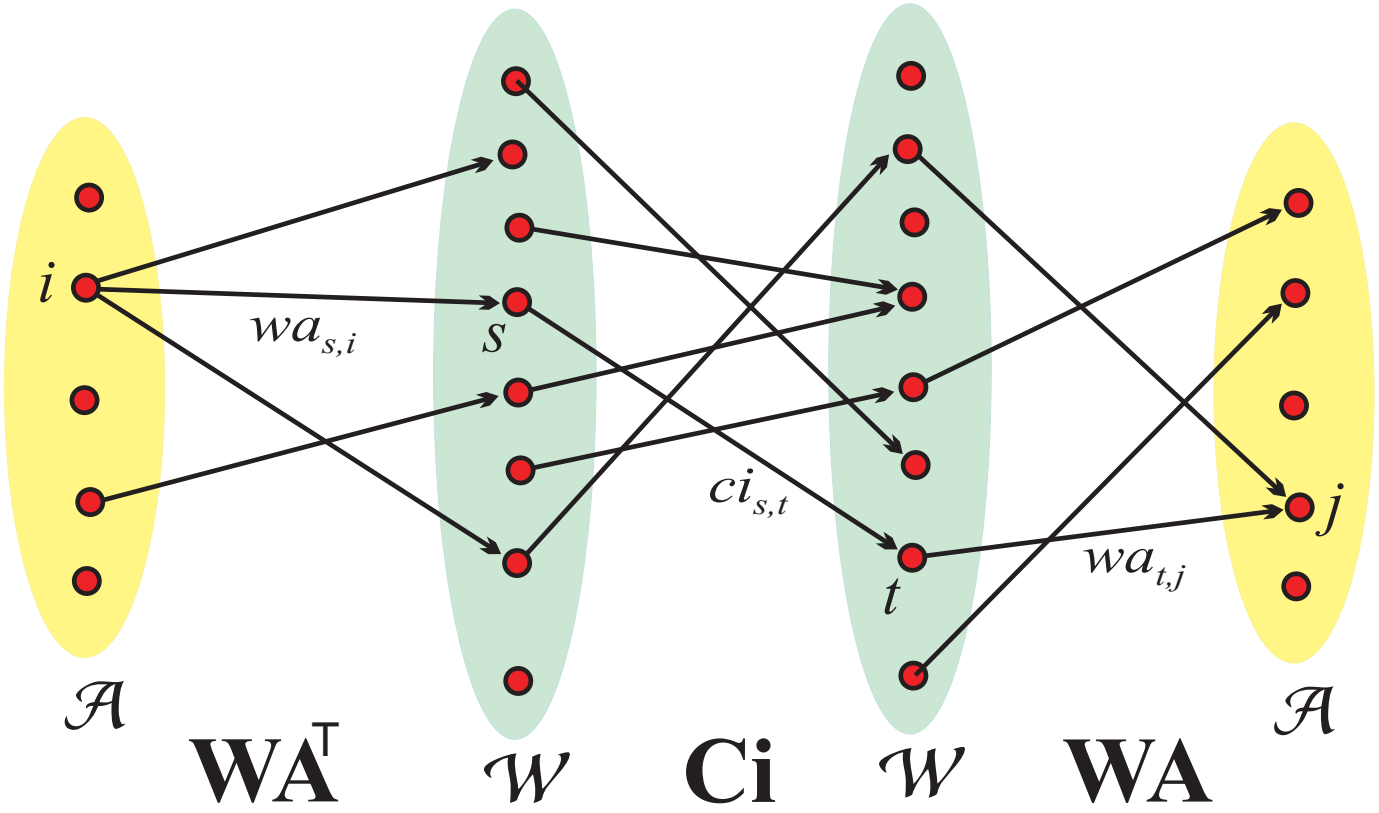
Omrežje SN5 (2008): za "social network*" + članki z največ sklici nanje + okrog 100 raziskovalcev;
$|W| = 193376$, $|C| = 7950$, $|A| = 75930$, $|J| = 14651$, $|K| = 29267$



p_S-sredica za prag 0.75 v **Ct**(SN5)



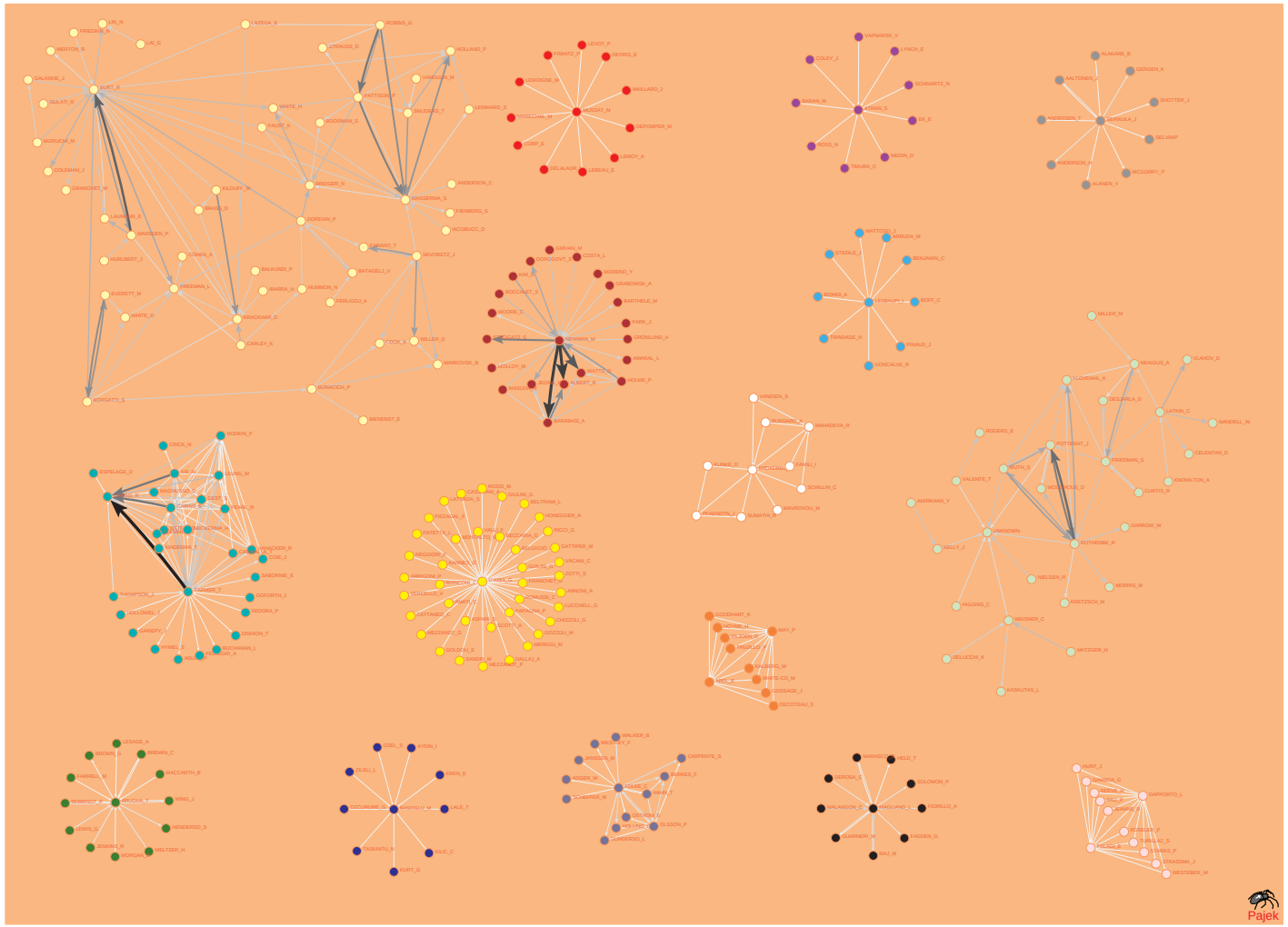
Omrežje sklicevanj med avtorji



$\mathbf{Ca} = \mathbf{AW} * \mathbf{Ci} * \mathbf{WA}$ je *omrežje sklicevanj med avtorji*. Utež ca_{ij} šteje kolikokrat se kako delo, ki ga je soustvaril avtor i, sklicuje na delo, ki ga je soustvaril avtor j.

Otoki v **Ca**(SN5)

Omrežje SN5 (2008): za "social network*" + članki z največ sklici nanje + okrog 100 raziskovalcev; $|W| = 193376$, $|C| = 7950$, $|A| = 75930$, $|J| = 14651$, $|K| = 29267$



(Sklicna) sklopljenost

V programu `WoS2Pajek` ima relacija sklicevanja **Ci** naslednji pomen $p\mathbf{Ci}q \equiv$ delo p se sklicuje na delo q. Zato je omrežje *(sklicne) sklopljenosti* **biCo** določeno z

$$\mathbf{biCo} = \mathbf{Ci} * \mathbf{Ci}^T$$

$bico_{pq} = \#$ del, na katera se sklicujeta obe deli p in q. $bico_{pq} = bico_{qp}$. Zopet so težave z deli z dolgimi seznami virov – posebej s preglednimi članki. Da bi omilili njihov vpliv, lahko uvedemo normalizirane mere kot je

$$\mathbf{biCon} = \frac{1}{2}(n(\mathbf{Ci}) * \mathbf{Ci}^T + \mathbf{Ci} * n(\mathbf{Ci})^T)$$

Velja $bicon_{pq} \in [0, 1]$ in $bicon_{pq} = bicon_{qp}$ (simetričnost). Velja še: $bicon_{pq} = 1$ ntk deli p in q se sklicujeta na ista dela.

Omrežje sosklicov in druga

Omrežje *sosklicov* **coCi** je določeno z

$$\mathbf{coCi} = \mathbf{Ci}^T * \mathbf{Ci}$$

$coci_{pq} = \#$ del, ki se sklicujejo hkrati na deli p in q. $coci_{pq} = coci_{qp}$.

Utež $w(a, p)$ v omrežju *avtorskih sklicev*

$$\mathbf{ACi} = \mathbf{AW} * \mathbf{Ci}$$

šteje kolikokrat se je avtor a sklical na delo p.

Omrežje *avtorskih sosklicov* dobimo z

$$\mathbf{ACo} = b(\mathbf{ACi}) * b(\mathbf{ACi})^T$$

Omrežje *Avtorske rabe gesel* pa z $\mathbf{AK} = \mathbf{AW} * \mathbf{WK}$.

Druga izpeljana omrežja

Sklicevanja med revijami

$$\mathbf{CJt} = \mathbf{WJ}^T * n(\mathbf{Ci}) * \mathbf{WJ}$$

Avtorjev prispevek reviji

$$\mathbf{AJn} = \mathbf{N}^T * \mathbf{WJ}$$

Prispevek avtorjev prve revije drugi reviji

$$\mathbf{JJr} = b(\mathbf{AJn})^T * \mathbf{AJn}$$

Prekrivanje in ohranjanje vsebine

V analizi omrežij sklicevanj se izkaže za koristno operacija $\mathcal{N}_A \rhd \mathcal{N}_B$ *prekrivanja* omrežja $\mathcal{N}_B$ z omrežjem $\mathcal{N}_A$.

Za omrežji $\mathcal{N}_A = (\mathcal{V}, \mathcal{A}_A, w_A)$ in $\mathcal{N}_B = (\mathcal{V}, \mathcal{A}_B)$ je $\mathcal{N}_A \rhd \mathcal{N}_B = \mathcal{N}_C = (\mathcal{V}, \mathcal{A}_C, w_C)$, kjer je $\mathcal{A}_C = \mathcal{A}_A \cap \mathcal{A}_B$ in $(u, v) \in \mathcal{A}_C \Rightarrow w_C(u, v) = w_A(u, v)$.

`Networks/Cross-Intersection/First`

Z uporabo prekrivanja lahko vpeljamo omrežje *ohranjanja vsebine* $\mathbf{CiK} = (\mathbf{WK} * \mathbf{WK}^T) \rhd \mathbf{Ci}$. Utež cik_{pq} = število skupnih gesel del p in q.

Razvrščanje omrežij

Bločni modeli

Posplošeni bločni modeli

Bločni modeli z omejitvami

Dvovrstni bločni modeli

Označeni grafi

Posplošitve

Razvrščanje z omejitvami

Analiza omrežij

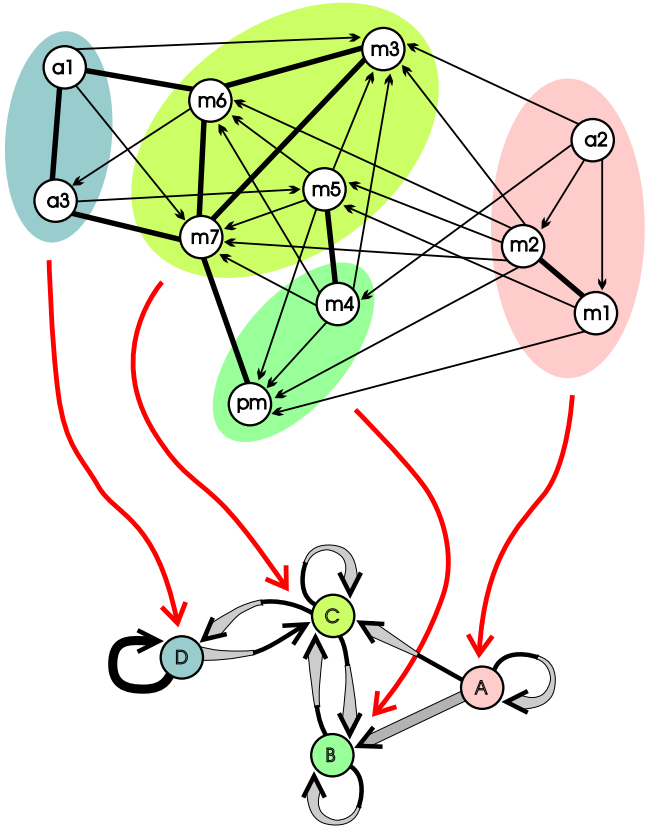
9. Razvrstitve in bločni modeli

Vladimir Batagelj

Magistrski program Uporabna statistika
Ljubljana, maj 2024

Kazalo

1 Razvrščanje omrežij
2 Bločni modeli
3 Posplošeni bločni modeli
4 Bločni modeli z omejitvami
5 Dvovrstni bločni modeli
6 Označeni grafi
7 Posplošitve
8 Razvrščanje z omejitvami

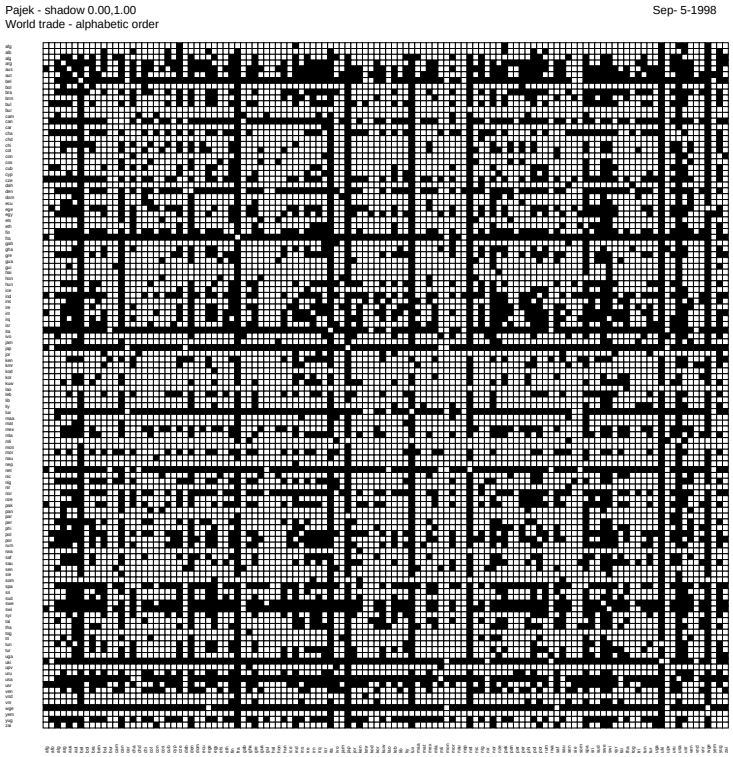


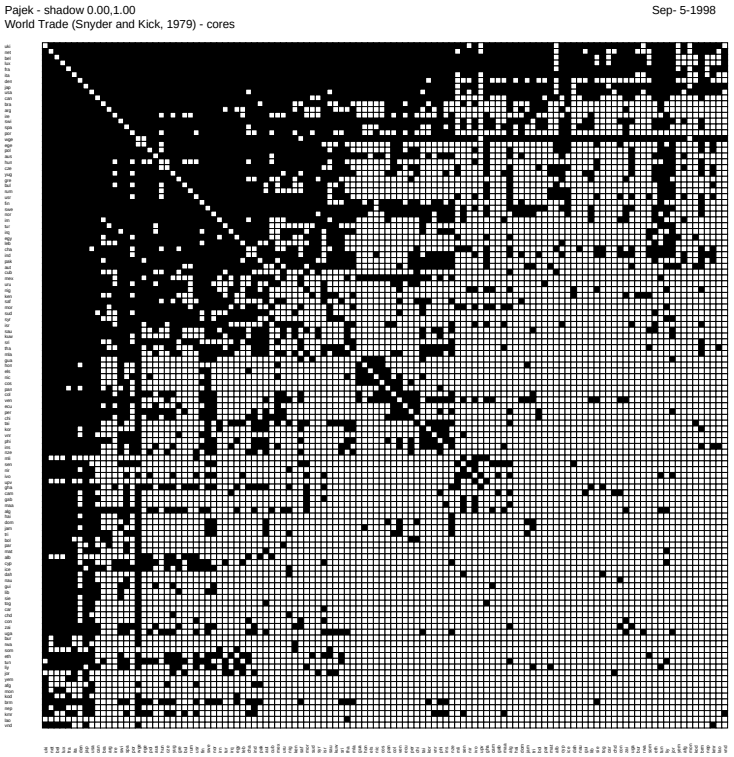
prof. Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si
prosojnice (PDF)

20. maj 2024 ob 18 : 02/ maj 2013

Preurejanja in bločni modeli

Snyder & Kickovo omrežje svetovne trgovine / $n = 118$, $m = 514$





Države urejene po abecedi (levo) urejenost iz razvrstitve (desno)

Preurejanje matrik

Dano matriko lahko preuredimo – določimo *urejenost* ali *permutacijo* njenih vrstic in stolpcev – na več načinov. Prikaz ustrezno preurejene matrike nam omogoča razkriti njeno zgradbo. Nekaj pristopov:

- urejenost po izbrani lastnosti: po šibkih/krepkih komponentah povezanosti; po stopnjah; po središčnih številih; po komponentah v središčnih slojih;
- urejenost glede na hierarhično razvrstitev.

Obstajajo tudi posebni postopki za preurejanje matrik: površčanje in kupčkanje (Murtagh), Cuthill-McKee-jev postopek, ...

```
Partition/Make Permutation
Vector/Make Permutation
Network/Create Permutation/Reverse Cuthill-McKee
Network/Create Permutation/Core+Degree
```

Površčanje (seriation)

Naj bo dano dvovrstno omrežje $\mathcal{N} = (\mathcal{U}, \mathcal{W}, R, w)$, $R \subseteq \mathcal{U} \times \mathcal{W}$, $w : R \to \mathbb{R}_0^+$.
Preslikavi $\rho : \mathcal{U} \to 1..n$, $\sigma : \mathcal{W} \to 1..m$ naj bosta bijekciji (permutaciji).
Za vrstično enoto $\mathrm{X} \in \mathcal{U}$ vpeljemo vrstično vsoto $r(\mathrm{X})$; za stolpčno enoto $\mathrm{Y} \in \mathcal{W}$ pa stolpčno vsoto $c(\mathrm{Y})$

$$r(\mathrm{X}) = \sum_{\mathrm{Y} \in R(\mathrm{X})} w(\mathrm{X}, \mathrm{Y}) \quad \text{in} \quad c(\mathrm{Y}) = \sum_{\mathrm{X} \in R^{-1}(\mathrm{Y})} w(\mathrm{X}, \mathrm{Y})$$

ter naprej vrstične uteži $p(\mathrm{X})$ in stolpčne uteži $q(\mathrm{Y})$

$$p(\mathrm{X}) = \frac{1}{r(\mathrm{X})} \sum_{\mathrm{Y} \in R(\mathrm{X})} \sigma(\mathrm{Y}) w(\mathrm{X}, \mathrm{Y}), \quad q(\mathrm{Y}) = \frac{1}{c(\mathrm{Y})} \sum_{\mathrm{X} \in R^{-1}(\mathrm{Y})} \rho(\mathrm{X}) w(\mathrm{X}, \mathrm{Y})$$

Če je $r(\mathrm{X}) = 0$, je tudi $p(\mathrm{X}) = 0$; če je $c(\mathrm{X}) = 0$, je $q(\mathrm{X}) = 0$.

... postopek površčanja

določi (slučajno) σ ;
repeat
  določi vrstične uteži $p(\mathrm{X})$, $\mathrm{X} \in \mathcal{U}$; $\rho :=$ sort_decreasing$(\mathcal{U}, p)$;
  določi stolpčne uteži $q(\mathrm{Y})$, $\mathrm{Y} \in \mathcal{W}$; $\sigma :=$ sort_decreasing$(\mathcal{W}, q)$;
until urejenosti se ustalita (ali število korakov doseže mejo)

Za enovrstna omrežja je $\rho = \sigma$ in za utež enote uporabimo $p(\mathrm{X}) + q(\mathrm{X})$.

`Operations/Network+Permutation/Numbering*/Seriation`

Kupčkanje (clumping)

Kupčkanje poskuša preurediti enote tako, da mera *nakopičenosti* doseže čim večjo vrednost. Urejenost se določa postopno s požrešnim dodajanjem nove enote na najustreznejše mesto v tekoči urejenosti določeni s seznamom, ki vsebuje k enot in na obeh koncih še dodatni 'stražarski' enoti $[\mathrm{X}_0 = \mathbf{0}, \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_k, \mathbf{0} = \mathrm{X}_{k+1}]$.
Če vstavimo vrstično enoto X v seznam za členom X_i, to ustvari vrstično nakopičenost

$$Q(i) = \sum_{\mathrm{Y} \in R(\mathrm{X}) \cap (R(\mathrm{X}_i) \cup R(\mathrm{X}_{i+1}))} w(\mathrm{X}, \mathrm{Y})(w(\mathrm{X}_i, \mathrm{Y}) + w(\mathrm{X}_{i+1}, \mathrm{Y}))$$

Če $(\mathrm{X}, \mathrm{Y}) \notin R$, je $w(\mathrm{X}, \mathrm{Y}) = 0$.
Na enak način (obratno omrežje) lahko preuredimo tudi stolpčne enote.
Za enovrstna omrežja uporabimo sestavljeno mero – vsoto vrstične in stolpčne nakopičenosti.

... postopek kupčkanja

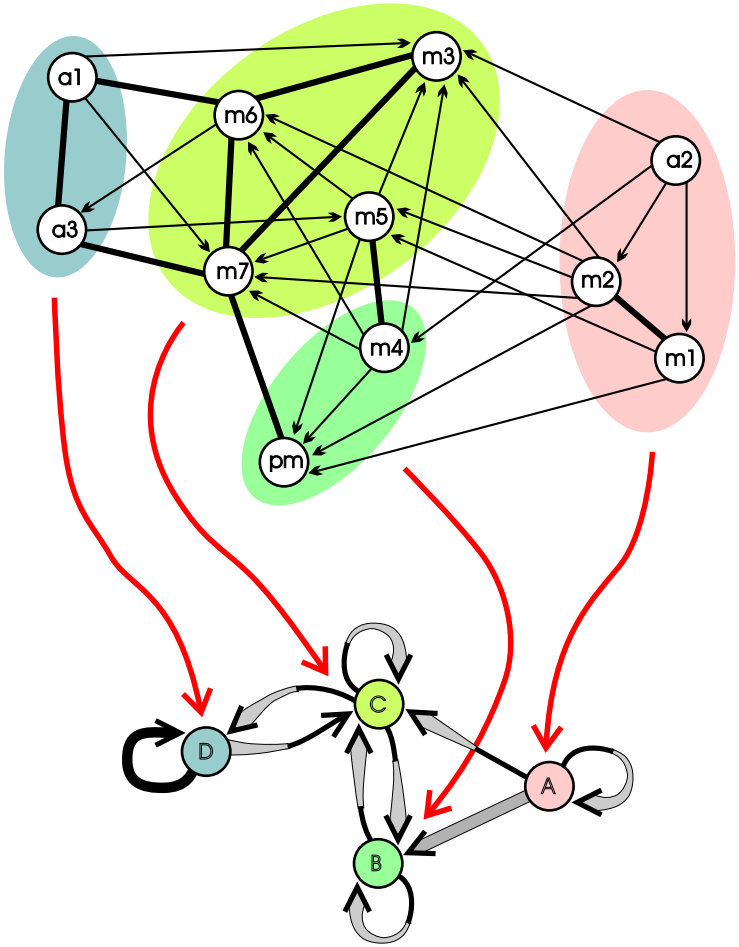
izberi (slučajno) $\mathsf{X} \in \mathcal{U}$; $\mathbf{S} := \mathcal{U} \setminus \{\mathsf{X}\}$
order $:= [\mathbf{0}, \mathsf{X}, \mathbf{0}]$; $k := 1$;
while $\mathbf{S} \neq \emptyset$ **do begin**
  izberi $\mathsf{X} \in \mathbf{S}$; $\mathbf{S} := \mathbf{S} \setminus \{\mathsf{X}\}$;
  for $i := 0$ **to** k **do** določi $Q(i)$;
  $j := \operatorname{argmax}_i Q(i)$;
  vstavi enoto X v urejenost *order* za enoto X_j;
  $k := k + 1$
end;

`Operations/Network+Permutation/Numbering*/Clumping`

Bločni modeli kot problem razvrščanja

Cilj *bločnega modeliranja* je skrčiti obsežno, morda neskladno omrežje v manjše, razumljivejše omrežje, ki ustrezno povzema zgradbo izvornega omrežja in ponuja njeno razlago.

Enote/vozlišča omrežja so pri bločnem modeliranju razvrščene v skupine glede na neko *smiselno* enakovrednost.



Skupina, razvrstitev, blok

Glavni cilj bločnega modeliranja je določiti v danem omrežju $\mathcal{N} = (\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$, *skupine* (razrede) enot, ki imajo enake ali podobne strukturne značilnosti glede na R – enote iz posamezne skupine so enako ali podobno povezane z enotami drugih skupin.
Skupine sestavljajo *razvrstitev* $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ ki je *razbitje* množice enot $\mathcal{U}$.
Kot vemo, vsako razbitje določa neko enakovrednost (in obratno).
Označimo s $\sim$ enakovrednost določeno z razbitjem $\mathbf{C}$.
Razvrstitev $\mathbf{C}$ razbije tudi relacijo R na *bloke*

$$R(C_i, C_j) = R \cap C_i \times C_j$$

Vsak tak blok sestavljajo enote iz skupin C_i in C_j in vse povezave, ki vodijo iz skupine C_i v skupino C_j. Če je $i = j$, bloku $R(C_i, C_i)$ rečemo *diagonalni* blok.

Strukturna in regularna enakovrednost

Večina vrst enakovrednosti temelji na ne enem od naslednjih dveh pristopov/pogledov (Faust, 1988):

- enakovredni enoti sta na enak način povezani (imata enak vzorec povezanosti) do **istih** sosedov;
- enakovredni enoti sta na enak ali podoben način povezani do (lahko) **različnih** sosedov.

Primer prve vrste enakovrednostni je strukturna enakovrednost; primer druge vrste pa regularna enakovrednost.

Strukturna enakovrednost

Enoti sta enakovredni, če sta povezani z ostalim omrežjem na *enak* način (Lorrain in White, 1971). Natančneje:
Enoti X in Y sta *strukturno enakovredni*, kar zapišemo $X \equiv Y$, ntk. ki je permutacija (premena) $\pi = (X\,Y)$ avtomorfizem of relacije R (Borgatti in Everett, 1992).
Drugače povedano, enoti X in Y sta strukturno enakovredni ntk.:

s1.	$XRY \Leftrightarrow YRX$	s3.	$\forall Z \in \mathcal{U} \setminus \{X, Y\} : (XRZ \Leftrightarrow YRZ)$
s2.	$XRX \Leftrightarrow YRY$	s4.	$\forall Z \in \mathcal{U} \setminus \{X, Y\} : (ZRX \Leftrightarrow ZRY)$

... strukturna enakovrednost

Za strukturno enakovrednost sta edina mogoča bloka prazni in polni blok (z morebitnim preklopom na diagonali za diagonalne bloke):

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{matrix}$$

Regularna enakovrednost

V dejanskih omrežjih je le malo enot strukturno enakovrednih. Skupni imenovalec vsem poskusom njene posplošitve je, da sta enoti enakovredni, če sta podobno povezani z drugimi enakovrednimi enotami.

White in Reitz (1983): Enakovrednost $\approx$ on $\mathcal{U}$ je *regularna enakovrednost* na omrežju $\mathcal{N} = (\mathcal{U}, R)$ ntk. za vse enote $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \in \mathcal{U}$, iz $\mathrm{X} \approx \mathrm{Y}$ izhaja tudi

$$\begin{array}{ll} \text{R1.} & \mathrm{X}R\mathrm{Z} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{Y}R\mathrm{W} \wedge \mathrm{W} \approx \mathrm{Z}) \\ \text{R2.} & \mathrm{Z}R\mathrm{X} \Rightarrow \exists \mathrm{W} \in \mathcal{U} : (\mathrm{W}R\mathrm{Y} \wedge \mathrm{W} \approx \mathrm{Z}) \end{array}$$

Drug pogled na regularno enakovrednost temelji na barvanjih vozlišč (Everett, Borgatti 1996). Naj bo $b : \mathcal{U} \to B$ barvanje vozlišč omrežja. Enakovrednost $\approx$ je *regularna* ntk. za vsak par $\mathrm{X}, \mathrm{Y} \in \mathcal{U}$ velja

$$\mathrm{X} \approx \mathrm{Y} \Leftrightarrow (b(\mathrm{X}) = b(\mathrm{Y})) \wedge (b(N(\mathrm{X})) = b(N(\mathrm{Y})))$$

... regularna enakovrednost

Theorem (Batagelj, Doreian, Ferligoj, 1992)

Naj bo $\mathbf{C} = \{C_i\}$ *razbitje, ki določa regularno enakovrednost* $\approx$ *na omrežju* $\mathcal{N} = (\mathcal{U}, R)$. *Tedaj je vsak blok* $R(C_u, C_v)$ *ali prazen ali pa ima lastnost da obstaja vsaj ena 1 v vsaki njegovi vrstici in vsakem njegovem stolpcu. Obratno, če so za dano razbitje* $\mathbf{C}$ *vsi bloki teh dveh vrst, je ustrezna enakovrednost regularna.*

Bloki regularne enakovrednosti so prazni ali 1-pokriti.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{matrix}$$

Določanje bločnih modelov

The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of *clustering problem* that can be formulated as an optimization problem (Φ, P) as follows:

Determine the clustering $\mathbf{C}^\star \in \Phi$ *for which*

$$P(\mathbf{C}^\star) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

where Φ is the set of *feasible clusterings* and P is a *criterion function*. Since the set of units $\mathcal{U}$ is finite, the set of feasible clusterings is also finite. Therefore the set $\mathrm{Min}(\Phi, P)$ of all solutions of the problem (optimal clusterings) is not empty.

- *posredni* pristop – kriterijsko funkcijo izrazimo na osnovi mere različnosti med vozlišči;
- *premi* pristop – kriterijska funkcija meri odstopanje dejanske razvrstitve vozlišč omrežja od ustreznega idealnega stanja za

Posredni pristop

OMREŽJE

R

omrežna relacija

OPISI ENOT

Q

matrika poti

trojice

orbite

MATRIKA RAZLIČNOSTI

D

OBIČAJNI POSTOPKI RAZVRŠČANJA

postopki združevanja,

postopki prestavljanj, metoda voditeljev, itd.

Dissimilarities

The property $t : \mathcal{U} \to \mathbb{R}$ is *structural property* if, for every automorphism φ, of the relation R, and every unit $x \in \mathcal{U}$, it holds that $t(x) = t(\varphi(x))$. Some examples of a structural property include

$$\begin{aligned} t(u) &= \text{the } \textit{degree} \text{ of unit } u; \\ t(u) &= \text{number of units at } \textit{distance } d \text{ from the unit } u; \\ t(u) &= \text{number of } \textit{triad}\text{s of type } x \text{ at the unit } u. \end{aligned}$$

Centrality measures are further examples of structural properties. We can define the description of the unit u as $[u] = [t_1(u), t_2(u), \ldots, t_m(u)]$. As a simple example, t_1 could be *degree* centrality, t_2 could be *closeness* centrality and t_3 could be *betweenness* centrality. The dissimilarity between units u and v could be defined as $d(u, v) = D([u], [v])$ where D is some (standard) dissimilarity between real vectors. In the simple example, D could be the *Euclidean* distance between the centrality profiles.

Dissimilarities based on matrices

We consider the following list of dissimilarities between units x_i and x_j where the description of the unit consists of the row and column of the property matrix $\mathbf{Q} = [q_{ij}]$. We take as units the rows of the matrix

$$\mathbf{X} = [\mathbf{Q} \mid \mathbf{Q}^T]$$

... Dissimilarities

Manhattan distance: $d_m(x_i, x_j) = \sum_{s=1}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$

Euclidean distance:

$$d_E(x_i, x_j) = \sqrt{\sum_{s=1}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$

Truncated Manhattan distance:

$$d_s(x_i, x_j) = \sum_{\substack{s=1 \\ s \neq i,j}}^{n} (|q_{is} - q_{js}| + |q_{si} - q_{sj}|)$$

Truncated Euclidean distance (Faust, 1988):

$$d_S(x_i, x_j) = \sqrt{\sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((q_{is} - q_{js})^2 + (q_{si} - q_{sj})^2)}$$

... Dissimilarities

Corrected Manhattan-like dissimilarity ($p \geq 0$):

$$d_c(p)(x_i, x_j) = d_s(x_i, x_j) + p \cdot (|q_{ii} - q_{jj}| + |q_{ij} - q_{ji}|)$$

Corrected Euclidean-like dissimilarity (Burt and Minor, 1983):

$$d_e(p)(x_i, x_j) = \sqrt{d_S(x_i, x_j)^2 + p \cdot ((q_{ii} - q_{jj})^2 + (q_{ij} - q_{ji})^2)}$$

Corrected dissimilarity:

$$d_C(p)(x_i, x_j) = \sqrt{d_c(p)(x_i, x_j)}$$

The parameter, p, can take any positive value. Typically, $p = 1$ or $p = 2$, where these values count the number of times the corresponding diagonal pairs are counted.

`Operations/Network+Cluster/Dissimilarity*/Network based`

. . . Dissimilarities

It is easy to verify that all expressions from the list define a dissimilarity (i.e. that $d(x, y) \geq 0$; $d(x, x) = 0$; and $d(x, y) = d(y, x)$). Each of the dissimilarities from the list can be assessed to see whether or not it is also a distance: $d(x, y) = 0 \Rightarrow x = y$ and $d(x, y) + d(y, z) \geq d(x, z)$.
The dissimilarity measure d is *compatible* with a considered equivalence $\sim$ if for each pair of units holds

$$X_i \sim X_j \Leftrightarrow d(X_i, X_j) = 0$$

Not all dissimilarity measures typically used are compatible with structural equivalence. For example, the *corrected Euclidean-like dissimilarity* is compatible with structural equivalence.
The indirect clustering approach does not seem suitable for establishing clusterings in terms of regular equivalence since there is no evident way how to construct a compatible (dis)similarity measure.

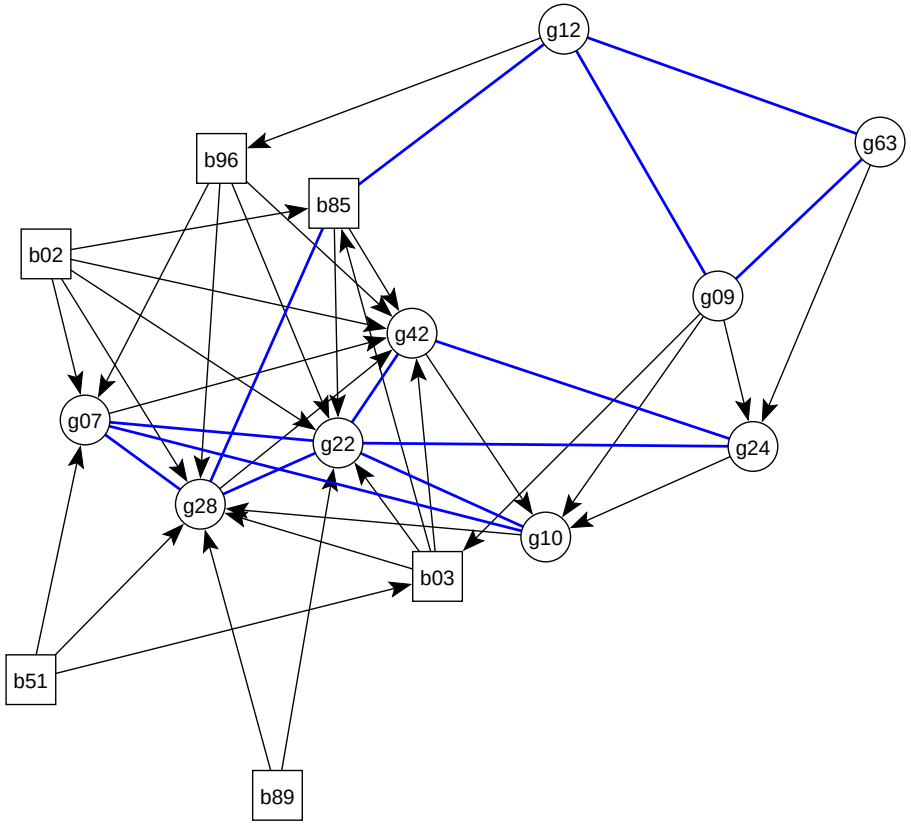
Example: Support network among informatics students

The analyzed network consists of social support exchange relation among fifteen students of the Social Science Informatics fourth year class (2002/2003) at the Faculty of Social Sciences, University of Ljubljana. Interviews were conducted in October 2002.

Support relation among students was identified by the following question:

> *Introduction: You have done several exams since you are in the second class now. Students usually borrow studying material from their colleagues.*
>
> *Enumerate (list) the names of your colleagues that you have most often borrowed studying material from. (The number of listed persons is not limited.)*

Class network

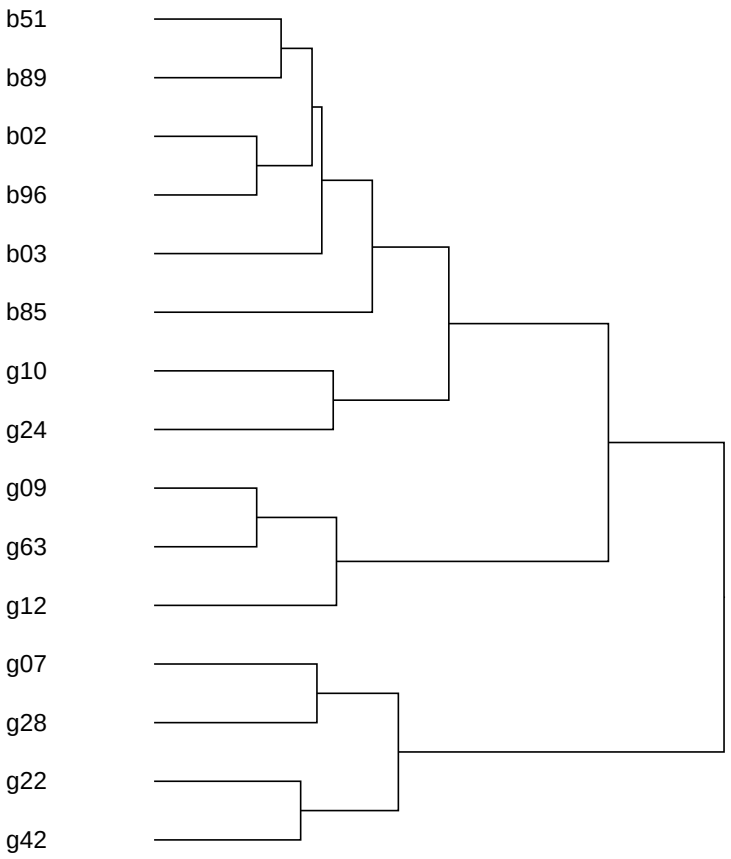


`class.net`

Vertices represent students in the class; circles – girls, squares – boys. Opposite pairs of arcs are replaced by edges.

```
Cluster/Create Complete Cluster
Operations/Network+Cluster/Dissimilarity*/Network based/
```

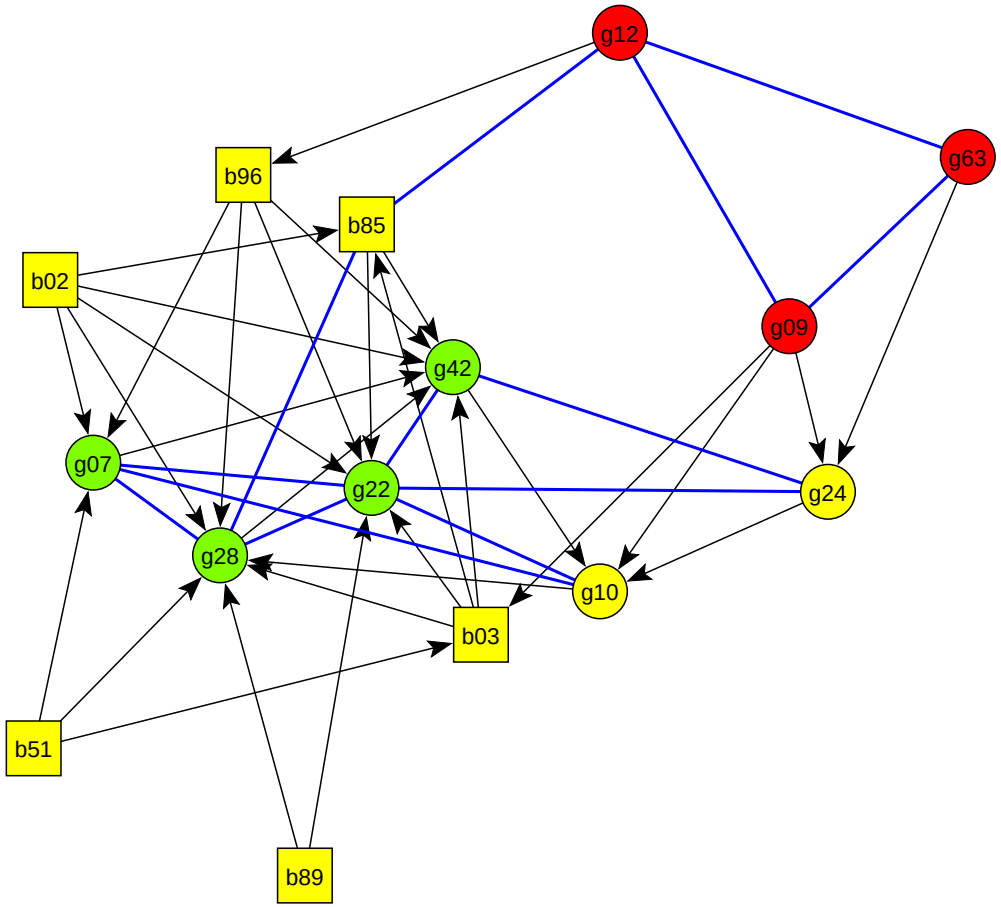
Indirect approach



Using *Corrected Euclidean-like dissimilarity* and *Ward clustering method* we obtain the following dendrogram.
From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.
If we select 3 clusters we get the partition **C**.

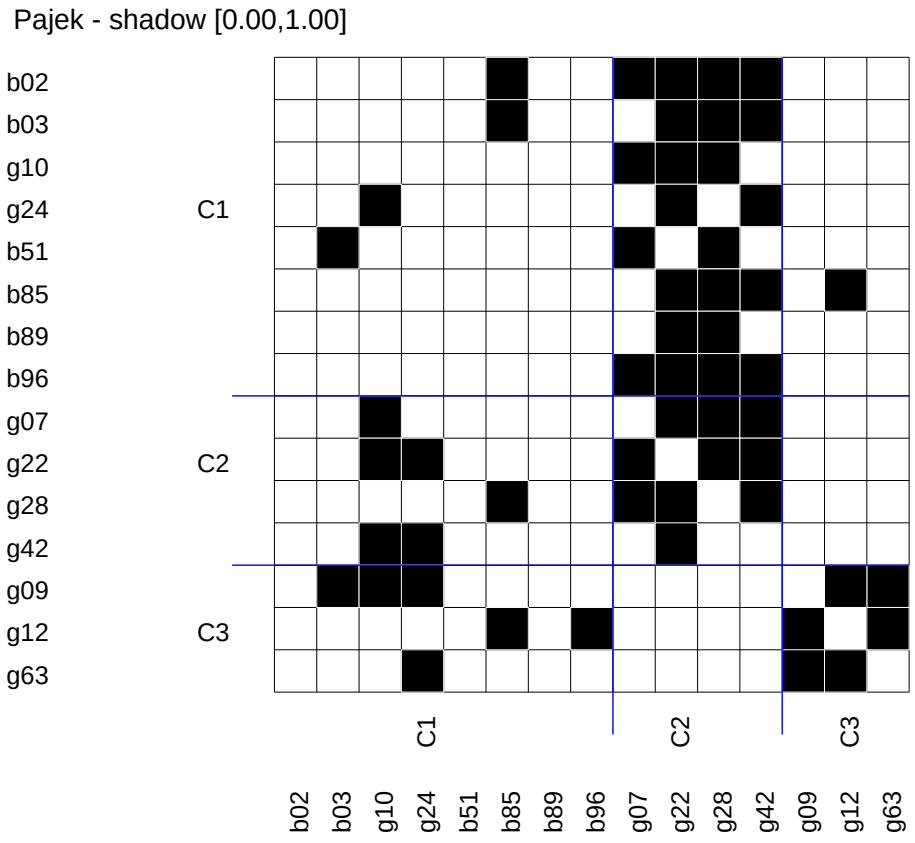
$$\mathbf{C} = \{\{b51, b89, b02, b96, b03, b85, g10, g24\}, \{g09, g63, g12\}, \{g07, g28, g22, g42\}\}$$

Partition in 3 clusters



On the picture, vertices in the same cluster are of the same color.

Matrix



The partition can be used also to reorder rows and columns of the matrix representing the network. Clusters are divided using blue vertical and horizontal lines.

```
determine clustering in the hierarchy
Hierarchy/Make Permutation
Hierarchy/Make Partition
File/Network/Export as Matrix to EPS/Using Permutation
```

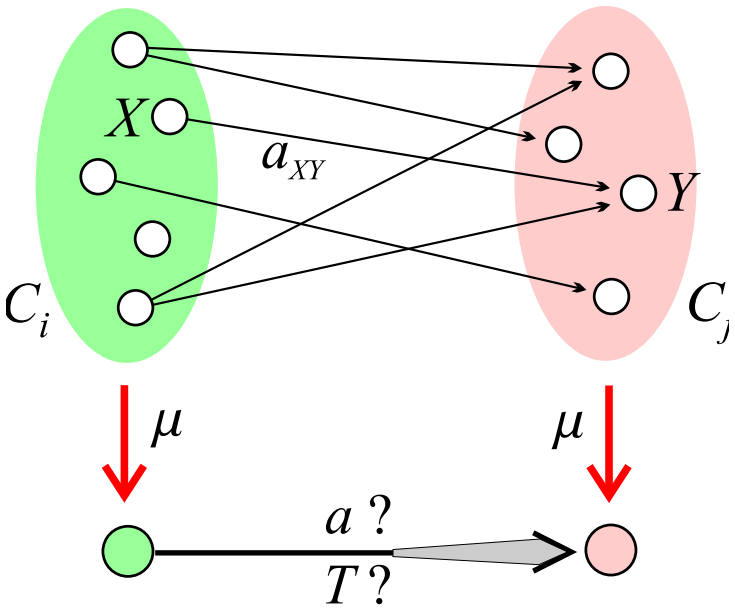
Direct Approach and Generalized Blockmodeling

The second possibility for solving the blockmodeling problem is to construct an appropriate criterion function directly and then use a local optimization algorithm to obtain a 'good' clustering solution.
Criterion function $P(\mathbf{C})$ has to be *sensitive* to considered equivalence:

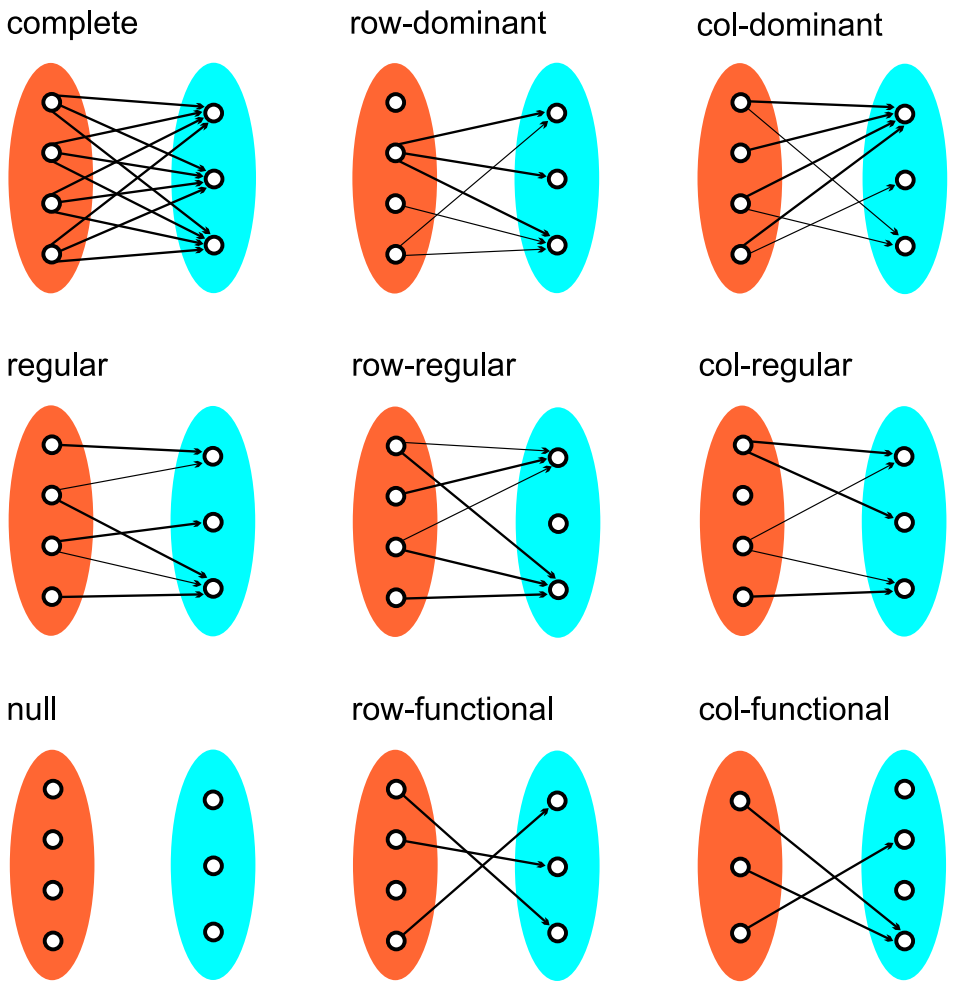
$$P(\mathbf{C}) = 0 \Leftrightarrow \mathbf{C} \text{ defines considered equivalence.}$$

Generalized Blockmodeling

A *blockmodel* consists of structures obtained by identifying all units from the same cluster of the clustering **C**. For an exact definition of a blockmodel we have to be precise also about which blocks produce an arc in the *reduced graph* and which do not, and of what *type*. Some types of connections are presented in the figure on the next slide. The reduced graph can be represented by relational matrix, called also *image matrix*.



Block Types


complete
row-dominant
col-dominant
regular
row-regular
col-regular
null
row-functional
col-functional

Generalized equivalence / Block Types

			Y		
X	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

complete

			Y		
X	0	1	0	0	0
	1	1	1	1	1
	0	0	0	0	0
	0	0	0	1	0

row-dominant

			Y		
X	0	0	1	0	0
	0	0	1	1	0
	1	1	1	0	0
	0	0	1	0	1

col-dominant

			Y		
X	0	1	0	0	0
	1	0	1	1	0
	0	0	1	0	1
	1	1	0	0	0

regular

			Y		
X	0	1	0	0	0
	0	1	1	0	0
	1	0	1	0	0
	0	1	0	0	1

row-regular

			Y		
X	0	1	0	1	0
	1	0	1	0	0
	1	1	0	1	1
	0	0	0	0	0

col-regular

			Y		
X	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0

null

			Y		
X	0	0	0	1	0
	0	0	1	0	0
	1	0	0	0	0
	0	0	0	1	0

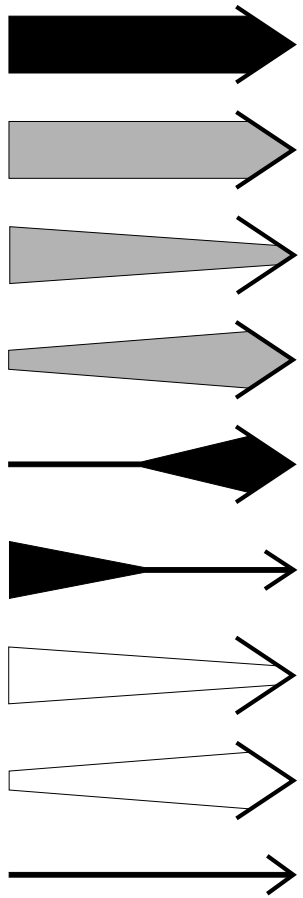
row-functional

		Y		
X	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	0
	0	0	0	1

col-functional

Characterizations of Types of Blocks

null	nul	all 0 *	
complete	com	all 1 *	
regular	reg	1-covered rows and columns	
row-regular	rre	each row is 1-covered	
col-regular	cre	each column is 1 -covered	
row-dominant	rdo	$\exists$ all 1 row *	
col-dominant	cdo	$\exists$ all 1 column *	
row-functional	rfn	$\exists!$ one 1 in each row	
col-functional	cfn	$\exists!$ one 1 in each column	
non-null	one	$\exists$ at least one 1	



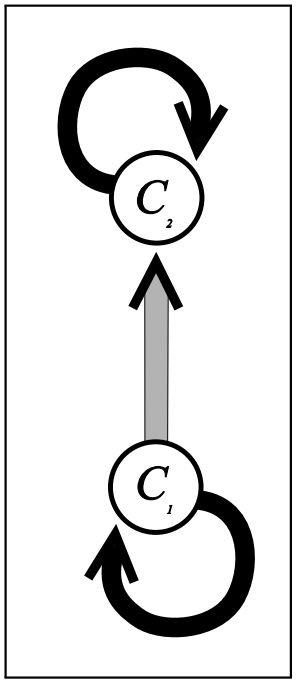
* except this may be diagonal

A block is *symmetric* iff $\forall X, Y \subset C_i \times C_j : (XRY \Leftrightarrow YRX)$

Block Types and Matrices

$$
\begin{array}{cccc|cccc}
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

	C_1	C_2
C_1	complete	regular
C_2	null	complete



Formalization of blockmodeling

Let V be a set of positions or images of clusters of units. Let $\mu : \mathcal{U} \to V$ denote a mapping which maps each unit to its position. The cluster of units $C(t)$ with the same position $t \in V$ is

$$C(t) = \mu^{-1}(t) = \{X \in \mathcal{U} : \mu(X) = t\}$$

Therefore

$$\mathbf{C}(\mu) = \{C(t) : t \in V\}$$

is a partition (clustering) of the set of units $\mathcal{U}$.

Blockmodel

A *blockmodel* is an ordered sextuple $\mathcal{M} = (V, K, \mathcal{T}, Q, \pi, \alpha)$ where:

- V is a set of *types of units* (images or representatives of classes);
- $K \subseteq V \times V$ is a set of *connections*;
- $\mathcal{T}$ is a set of predicates used to describe the *types of connections* between different classes (clusters, groups, types of units) in a network. We assume that $\mathrm{nul} \in \mathcal{T}$. A mapping $\pi : K \to \mathcal{T} \setminus \{\mathrm{nul}\}$ assigns predicates to connections;
- Q is a set of *averaging rules*. A mapping $\alpha : K \to Q$ determines rules for computing values of connections.

A (surjective) mapping $\mu : \mathcal{U} \to V$ determines a blockmodel $\mathcal{M}$ of network $\mathcal{N}$ iff it satisfies the conditions: $\forall (t, w) \in K : \pi(t, w)(C(t), C(w))$ and $\forall (t, w) \in V \times V \setminus K : \mathrm{nul}(C(t), C(w))$.

Equivalences

Let $\sim$ be an equivalence relation over $\mathcal{U}$ and $[\mathrm{X}] = \{\mathrm{Y} \in \mathcal{U} : \mathrm{X} \sim \mathrm{Y}\}$. We say that $\sim$ is *compatible* with $\mathcal{T}$ over a network $\mathcal{N}$ iff

$$\forall \mathrm{X}, \mathrm{Y} \in \mathcal{U} \exists T \in \mathcal{T} : T([\mathrm{X}], [\mathrm{Y}]).$$

It is easy to verify that the notion of compatibility for $\mathcal{T} = \{\mathrm{nul}, \mathrm{reg}\}$ reduces to the usual definition of regular equivalence (White and Reitz 1983). Similarly, compatibility for $\mathcal{T} = \{\mathrm{nul}, \mathrm{com}\}$ reduces to structural equivalence (Lorrain and White 1971).
For a compatible equivalence $\sim$ the mapping $\mu \colon \mathrm{X} \mapsto [\mathrm{X}]$ determines a blockmodel with $V = \mathcal{U}/\sim$.
The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of **clustering problem** that can be formulated as an optimization problem.

Criterion function

One of the possible ways of constructing a criterion function that directly reflects the considered equivalence is to measure the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered equivalence.
Given a clustering $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$, let $\mathcal{B}(C_u, C_v)$ denote the set of all ideal blocks corresponding to block $R(C_u, C_v)$. Then the global error of clustering $\mathbf{C}$ can be expressed as

$$P(\mathbf{C}) = \sum_{C_u, C_v \in \mathbf{C}} \min_{B \in \mathcal{B}(C_u, C_v)} d(R(C_u, C_v), B)$$

where the term $d(R(C_u, C_v), B)$ measures the difference (error) between the block $R(C_u, C_v)$ and the ideal block B. d is constructed on the basis of characterizations of types of blocks. The function d has to be compatible with the selected type of equivalence.

... criterion function

For example, for structural equivalence, the term $d(R(C_u, C_v), B)$ can be expressed, for non-diagonal blocks, as

$$d(R(C_u, C_v), B) = \sum_{X \in C_u, Y \in C_v} |r_{XY} - b_{XY}|.$$

where r_{XY} is the observed tie and b_{XY} is the corresponding value in an ideal block. This criterion function counts the number of 1s in erstwhile null blocks and the number of 0s in otherwise complete blocks. These two types of inconsistencies can be weighted differently.
Determining the block error, we also determine the type of the best fitting ideal block (the types are ordered).
The criterion function $P(\mathbf{C})$ is *sensitive* iff $P(\mathbf{C}) = 0 \Leftrightarrow \mu$ (determined by $\mathbf{C}$) is an exact blockmodeling. For all presented block types sensitive criterion functions can be constructed (Batagelj, 1997).

Deviations Measures for Types of Blocks

We can efficiently test whether a block $R(X, Y)$ is of the type T by making use of the characterizations of block types. On this basis we can construct the corresponding deviation measures. The quantities used in the expressions for deviations have the following meaning:

s_t	– total block sum = # of 1s in a block,
n_r	= card(X) – # of rows in a block,
n_c	= card(Y)– # of columns in a block,
p_r	– # of non-null rows in a block,
p_c	– # of non-null columns in a block,
m_r	– maximal row-sum,
m_c	– maximal column-sum,
s_d	– diagonal block sum = # of 1s on a diagonal,
d	– diagonal error = $\min(s_d, n_r - s_d)$.

Throughout the number of elements in a block is $n_r n_c$.

... Deviations Measures for Types of Blocks

Connection	$\delta(X, Y; T)$	
null	s_t	nondiagonal
	$s_t + d - s_d$	diagonal
complete	$n_r n_c - s_t$	nondiagonal
	$n_r n_c - s_t + d + s_d - n_r$	diagonal
row-dominant	$(n_c - m_r - 1)n_r$	diagonal, $s_d = 0$
	$(n_c - m_r)n_r$	otherwise
col-dominant	$(n_r - m_c - 1)n_c$	diagonal, $s_d = 0$
	$(n_r - m_c)n_c$	otherwise
row-regular	$(n_r - p_r)n_c$	
col-regular	$(n_c - p_c)n_r$	
regular	$(n_c - p_c)n_r + (n_r - p_r)p_c$	
row-functional	$s_t - p_r + (n_r - p_r)n_c$	
col-functional	$s_t - p_c + (n_c - p_c)n_r$	
density γ	$\max(0, \gamma n_r n_c - s_t)$	

For the null, complete, row-dominant and column-dominant blocks it is necessary to distinguish diagonal blocks and non-diagonal blocks.

Solving the blockmodeling problem

The obtained optimization problem can be solved by local optimization. Once a partitioning μ and types of connection π are determined, we can also compute the values of connections by using averaging rules.

Benefits from Optimization Approach

- *ordinary / inductive blockmodeling*: Given a network $\mathcal{N}$ and set of types of connection $\mathcal{T}$, determine the model $\mathcal{M}$;
- *evaluation of the quality of a model, comparing different models, analyzing the evolution of a network* (Sampson data, Doreian and Mrvar 1996): Given a network $\mathcal{N}$, a model $\mathcal{M}$, and blockmodeling μ, compute the corresponding criterion function;
- *model fitting / deductive blockmodeling*: Given a network $\mathcal{N}$, set of types $\mathcal{T}$, and a family of models, determine μ which minimizes the criterion function (Batagelj, Ferligoj, Doreian, 1998).
- we can fit the network to a partial model and analyze the residual afterward;
- we can also introduce different constraints on the model, for example: units X and Y are of the same type; or, types of units X and Y are not connected; ...

Pre-specified blockmodeling

In the previous slides the inductive approaches for establishing blockmodels for a set of social relations defined over a set of units were discussed. Some form of equivalence is specified and clusterings are sought that are consistent with a specified equivalence.
Another view of blockmodeling is deductive in the sense of starting with a blockmodel that is specified in terms of substance prior to an analysis.
In this case given a network, set of types of ideal blocks, and a reduced model, a solution (a clustering) can be determined which minimizes the criterion function.

Pre-Specified Blockmodels

The pre-specified blockmodeling starts with a blockmodel specified, in terms of substance, **prior to an analysis**. Given a network, a set of ideal blocks is selected, a family of reduced models is formulated, and partitions are established by minimizing the criterion function.
The basic types of models are:

*	*
*	0

center - periphery

*	0
*	*

hierarchy

*	0
0	*

clustering

0	*
*	0

bipartition

Prespecified blockmodeling example

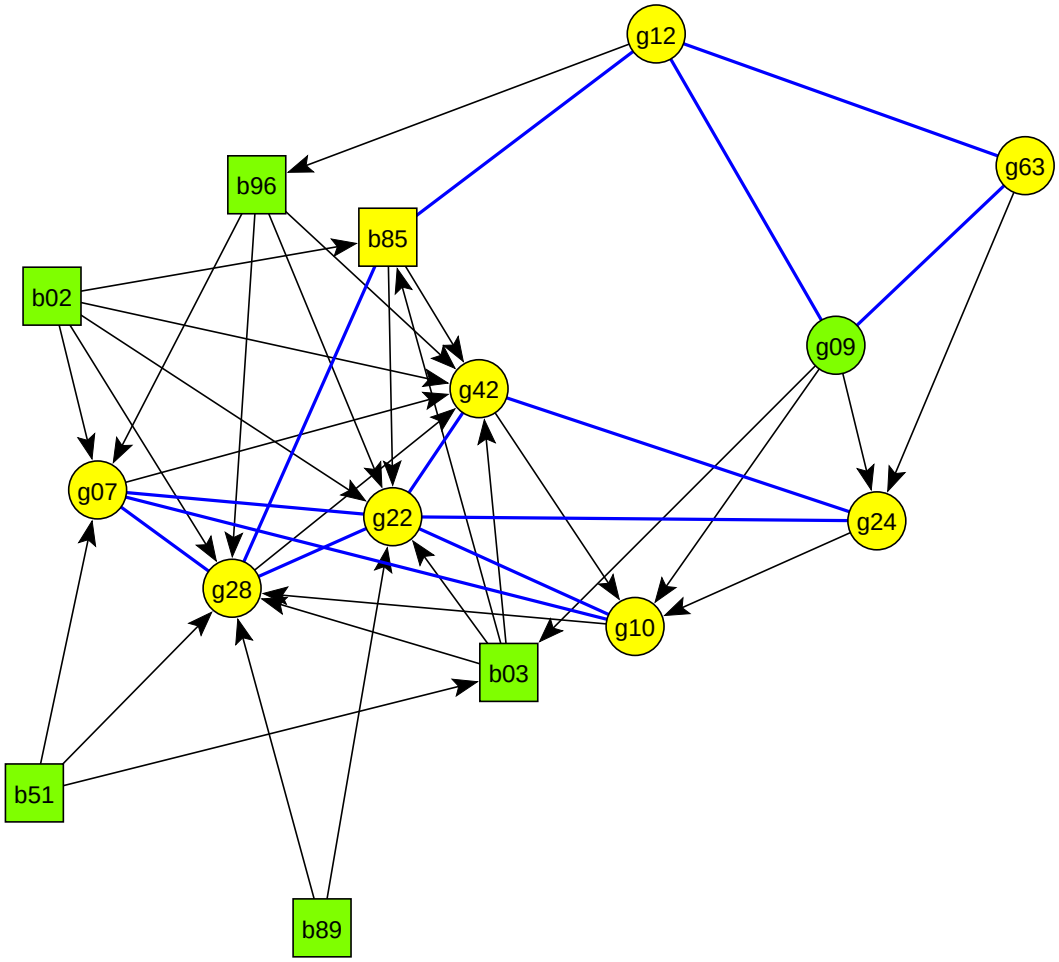
We expect that center-periphery model exists in the network: some students having good studying material, some not.
Prespecified blockmodel: (com/complete, reg/regular, -/null block)

	1	2
1	[com reg]	-
2	[com reg]	-

Using local optimization we get the partition:

$$\mathbf{C} = \{\{b02, b03, b51, b85, b89, b96, g09\}, \{g07, g10, g12, g22, g24, g28, g42, g63\}\}$$

2 clusters solution



Model

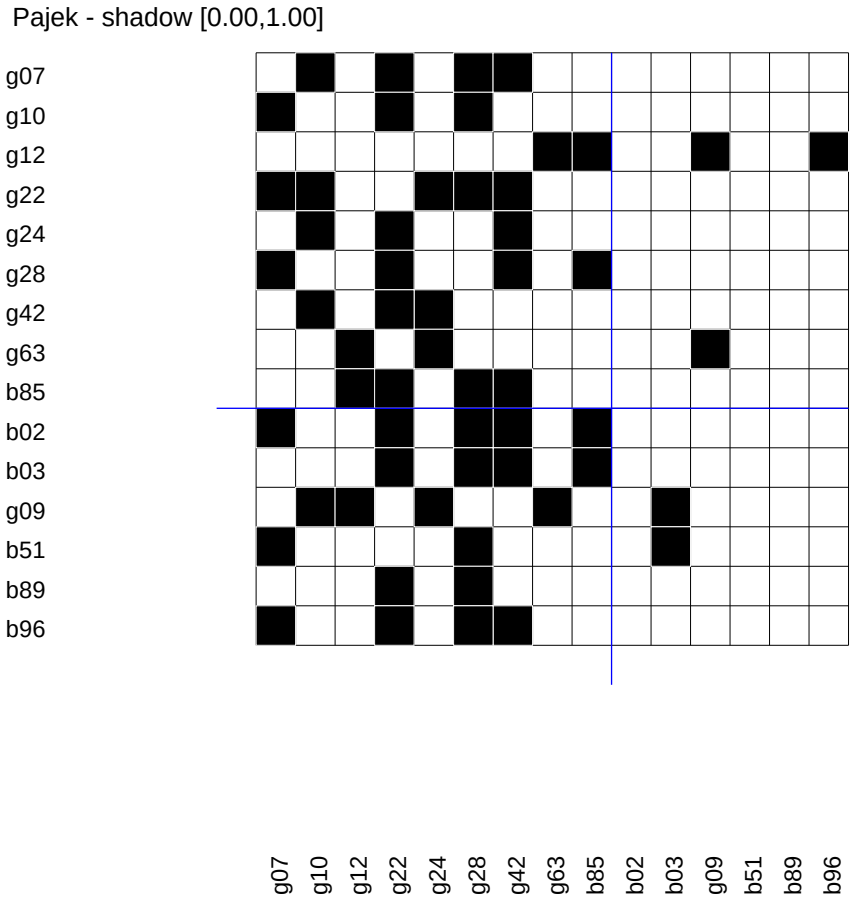


Image and Error Matrices:

	1	2
1	reg	-
2	reg	-

	1	2
1	0	3
2	0	2

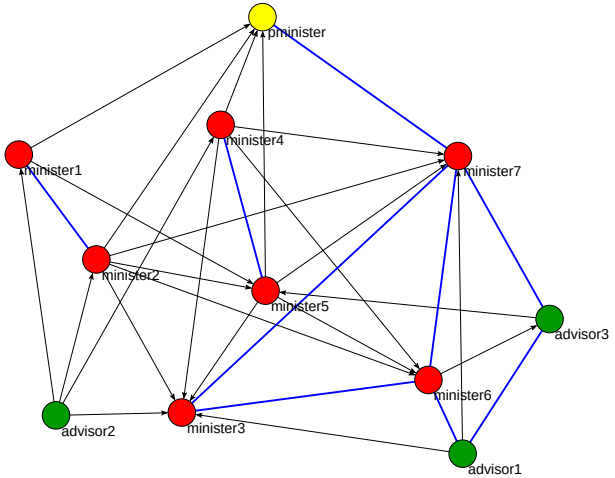
Total error $= 5$

The Student Government at the University of Ljubljana in 1992

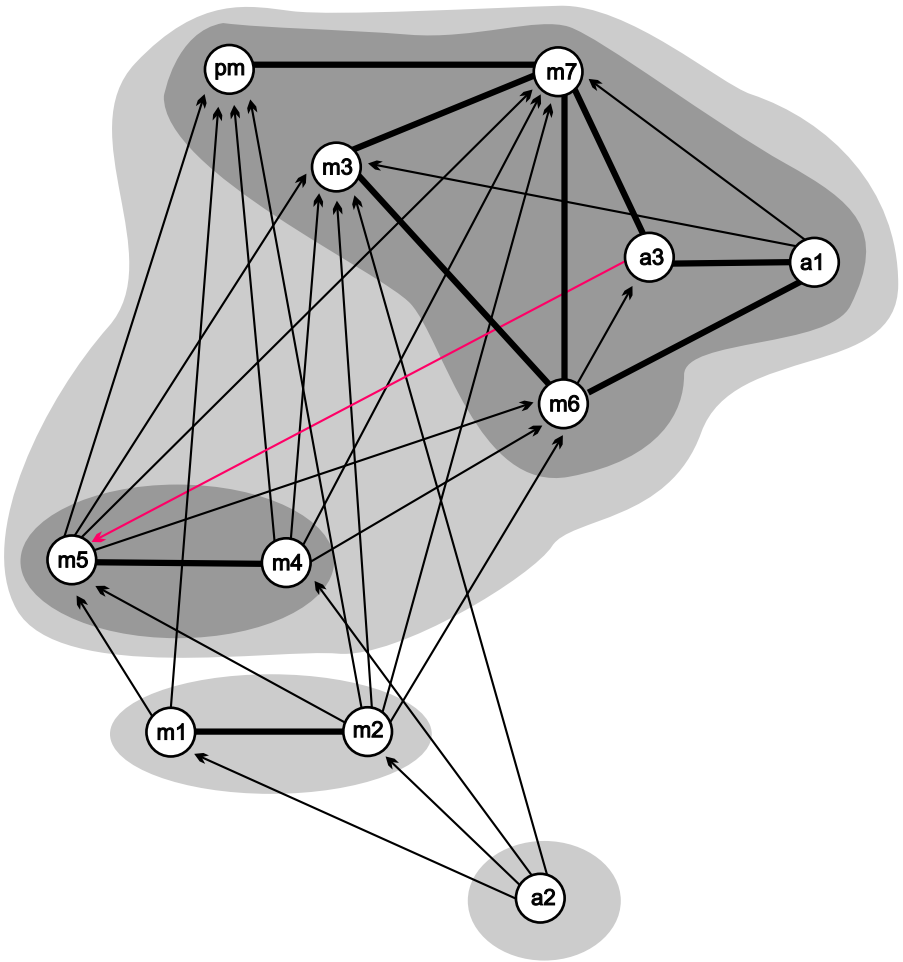
The relation is determined by the following question (Hlebec, 1993):

> *Of the members and advisors of the Student Government, whom do you most often talk with about the matters of the Student Government?*

		m 1	p 2	m 3	m 4	m 5	m 6	m 7	m 8	a 9	a 10	a 11
minister 1	1	·	1	1	·	·	1	·	·	·	·	·
p.minister	2	·	·	·	·	·	·	·	1	·	·	·
minister 2	3	1	1	·	1	·	1	1	1	·	·	·
minister 3	4	·	·	·	·	·	·	1	1	·	·	·
minister 4	5	·	1	·	1	·	1	1	1	·	·	·
minister 5	6	·	1	·	1	1	·	1	1	·	·	·
minister 6	7	·	·	·	1	·	·	·	1	1	·	1
minister 7	8	·	1	·	1	·	·	1	·	·	·	1
adviser 1	9	·	·	·	1	·	·	1	1	·	·	1
adviser 2	10	1	·	1	1	1	·	·	·	·	·	·
adviser 3	11	·	·	·	·	·	1	·	1	1	·	·



A Symmetric Acyclic Blockmodel of Student Government

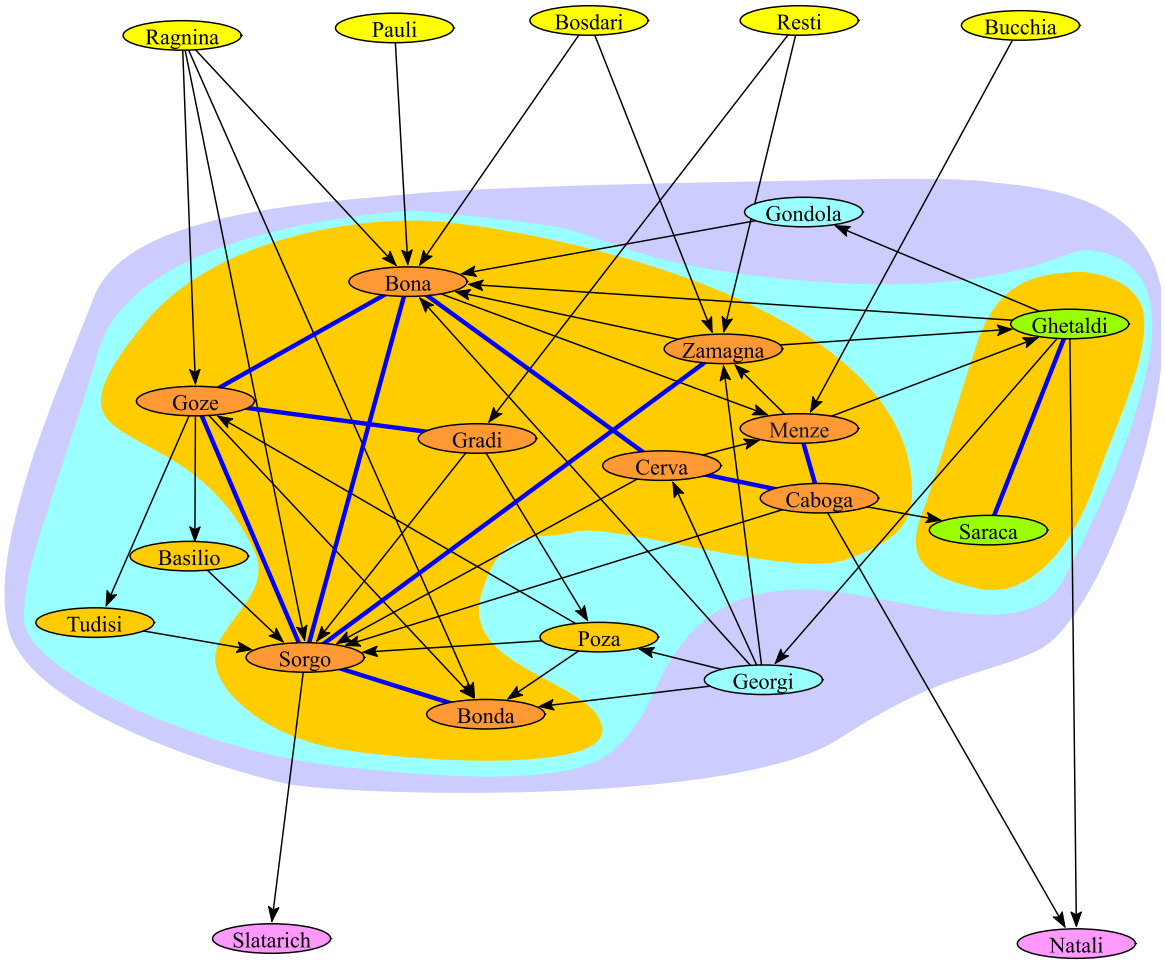


The obtained clustering in 4 clusters is almost exact. The only error is produced by the arc $(a3, m5)$.

Ragusan Noble Families Marriage Network, 18th and 19th Century

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Basilio	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1	·	·
Bona	2	·	·	·	·	·	·	1	·	·	·	2	·	2	·	·	·	·	·	·	·	1	·	·
Bonda	3	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	2	·	·
Bosdari	4	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1
Bucchia	5	·	·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·	·
Caboga	6	·	·	·	·	·	·	1	·	·	·	·	·	1	1	·	·	·	·	1	·	1	·	·
Cerva	7	·	1	·	·	·	1	·	·	·	·	·	·	1	·	·	·	·	·	·	·	1	·	·
Georgi	8	·	1	2	·	·	·	1	·	·	·	·	·	·	·	·	4	·	·	·	·	·	·	1
Ghetaldi	9	·	1	·	·	·	·	·	1	·	1	·	·	·	1	·	·	·	·	1	·	·	·	·
Gondola	10	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Goze	11	1	2	1	·	·	·	·	·	·	·	2	2	·	·	·	·	·	·	·	·	2	1	·
Gradi	12	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	1	·	·	·	·	3	·	·
Menze	13	·	·	·	·	·	1	·	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	1
Natali	14	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Pauli	15	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Poza	16	·	·	2	·	·	·	·	·	·	·	1	·	·	·	·	1	·	·	·	·	1	·	·
Ragnina	17	·	1	1	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·	1	·	·
Resti	18	·	·	·	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·	·	1
Saraca	19	·	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Slatarich	20	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Sorgo	21	·	2	1	·	·	·	·	·	·	·	1	·	·	·	·	·	·	·	·	1	1	·	1
Tudisi	22	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	1	·	·
Zamagna	23	·	1	·	·	·	·	·	·	2	·	·	·	·	·	·	·	·	·	·	·	1	·	·

A Symmetric-Acyclic Decomposition of the Ragusan Families Network



Demo with Pajek

```
Read Network Tina.net
Net/Transform/Arcs-->Edges/Bidirected Only/Max
Draw/Draw
Layout/Energy/Kamada-Kawai/Free
Operations/Blockmodeling/Restricted Options [On]
Operations/Blockmodeling/Random Start
    [4, Ranks.MDL], [Repetitions, 100], [Clusters, 4], [RUN]
    extend the dialog box to see the model
Draw/Draw-Partition
```

Blockmodeling in 2-mode networks

We already presented some ways of rearranging 2-mode network matrices at the beginning of this lecture.

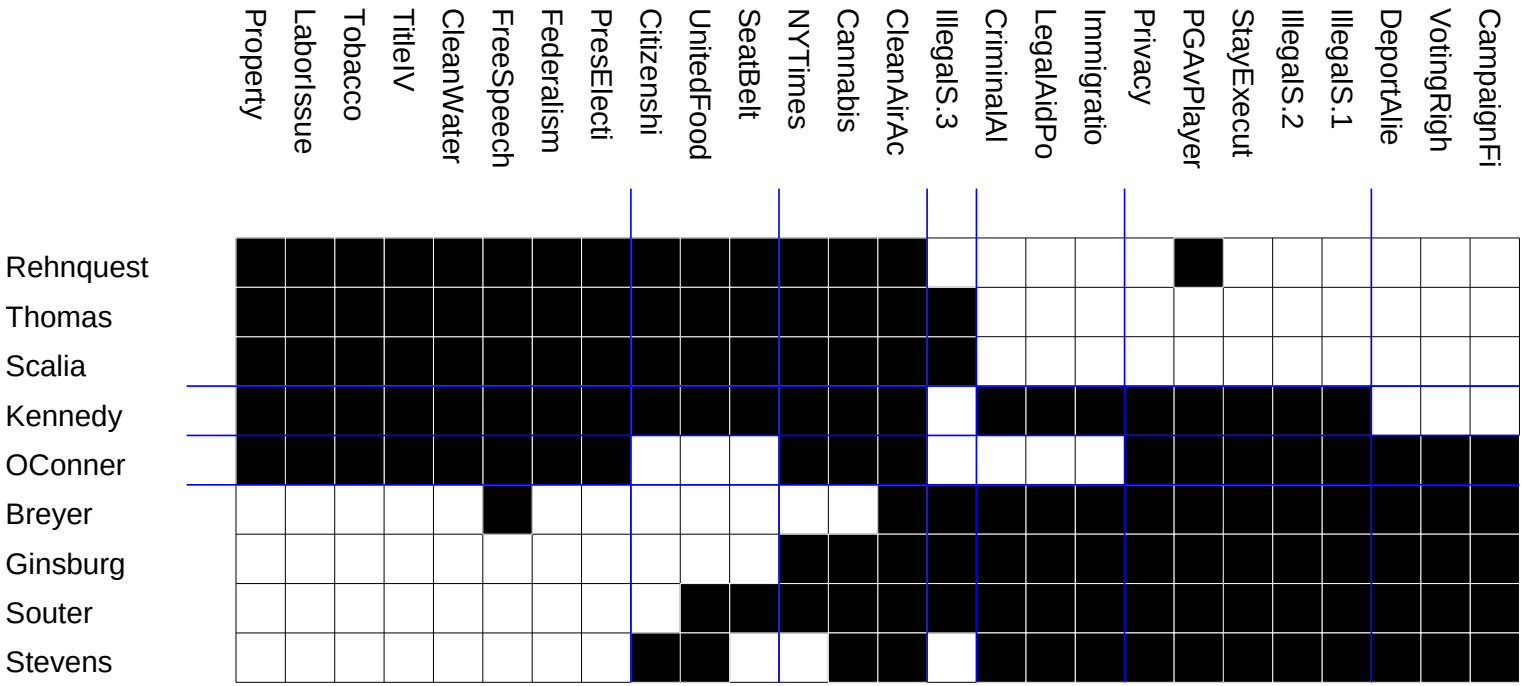
It is also possible to formulate this goal as a generalized blockmodeling problem where the solutions consist of two partitions — row-partition and column-partition.

Supreme Court Voting for Twenty-Six Important Decisions

Issue	Label	Br	Gi	So	St	OC	Ke	Re	Sc	Th
Presidential Election	PE	-	-	-	-	+	+	+	+	+
Criminal Law Cases										
Illegal Search 1	CL1	+	+	+	+	+	+	-	-	-
Illegal Search 2	CL2	+	+	+	+	+	+	-	-	-
Illegal Search 3	CL3	+	+	+	-	-	-	-	+	+
Seat Belts	CL4	-	-	+	-	-	+	+	+	+
Stay of Execution	CL5	+	+	+	+	+	+	-	-	-
Federal Authority Cases										
Federalism	FA1	-	-	-	-	+	+	+	+	+
Clean Air Action	FA2	+	+	+	+	+	+	+	+	+
Clean Water	FA3	-	-	-	-	+	+	+	+	+
Cannabis for Health	FA4	0	+	+	+	+	+	+	+	+
United Foods	FA5	-	-	+	+	-	+	+	+	+
NY Times Copyrights	FA6	-	+	+	-	+	+	+	+	+
Civil Rights Cases										
Voting Rights	CR1	+	+	+	+	+	-	-	-	-
Title VI Disabilities	CR2	-	-	-	-	+	+	+	+	+
PGA v. Handicapped Player	CR3	+	+	+	+	+	+	+	-	-
Immigration Law Cases										
Immigration Jurisdiction	Im1	+	+	+	+	-	+	-	-	-
Deporting Criminal Aliens	Im2	+	+	+	+	+	-	-	-	-
Detaining Criminal Aliens	Im3	+	+	+	+	-	+	-	-	-
Citizenship	Im4	-	-	-	+	-	+	+	+	+
Speech and Press Cases										
Legal Aid for Poor	SP1	+	+	+	+	-	+	-	-	-
Privacy	SP2	+	+	+	+	+	+	-	-	-
Free Speech	SP3	+	-	-	-	+	+	+	+	+
Campaign Finance	SP4	+	+	+	+	+	-	-	-	-
Tobacco Ads	SP5	-	-	-	-	+	+	+	+	+
Labor and Property Rights Cases										
Labor Rights	LPR1	-	-	-	-	+	+	+	+	+
Property Rights	LPR2	-	-	-	-	+	+	+	+	+

The Supreme Court Justices and their 'votes' on a set of 26 "important decisions" made during the 2000-2001 term, Doreian and Fujimoto (2002). The Justices (in the order in which they joined the Supreme Court) are: Rehnquist (1972), Stevens (1975), O'Conner (1981), Scalia (1982), Kennedy (1988), Souter (1990), Ginsburg (1993) and Breyer (1994).

...Supreme Court Voting / a (4,7) partition



upper – conservative / lower – liberal

Signed graphs

A *signed graph* is an ordered pair (G, σ) where

- $G = (V, R)$ is a directed graph (without loops) with set of vertices V and set of arcs $R \subseteq V \times V$;
- $\sigma : R \to \{p, n\}$ is a *sign* function. The arcs with the sign p are *positive* and the arcs with the sign n are *negative*. We denote the set of all positive arcs by R^+ and the set of all negative arcs by R^-.

The case when the graph is undirected can be reduced to the case of directed graph by replacing each edge e by a pair of opposite arcs both signed with the sign of the edge e.

Balanced and clusterable signed graphs

The signed graphs were introduced in Harary, 1953 and later studied by several authors. Following Roberts (1976, p. 75–77) a signed graph (G, σ) is:

- *balanced* iff the set of vertices V can be partitioned into two subsets so that every positive arc joins vertices of the same subset and every negative arc joins vertices of different subsets.
- *clusterable* iff the set of V can be partitioned into subsets, called *clusters*, so that every positive arc joins vertices of the same subset and every negative arc joins vertices of different subsets.

... Properties

The (semi)walk on the signed graph is *positive* iff it contains an even number of negative arcs; otherwise it is *negative*.
The balanced and clusterable signed graphs are characterised by the following theorems:
THEOREM 1. A signed graph (G, σ) is balanced iff every closed semiwalk is positive.

THEOREM 2. A signed graph (G, σ) is clusterable iff G contains no closed semiwalk with exactly one negative arc.

Balance semiring

To construct a semiring corresponding to the *balance* problem we take the set A with four elements: 0 – no walk; n – all walks are negative; p – all walks are positive; a – at least one positive and at least one negative walk.

$+$	0	n	p	a
0	0	n	p	a
n	n	n	a	a
p	p	a	p	a
a	a	a	a	a

$\cdot$	0	n	p	a
0	0	0	0	0
n	0	p	n	a
p	0	n	p	a
a	0	a	a	a

x	$x^\star$
0	p
n	a
p	p
a	a

The balance semiring is idempotent closed semiring with zero 0 and unit p.

Cluster semirings

For construction of the *cluster* semiring corresponding to the *clusterability* problem we need the set A with five elements: 0 – no walk; n – at least one walk with exactly one negative arc, no walk with only positive arcs; p – at least one walk with only positive arcs, no walk with exactly one negative arc; a – at least one walk with only positive arcs, at least one walk with exactly one negative arc; q – each walk has at least two negative arcs.

$+$	0	n	p	a	q
0	0	n	p	a	q
n	n	n	a	a	n
p	p	a	p	a	p
a	a	a	a	a	a
q	q	n	p	a	q

$\cdot$	0	n	p	a	q
0	0	0	0	0	0
n	0	q	n	n	q
p	0	n	p	a	q
a	0	n	a	a	q
q	0	q	q	q	q

x	$x^\star$
0	p
n	a
p	p
a	a
q	p

The cluster semiring is idempotent closed semiring with zero 0 and unit p.

Consequences

Let us define the *symmetric closure* of the value matrix by $\mathbf{D}^\bullet = (\mathbf{D} + \mathbf{D}^T)^\star$ We get:

THEOREM 1'. A signed graph (G, σ) is balanced iff the diagonal of its balance-closure matrix $\mathbf{D}^\bullet_B$ contains only elements with value p.

THEOREM 2'. A signed graph (G, σ) is clusterable iff the diagonal of its cluster-closure matrix $\mathbf{D}^\bullet_C$ contains only elements with value p.

The balance-closure matrix of balanced signed graph contains no element with value a, since in this case the corresponding diagonal elements should also have value a. Similary the cluster-closure matrix of clusterable signed graph contains no element with value a.

A block is a maximal set of vertices with equal lines in matrix $\mathbf{D}^\bullet$.

... Consequences

In balance-closure of balanced signed graph and in cluster-closure of clusterable signed graph all the entries between vertices of two blocks have the same value. The value of entries between vertices of the same block is p.

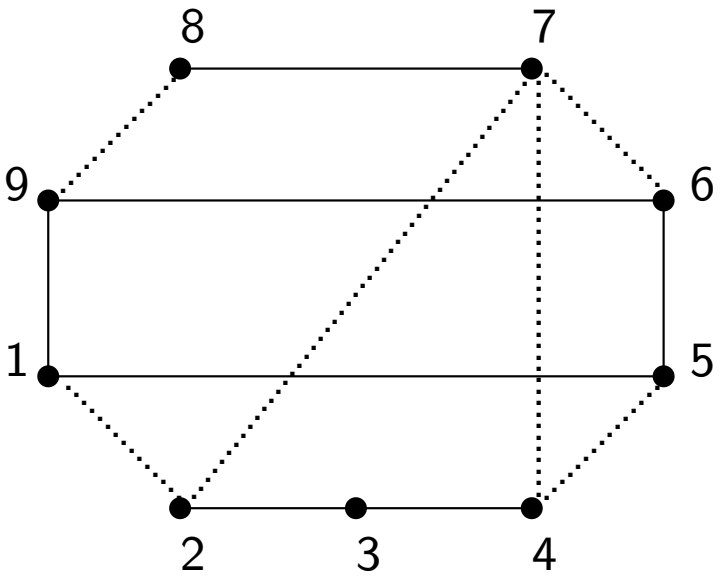
In both cases different partitions of the set of vertices correspond to the (nonequivalent) colorings of the graph with blocks as vertices in which there is an edge between two vertices iff the entries between the corresponding blocks in matrix $\mathbf{D}^\bullet$ have value n.

There is another way to test the clusterability of a given signed graph:

THEOREM 2". A signed graph (G, σ) is clusterable iff $(R^+)^\bullet \cap R^- = \emptyset$, where the closure $^\bullet$ is computed in the semiring $(\{0,1\}, \vee, \wedge, 0, 1)$.

This form of the Theorem 4 is interesting because the intersection $(R^+)^\bullet \cap R^-$ consists of arcs which prevent the signed graph (G, σ) to be clusterable.

Chartrand's example – graph



	1	2	3	4	5	6	7	8	9
1	0	*n*	0	0	*p*	0	0	0	*p*
2	*n*	0	*p*	0	0	0	*n*	0	0
3	0	*p*	0	*p*	0	0	0	0	0
4	0	0	*p*	0	*n*	0	*n*	0	0
5	*p*	0	0	*n*	0	*p*	0	0	0
6	0	0	0	0	*p*	0	*n*	0	*p*
7	0	*n*	0	*n*	0	*n*	0	*p*	0
8	0	0	0	0	0	0	*p*	0	*n*
9	*p*	0	0	0	0	*p*	0	*n*	0

In the figure the graph from Chartarand (1985, p. 181) and its value matrix are given. The positive edges are drawn with solid lines, and the negative edges with dotted lines.

Chartrand's example – closures

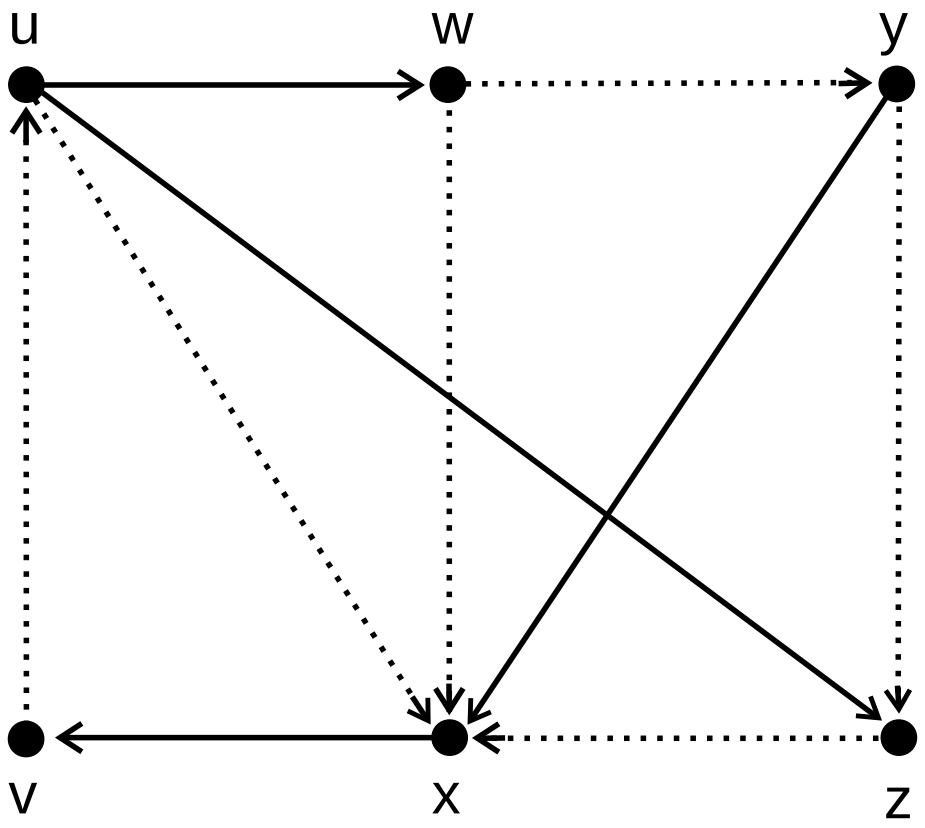
	1	2	3	4	5	6	7	8	9
1	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
2	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
3	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
4	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
5	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
6	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
7	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
8	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*
9	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*	*a*

	1	2	3	4	5	6	7	8	9
1	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
2	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
3	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
4	*n*	**p**	**p**	**p**	*n*	*n*	*n*	*n*	*n*
5	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
6	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**
7	*n*	*n*	*n*	*n*	*n*	*n*	**p**	**p**	*n*
8	*n*	*n*	*n*	*n*	*n*	*n*	**p**	**p**	*n*
9	**p**	*n*	*n*	*n*	**p**	**p**	*n*	*n*	**p**

On the left side of the table the corresponding balance-closure is given – the graph is not balanced. From the cluster-closure on the right side of the table we can see that the graph is clusterable and it has the clusters

$$V_1 = \{1, 5, 6, 9\}, \qquad V_2 = \{2, 3, 4\}, \qquad V_3 = \{7, 8\}$$

Roberts's example



In the figure the signed graph from Roberts (1976, page 77, exercise 16) is presented. Its value matrix on the left side of the following table.

Roberts's example – value matrix and its closure

	u	v	w	x	y	z
u	0	0	*p*	*n*	0	*p*
v	*n*	0	0	0	0	0
w	0	0	0	*n*	*n*	0
x	0	*p*	0	0	0	0
y	0	0	0	*p*	0	*n*
z	0	0	0	*n*	0	0

	u	v	w	x	y	z
u	**p**	*n*	**p**	*n*	*n*	**p**
v	*n*	**p**	*n*	**p**	**p**	*n*
w	**p**	*n*	**p**	*n*	*n*	**p**
x	*n*	**p**	*n*	**p**	**p**	*n*
y	*n*	**p**	*n*	**p**	**p**	*n*
z	**p**	*n*	**p**	*n*	*n*	**p**

In this case the balance-closure and the cluster-closure are equal (right side of the table). The corresponding partition is

$$V_1 = \{v, x, y\}, \qquad V_2 = \{u, w, z\}$$

Razcepnost označenih grafov in bločni modeli

Problemu razcepnosti ustrezajo tri vrste blokov:

- *ničelni* vsi elementi v bloku so enaki 0;
- *pozitivni* vsi elementi v bloku so pozitivni ali enaki 0;
- *negativni* vsi elementi v bloku so negativni ali enaki 0;

Če dana razvrstitev določa razcep omrežja, so diagonalni bloki pozitivni (ali ničelni), nediagonalni pa negativni ali ničelni.
Kakovost dane razvrstitve $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ torej lahko izmerimo takole ($0 \leq \alpha \leq 1$):

$$P_\alpha(\mathbf{C}) = \alpha \sum_{C \in \mathbf{C}} \sum_{u,v \in C} \max(0, -w_{uv}) + (1-\alpha) \sum_{\substack{C,C' \in \mathbf{C} \\ C \neq C'}} \sum_{u \in C, v \in C'} \max(0, w_{uv})$$

Za določitev čim boljše razvrstitve uporabimo lokalno optimizacijo.

Slovenske stranke 1994 (S. Kropivnik)

		1	2	3	4	5	6	7	8	9	10
SKD	1	0	-215	114	-89	-77	94	-170	176	117	-210
ZLSD	2	-215	0	-217	134	77	-150	57	-253	-230	49
SDSS	3	114	-217	0	-203	-80	138	-109	177	180	-174
LDS	4	-89	134	-203	0	157	-142	173	-241	-254	23
ZSESS	5	-77	77	-80	157	0	-188	170	-120	-160	-9
ZS	6	94	-150	138	-142	-188	0	-97	140	116	-106
DS	7	-170	57	-109	173	170	-97	0	-184	-191	-6
SLS	8	176	-253	177	-241	-120	140	-184	0	235	-132
SPS-SNS	9	117	-230	180	-254	-160	116	-191	235	0	-164
SNS	10	-210	49	-174	23	-9	-106	-6	-132	-164	0

Slovenske stranke 1994 / preurejene

		1	3	6	8	9	2	4	5	7	10
SKD	1	0	114	94	176	117	-215	-89	-77	-170	-210
SDSS	3	114	0	138	177	180	-217	-203	-80	-109	-174
ZS	6	94	138	0	140	116	-150	-142	-188	-97	-106
SLS	8	176	177	140	0	235	-253	-241	-120	-184	-132
SPS-SNS	9	117	180	116	235	0	-230	-254	-160	-191	-164
ZLSD	2	-215	-217	-150	-253	-230	0	134	77	57	49
LDS	4	-89	-203	-142	-241	-254	134	0	157	173	23
ZSESS	5	-77	-80	-188	-120	-160	77	157	0	170	-9
DS	7	-170	-109	-97	-184	-191	57	173	170	0	-6
SNS	10	-210	-174	-106	-132	-164	49	23	-9	-6	0

3-way blockmodeling

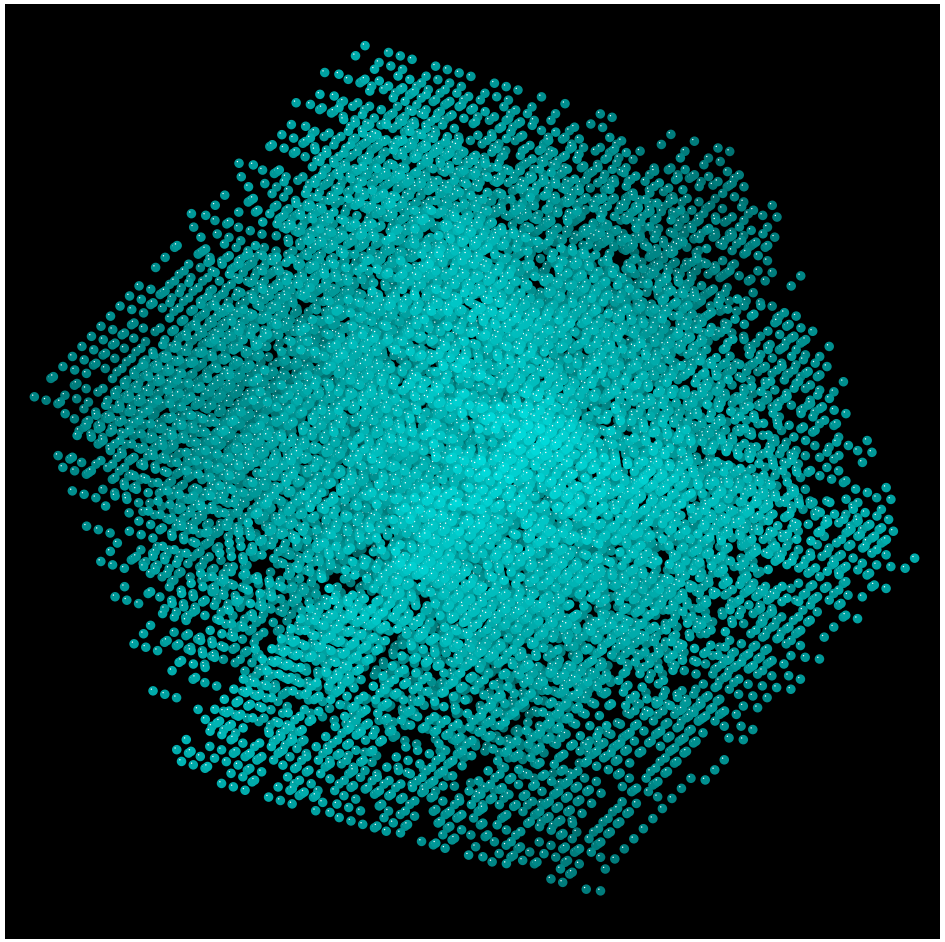
We started to work on blockmodeling of 3-way networks. We developed the indirect approach to *structural equivalence blockmodeling in 3-way networks*. *Indirect* means – embedding the notion of equivalence in a dissimilarity and determining it using clustering.
3-way network is defined by three sets of units X, Y and Z. There are three basic cases:

- all three sets are different (3-mode netork)
- two sets are the same (2-mode network)
- all three sets are the same (1-mode network)

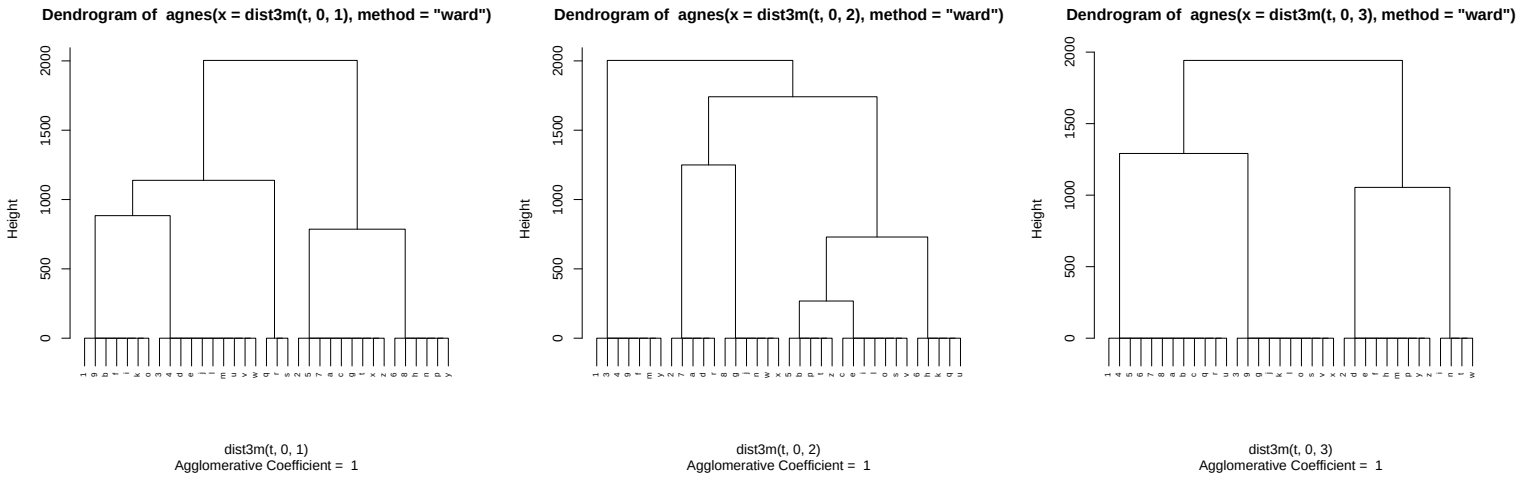
For all three cases we constructed compatible dissimilarities for structural equivalence .

Example 1: Artificial dataset

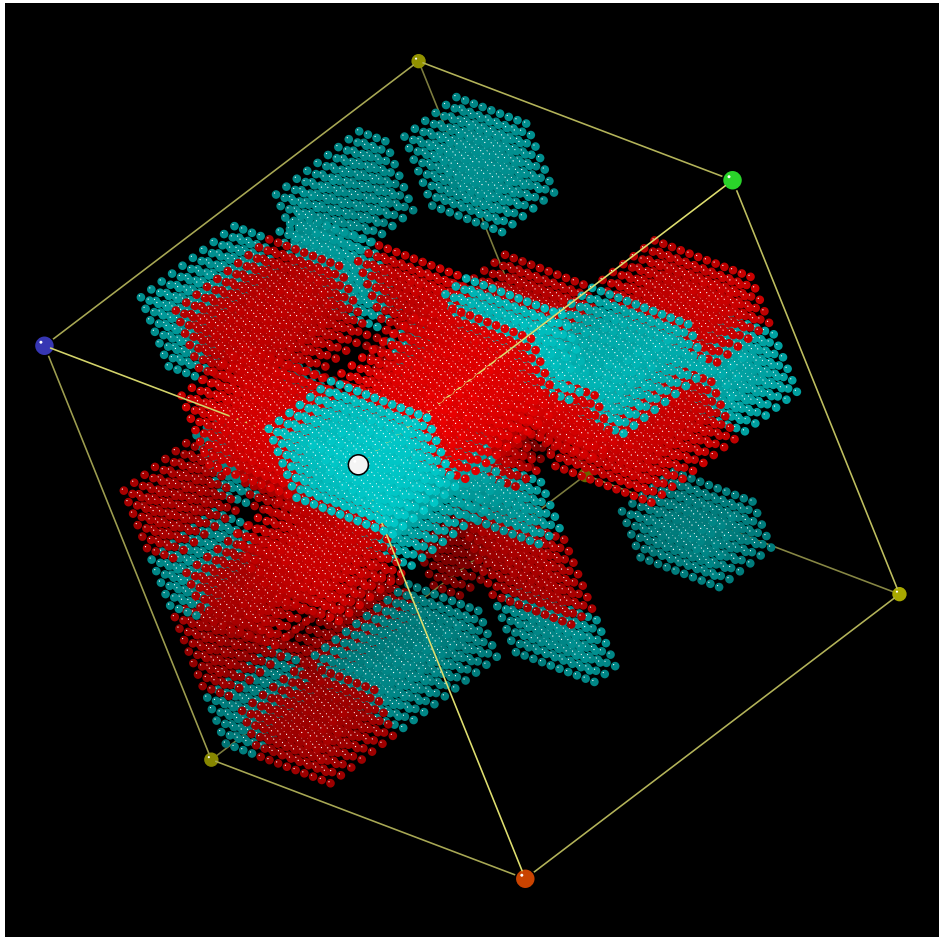
Randomly generated ideal structure `rndTest(c(5,6,4),c(35,35,35))`:



Example 1: Dendrograms

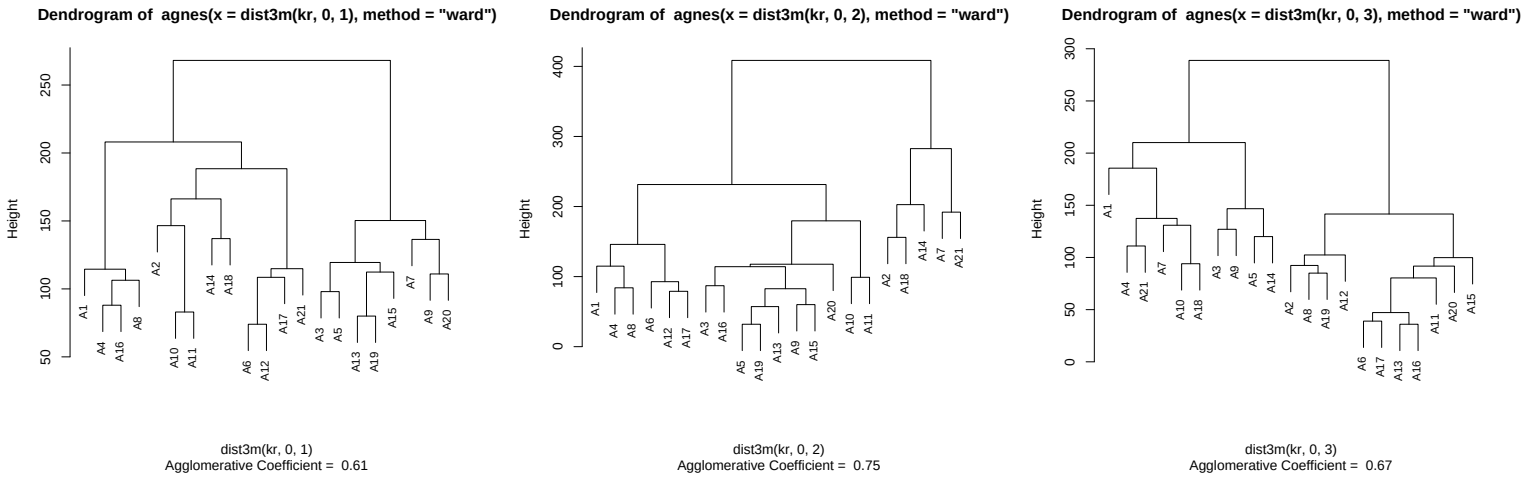


Example 1: Solutions

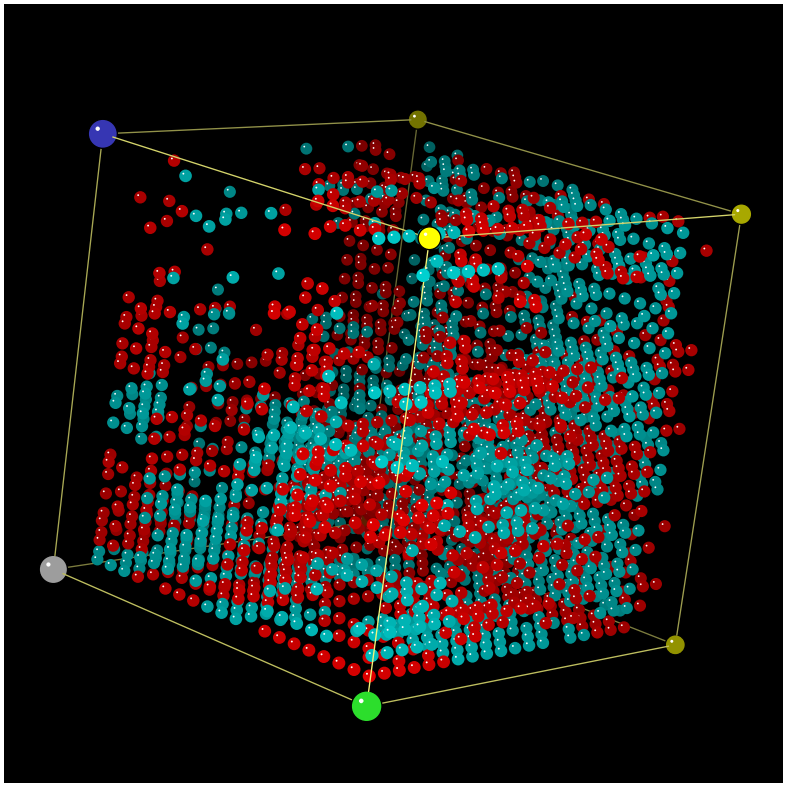


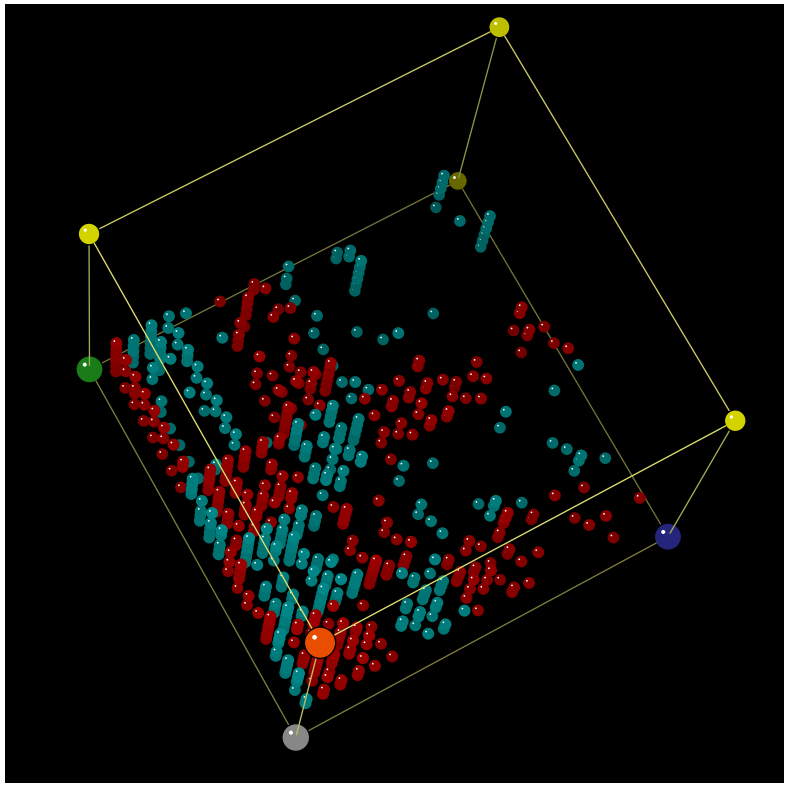
Click on the picture!

Example 2: Krackhardt / Dendrograms



Example 2: Krackhardt / Solutions

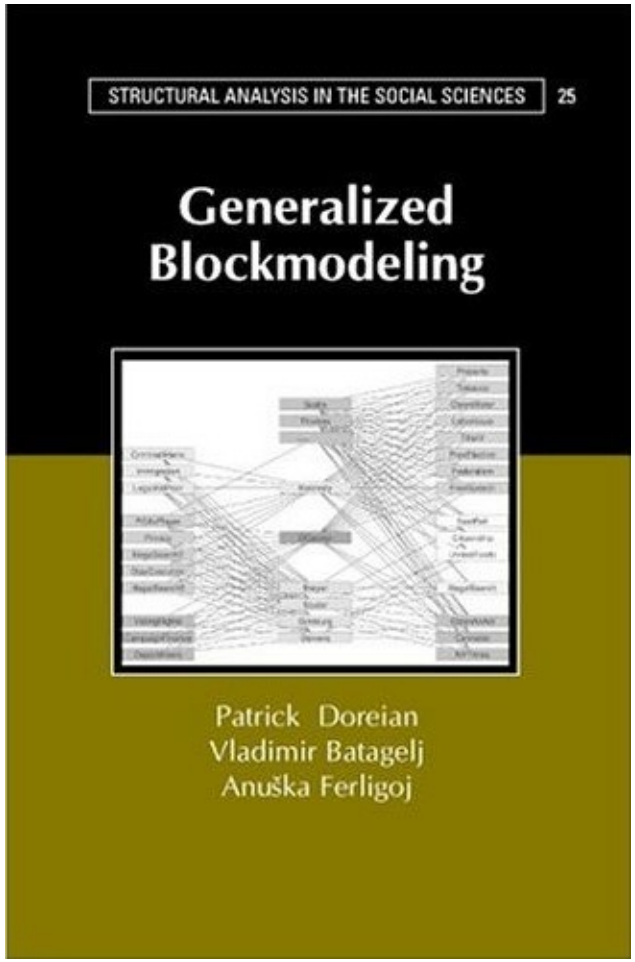




Blockmodeling of Valued Networks

Batagelj and Ferligoj (2000) proposed an approach to blockmodel of valued networks as an example of *relational data analysis*. These ideas were further developed by Žiberna (2007) who proposed some approaches for generalized blockmodeling of valued networks.
The first one is a straightforward generalization of the generalized blockmodeling of binary networks to valued blockmodeling. The second approach is homogeneity blockmodeling where the basic idea is that the inconsistency of an empirical block with its ideal block can be measured by within block variability of appropriate values. Žiberna provided new ideal blocks appropriate for blockmodeling of valued networks together with definitions of their block inconsistencies.

More on blockmodeling



The details about the generalized blockmodeling can be found in our book:
P. Doreian, V. Batagelj, A. Ferligoj:
Generalized Blockmodeling, CUP, 2005.

Conditions for hierarchical clustering methods

The set of feasible clusterings $\mathbf{\Phi}$ determines the *feasibility predicate* $\Phi(\mathbf{C}) \equiv \mathbf{C} \in \mathbf{\Phi}$ defined on $\mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\})$; and conversely $\mathbf{\Phi} \equiv \{\mathbf{C} \in \mathcal{P}(\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}) : \Phi(\mathbf{C})\}$.
In the set $\mathbf{\Phi}$ the relation of *clustering inclusion* $\sqsubseteq$ can be introduced by

$$\mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \equiv \forall C_1 \in \mathbf{C}_1, C_2 \in \mathbf{C}_2 : C_1 \cap C_2 \in \{\emptyset, C_1\}$$

we say also that the clustering $\mathbf{C}_1$ is a *refinement* of the clustering $\mathbf{C}_2$.
It is well known that $(\Pi(\mathcal{U}), \sqsubseteq)$ is a partially ordered set (even more, semimodular lattice). Because any subset of partially ordered set is also partially ordered, we have: Let $\mathbf{\Phi} \subseteq \Pi(\mathcal{U})$ then $(\mathbf{\Phi}, \sqsubseteq)$ is a partially ordered set.
The clustering inclusion determines two related relations (on $\mathbf{\Phi}$):
$\mathbf{C}_1 \sqsubset \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubseteq \mathbf{C}_2 \wedge \mathbf{C}_1 \neq \mathbf{C}_2$ – strict inclusion, and
$\mathbf{C}_1 \dot{\sqsubset} \mathbf{C}_2 \equiv \mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \neg\exists \mathbf{C} \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2)$ – predecessor.

Conditions on the structure of the set of feasible clusterings

We shall assume that the set of feasible clusterings $\mathbf{\Phi} \subseteq \Pi(\mathcal{U})$ satisfies the following conditions:

F1. $\mathbf{O} \equiv \{\{X\} : X \in \mathcal{U}\} \in \mathbf{\Phi}$

F2. The feasibility predicate Φ is *local* – it has the form $\Phi(\mathbf{C}) = \bigwedge_{C \in \mathbf{C}} \varphi(C)$ where $\varphi(C)$ is a predicate defined on $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ (clusters).
The intuitive meaning of $\varphi(C)$ is: $\varphi(C) \equiv$ the cluster C is 'good'.
Therefore the locality condition can be read: a 'good' clustering $\mathbf{C} \in \mathbf{\Phi}$ consists of 'good' clusters.

F3. The predicate Φ has the property of *binary heredity* with respect to the *fusibility* predicate $\psi(C_1, C_2)$, i.e.,

$$C_1 \cap C_2 = \emptyset \wedge \varphi(C_1) \wedge \varphi(C_2) \wedge \psi(C_1, C_2) \Rightarrow \varphi(C_1 \cup C_2)$$

This condition means: in a 'good' clustering, a fusion of two 'fusible' clusters produces a 'good' clustering.

. . . conditions

F4. The predicate ψ is *compatible* with clustering inclusion $\sqsubseteq$, i.e.,

$$\forall \mathbf{C}_1, \mathbf{C}_2 \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathbf{C}_1 \setminus \mathbf{C}_2 = \{C_1, C_2\} \Rightarrow \psi(C_1, C_2) \vee \psi(C_2, C_1))$$

F5. The *interpolation* property holds in $\mathbf{\Phi}$, i.e., $\forall \mathbf{C}_1, \mathbf{C}_2 \in \mathbf{\Phi}$:

$$(\mathbf{C}_1 \sqsubset \mathbf{C}_2 \wedge \mathrm{card}((\,)\mathbf{C}_1) > \mathrm{card}((\,)\mathbf{C}_2) + 1 \Rightarrow \exists \mathbf{C} \in \mathbf{\Phi} : (\mathbf{C}_1 \sqsubset \mathbf{C} \wedge \mathbf{C} \sqsubset \mathbf{C}_2))$$

These conditions provide a framework in which the hierarchical methods can be applied also for constrained clustering problems $\mathbf{\Phi}_k(\mathcal{U}) \subset \Pi_k(\mathcal{U})$. In the ordinary problem both predicates $\varphi(C)$ and $\psi(C_p, C_q)$ are always true – all conditions F1-F5 are satisfied.

Clustering with relational constraint

Suppose that the units are described by attribute data $a\colon \mathcal{U} \to [\mathcal{U}]$ and related by a binary *relation* $R \subseteq \mathcal{U} \times \mathcal{U}$ that determine the *relational data* $(\mathcal{U}, R, a)$.
We want to cluster the units according to the similarity of their descriptions, but also considering the relation R – it imposes *constraints* on the set of feasible clusterings, usually in the following form:

$$\mathbf{\Phi}(R) = \{\mathbf{C} \in P(\mathcal{U}) : \text{each cluster } C \in \mathbf{C} \text{ is a subgraph } (C, R \cap C \times C) \text{ in the graph } (\mathcal{U}, R) \text{ of the required type of connectedness}\}$$

Example: regionalization problem – group given territorial units into regions such that units inside the region will be similar and form contiguous part of the territory.

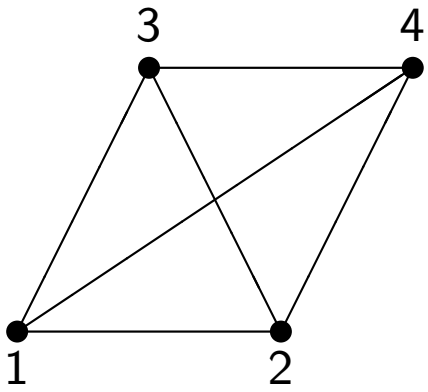
Some types of relational constraints

We can define different types of sets of feasible clusterings for the same relation R. Some examples of *types of relational constraint* $\Phi^i(R)$ are

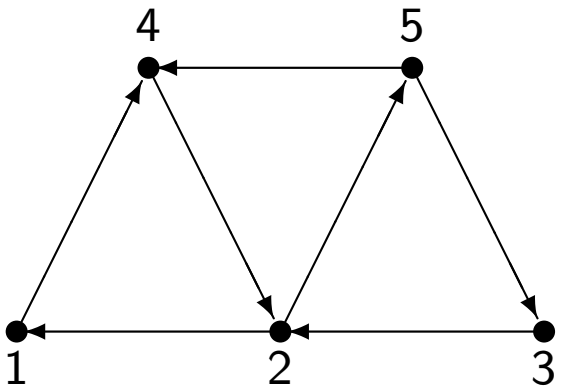
type of clusterings	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	the existence of a trail containing all the units of the cluster

Trail – all arcs are distinct.
A set of units $L \subseteq C$ is a *center* of cluster C in the clustering of type $\Phi^2(R)$ iff the subgraph induced by L is strongly connected and $R(L) \cap (C \setminus L) = \emptyset$.

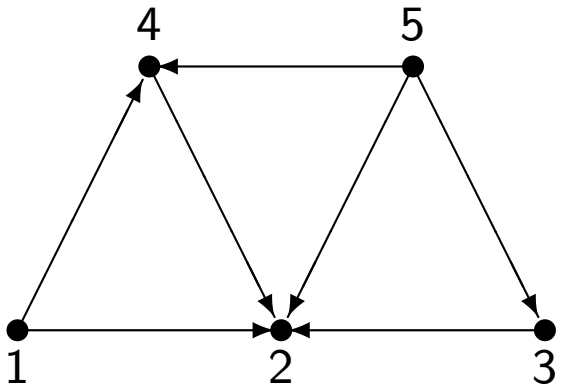
Some graphs of different types



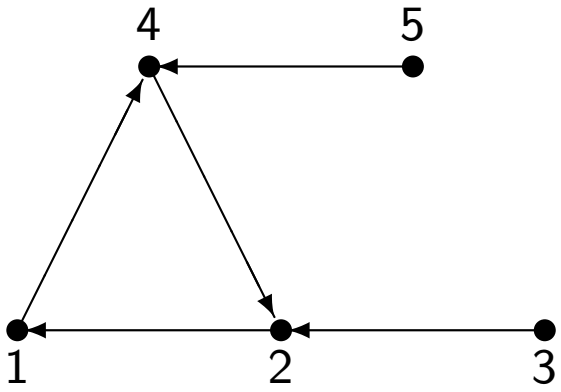
a clique



strongly connected units



weakly connected units



weakly connected units

with a center $\{1, 2, 4\}$

Properties of relational constraints

The sets of feasible clusterings $\Phi^i(R)$ are linked as follows:
$\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$
$\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$
If the relation R is symmetric, then $\Phi^3(R) = \Phi^1(R)$
If the relation R is an equivalence relation, then $\Phi^4(R) = \Phi^1(R)$
Here are also examples of the corresponding fusibility predicates:
$\psi^1(C_1, C_2) \equiv \exists X \in C_1 \exists Y \in C_2 : (XRY \vee YRX)$
$\psi^2(C_1, C_2) \equiv (\exists X \in L_1 \exists Y \in C_2 : XRY) \vee (\exists X \in C_1 \exists Y \in L_2 : YRX)$
$\psi^3(C_1, C_2) \equiv (\exists X \in C_1 \exists Y \in C_2 : XRY) \wedge (\exists X \in C_1 \exists Y \in C_2 : YRX)$
$\psi^4(C_1, C_2) \equiv \forall X \in C_1 \forall Y \in C_2 : (XRY \wedge YRX)$
$\psi^5(C_1, C_2) \equiv (\exists X \in T_1 \exists Y \in I_2 : XRY) \vee (\exists X \in I_1 \exists Y \in T_2 : YRX)$
For ψ^3 the property F5 fails.

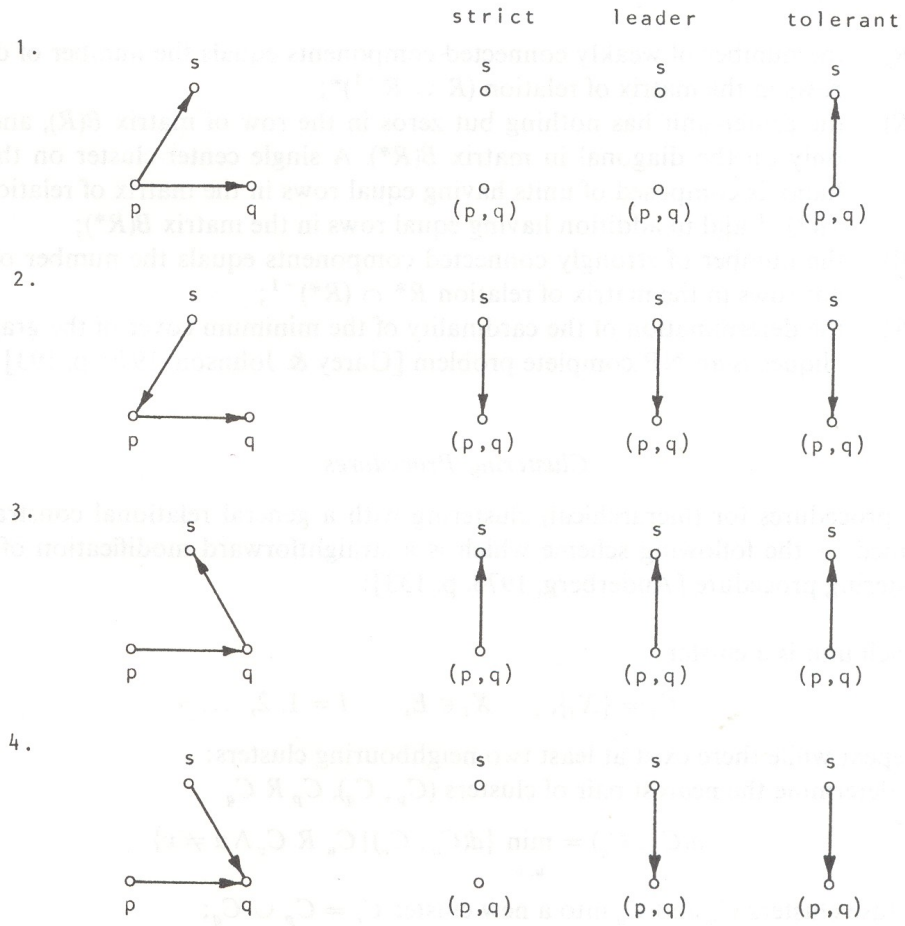
Agglomerative method for relational constraints

We can use both hierarchical and local optimization methods for solving some types of problems with relational constraint (Ferligoj, Batagelj 1983).

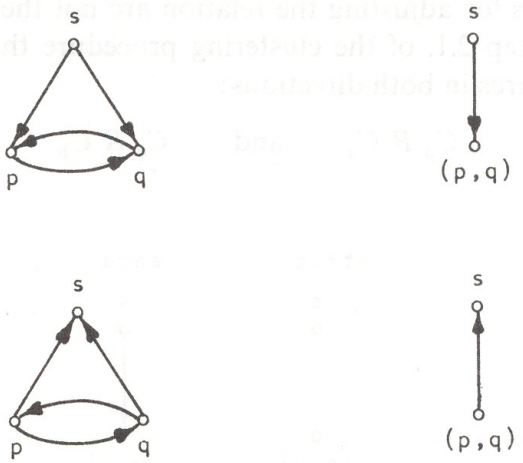
1. $k := n$; $\mathbf{C}(k) := \{\{\mathrm{X}\} : \mathrm{X} \in \mathcal{U}\}$;
2. **while** $\exists C_i, C_j \in \mathbf{C}(k)$: $(i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \mathrm{argmin}\{D(C_i, C_j) : i \neq j \wedge \psi(C_i, C_j)\}$;
2.2. $C := C_p \cup C_q$; $k := k - 1$;
2.3. $\mathbf{C}(k) := \mathbf{C}(k+1) \setminus \{C_p, C_q\} \cup \{C\}$;
2.4. determine $D(C, C_s)$ for all $C_s \in \mathbf{C}(k)$
2.4. adjust the relation R as required by the clustering type
3. $m := k$

The condition $\psi(C_i, C_j)$ is equivalent to $C_i R C_j$ for tolerant, leader and strict method; and to $C_i R C_j \wedge C_j R C_i$ for two-way method.

Adjusting relation after joining

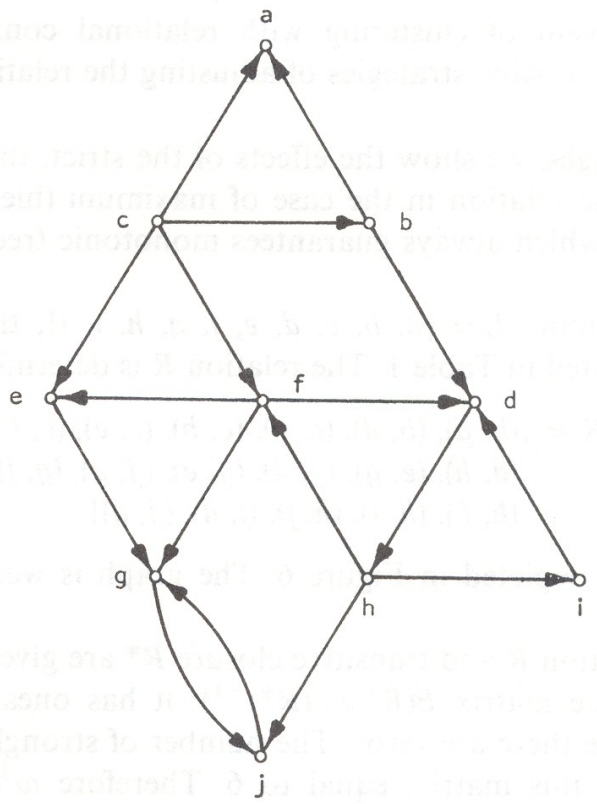


Φ^1 – tolerant
Φ^2 – leader
Φ^4 – two-way
Φ^5 – strict

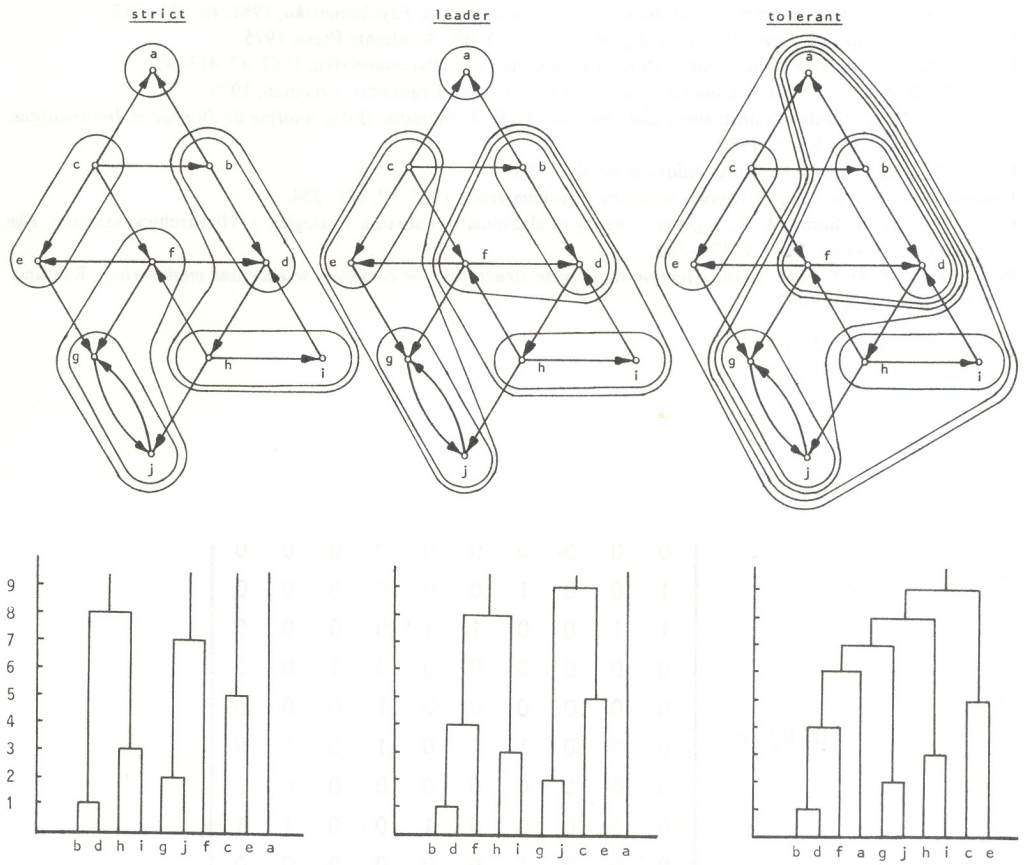


Example - problem

	a	b	c	d	e	f	g	h	i	j
a	0	5	7	4	6	6	2	4	2	3
b		0	8	1	3	4	4	5	3	4
c			0	4	5	7	9	3	2	5
d				0	3	2	4	8	6	3
e					0	6	4	6	5	7
f						0	6	8	5	7
g							0	4	8	2
h								0	3	4
i									0	5
j										0



Example - solution



Dissimilarities between clusters

In the original approach a complete dissimilarity matrix is needed. To obtain fast algorithms we propose to *consider only the dissimilarities between linked units*.

Let $(\mathcal{U}, R)$, $R \subseteq \mathcal{U} \times \mathcal{U}$ be a graph and $\emptyset \subset S, T \subset \mathcal{U}$ and $S \cap T = \emptyset$.

We call a *block* of relation R for S and T its part $R(S, T) = R \cap S \times T$.

The *symmetric closure* of relation R we denote with $\hat{R} = R \cup R^{-1}$. It holds: $\hat{R}(S, T) = \hat{R}(T, S)$.

For all dissimilarities between clusters $D(S, T)$ we set:

$$D(\{s\}, \{t\}) = \begin{cases} d(s, t) & s\hat{R}t \\ \infty & \text{otherwise} \end{cases}$$

where d is a selected dissimilarity between units.

Minimum and Maximum

Minimum

$$D_{\min}(S, T) = \min_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$D_{\min}(S, T_1 \cup T_2) = \min(D_{\min}(S, T_1), D_{\min}(S, T_2))$$

Maximum

$$D_{\max}(S, T) = \max_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$D_{\max}(S, T_1 \cup T_2) = \max(D_{\max}(S, T_1), D_{\max}(S, T_2))$$

Average

Average

$w : V \to \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathcal{U}$.

$$D_{\mathrm{a}}(S, T) = \frac{1}{w(\hat{R}(S, T))} \sum_{(s,t) \in \hat{R}(S,T)} d(s, t)$$

$$w(\hat{R}(S, T_1 \cup T_2)) = w(\hat{R}(S, T_1)) + w(\hat{R}(S, T_2))$$

$$D_{\mathrm{a}}(S, T_1 \cup T_2) = \frac{w(\hat{R}(S, T_1))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_1) + \frac{w(\hat{R}(S, T_2))}{w(\hat{R}(S, T_1 \cup T_2))} D_{\mathrm{a}}(S, T_2)$$

Hierarchies

The agglomerative clustering procedure produces a series of feasible clusterings $\mathbf{C}(n)$, $\mathbf{C}(n-1)$, ..., $\mathbf{C}(m)$ with $\mathbf{C}(m) \in \operatorname{Max} \mathbf{\Phi}$ (maximal elements for $\sqsubseteq$).
Their union $\mathcal{T} = \bigcup_{k=m}^{n} \mathbf{C}(k)$ is called a *hierarchy* and has the property

$$\forall C_p, C_q \in \mathcal{T} : C_p \cap C_q \in \{\emptyset, C_p, C_q\}$$

The set inclusion $\subseteq$ is a *tree* or *hierarchical* order on $\mathcal{T}$. The hierarchy $\mathcal{T}$ is *complete* iff $\mathcal{U} \in \mathcal{T}$.
For $W \subseteq \mathcal{U}$ we define the *smallest cluster* $C_{\mathcal{T}}(W)$ from $\mathcal{T}$ containing W as:
c1. $W \subseteq C_{\mathcal{T}}(W)$
c2. $\forall C \in \mathcal{T} : (W \subseteq C \Rightarrow C_{\mathcal{T}}(W) \subseteq C)$
$C_{\mathcal{T}}$ is a *closure* on $\mathcal{T}$ with a special property

$$\mathrm{Z} \notin C_{\mathcal{T}}(\{\mathrm{X},\mathrm{Y}\}) \Rightarrow C_{\mathcal{T}}(\{\mathrm{X},\mathrm{Y}\}) \subset C_{\mathcal{T}}(\{\mathrm{X},\mathrm{Y},\mathrm{Z}\}) = C_{\mathcal{T}}(\{\mathrm{X},\mathrm{Z}\}) = C_{\mathcal{T}}(\{\mathrm{Y},\mathrm{Z}\})$$

Level functions

A mapping $h : \mathcal{T} \to \mathbb{R}_0^+$ is a *level function* on $\mathcal{T}$ iff

l1. $\forall X \in \mathcal{U} : h(\{X\}) = 0$

l2. $C_p \subseteq C_q \Rightarrow h(C_p) \leq h(C_q)$

A simple example of level function is $h(C) = \text{card}(()C) - 1$.

Every hierarchy / level function determines an ultrametric dissimilarity on $\mathcal{U}$

$$\delta(X, Y) = h(C_{\mathcal{T}}(\{X, Y\}))$$

The converse is also true (see Dieudonne (1960)): Let d be an ultrametric on $\mathcal{U}$. Denote $\overline{B}(X, r) = \{Y \in \mathcal{U} : d(X, Y) \leq r\}$. Then for any given set $A \subset \mathbb{R}^+$ the set

$$\mathbf{C}(A) = \{\overline{B}(X, r) : X \in \mathcal{U}, r \in A\} \cup \{\{\mathcal{U}\}\} \cup \{\{X\} : X \in \mathcal{U}\}$$

is a complete hierarchy, and $h(C) = \mathrm{diam}(C)$ is a level function.

The pair $(\mathcal{T}, h)$ is called a *dendrogram* or a *clustering tree* because it can be visualized as a tree.

Reducibility

The dissimilarity D has the *reducibility* property (Bruynooghe, 1977) iff

$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \;\Rightarrow$$

$$\min(D(C_p, C_s), d(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

or equivalently

$$D(C_p, C_q) \leq t,\; D(C_p, C_s) \geq t,\; D(C_q, C_s) \geq t \;\Rightarrow\; D(C_p \cup C_q, C_s) \geq t$$

Theorem

If a dissimilarity D has the reducibility property then h_D is a level function.

Nearest neighbors graphs

For a given dissimilarity d on the set of units $\mathcal{U}$ and relational constraint R we define the *k nearest neighbors graph* $\mathbf{G}_{NN} = (\mathcal{U}, A)$

$(\mathrm{X}, \mathrm{Y}) \in A \Leftrightarrow \mathrm{Y}$ is selected among the nearest neighbors of X and $\mathrm{X}\hat{R}\mathrm{Y}$

By setting for $(\mathrm{X}, \mathrm{Y}) \in A$ its value to $w((\mathrm{X}, \mathrm{Y})) = d(\mathrm{X}, \mathrm{Y})$ we obtain a network $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$.

In the case of equidistant pairs of units we have to decide – or to include them all in the graph, or specify an additional selection rule. We shall denote by $\mathbf{G}^*_{NN}$ the graph with included all equidistant pairs, and by $\mathbf{G}_{NN}$ a graph where a single nearest neighbor is always selected.

Structure and properties of the nearest neighbor graphs

Let $\mathcal{N}_{NN} = (\mathcal{U}, A, w)$ be a nearest neighbor network. A pair of units $\mathrm{X}, \mathrm{Y} \in \mathcal{U}$ are *reciprocal nearest neighbors* or RNNs iff $(\mathrm{X}, \mathrm{Y}) \in A$ and $(\mathrm{Y}, \mathrm{X}) \in A$.
Suppose $\mathrm{card}(()\mathcal{U}) > 1$ and R has no isolated units. Then in $\mathcal{N}$

- every unit/vertex $\mathrm{X} \in \mathcal{U}$ has the $\mathrm{outdeg}(\mathrm{X}) \geq 1$ — there is no isolated unit;
- along every walk the values of w are not increasing.

using these two observations we can show that in $\mathcal{N}^*_{NN}$:

- all the values of w on a closed walk are the same and all its arcs are reciprocal — all arcs between units in a nontrivial (at least 2 units) strong component are reciprocal;
- every maximal (can not be extended) elementary (no arc is repeated) walk ends in a RNNs pair;
- there exists at least one RNNs pair – corresponding to $\min_{\mathrm{X},\mathrm{Y} \in \mathcal{U}, \mathrm{X} \neq \mathrm{Y}} d(\mathrm{X}, \mathrm{Y})$.

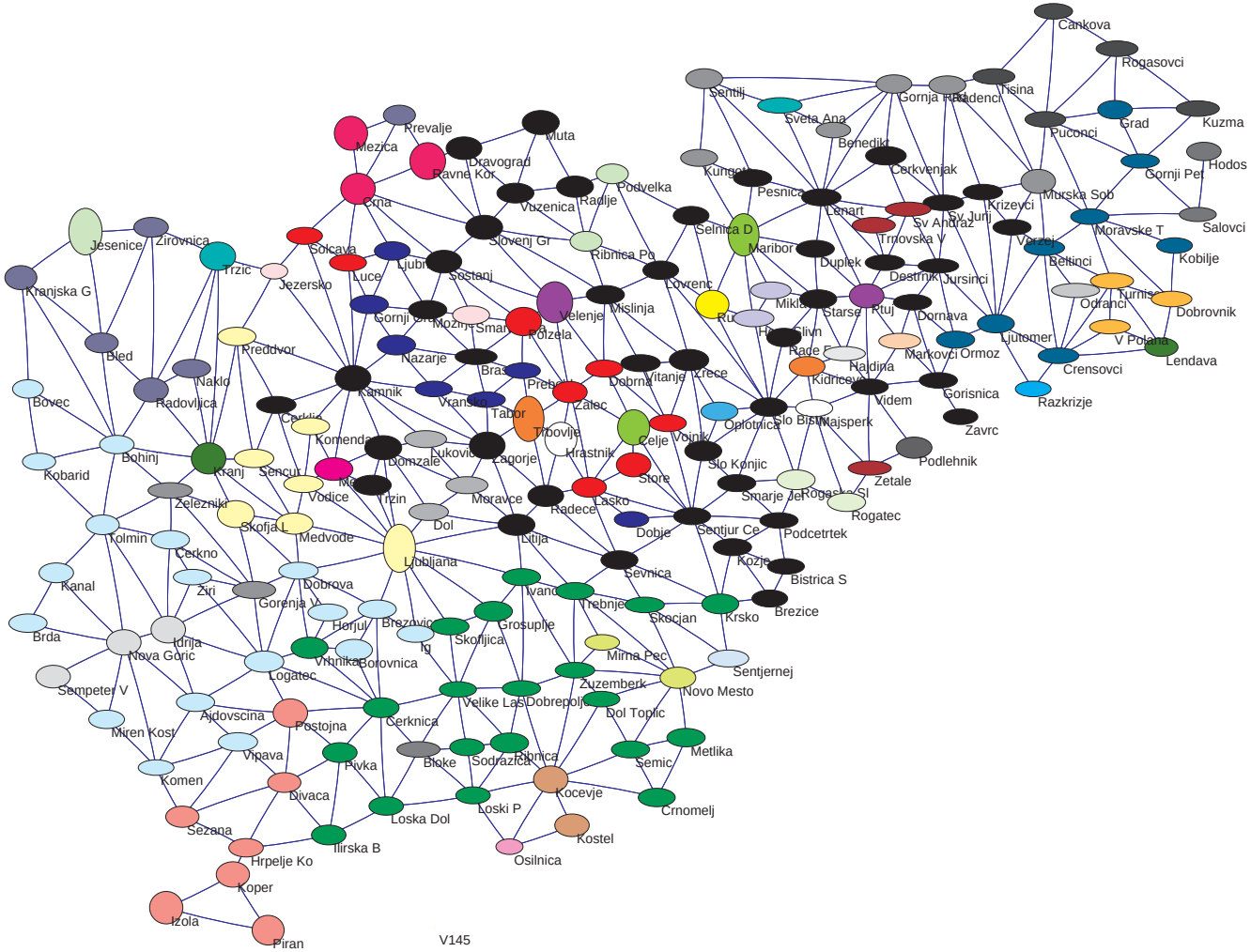
Fast agglomerative clustering algorithms

Any network $\mathcal{N}_{NN}$ is a subnetwork of $\mathcal{N}^*_{NN}$. Its connected components are directed (acyclic) trees with a single RNNs pair in the root.
Based on the nearest neighbor network very efficient algorithms for agglomerative clustering for methods with the reducibility property can be built.

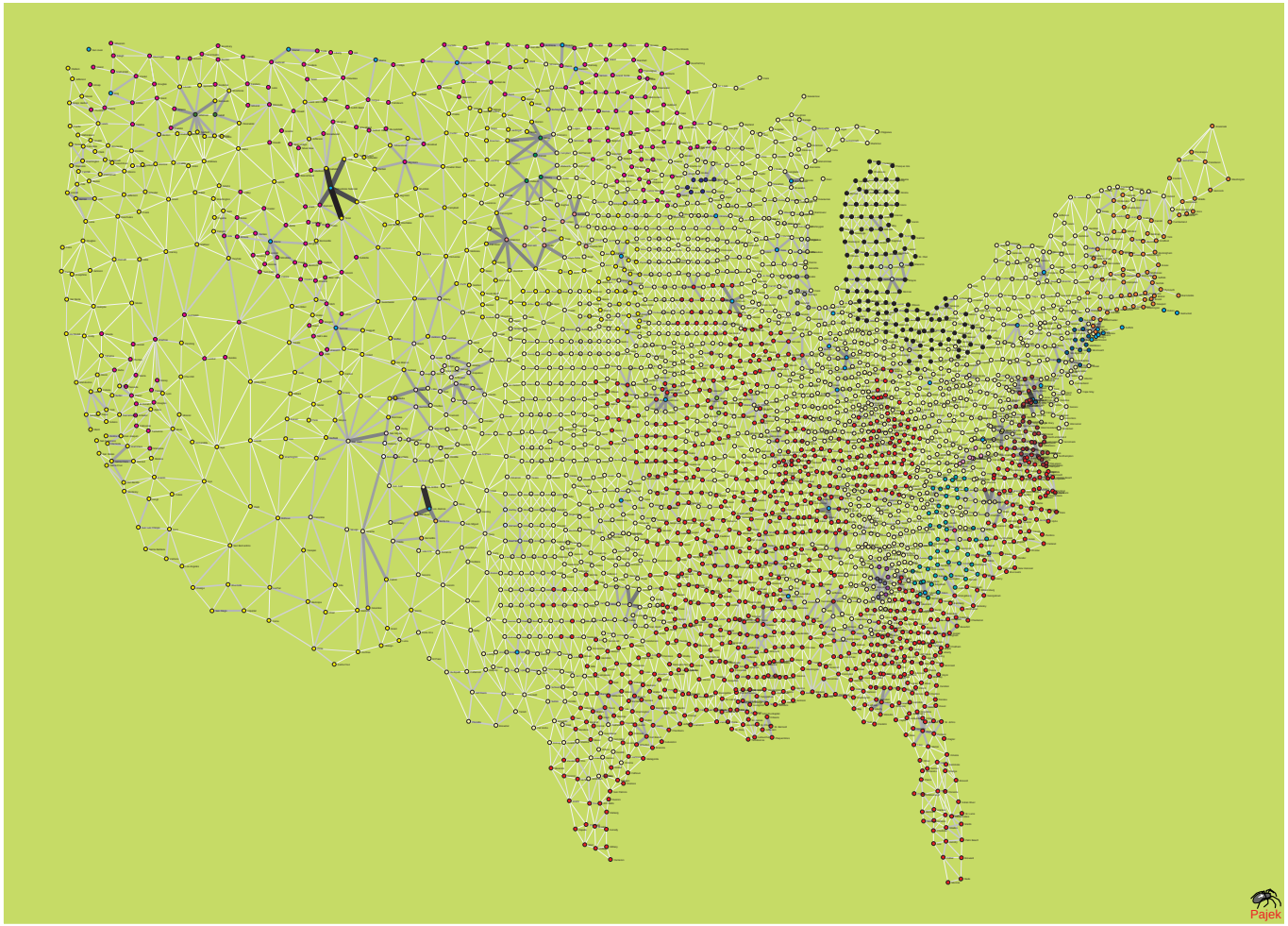
chain := [ ]; **W** := $\mathcal{U}$;
while card(()**W**) > 1 **do begin**
  if *chain* = [ ] **then** select an arbitrary unit X $\in$ **W** **else** X := *last*(*chain*);
  grow a NN-chain from X until a pair (Y, Z) of RNNs are obtained;
  agglomerate Y and Z:
    T := Y $\cup$ Z; **W** := **W** $\setminus$ {Y, Z} $\cup$ {T}; compute $D(T, W)$, $W \in$ **W**
end;

It can be shown that if the clustering method has the reducibility property then the NN-chain remains a NN-chain also after the agglomeration of the RNNs pair.
`Network/Create Hierarchy/Clustering with Relational Constraint/`

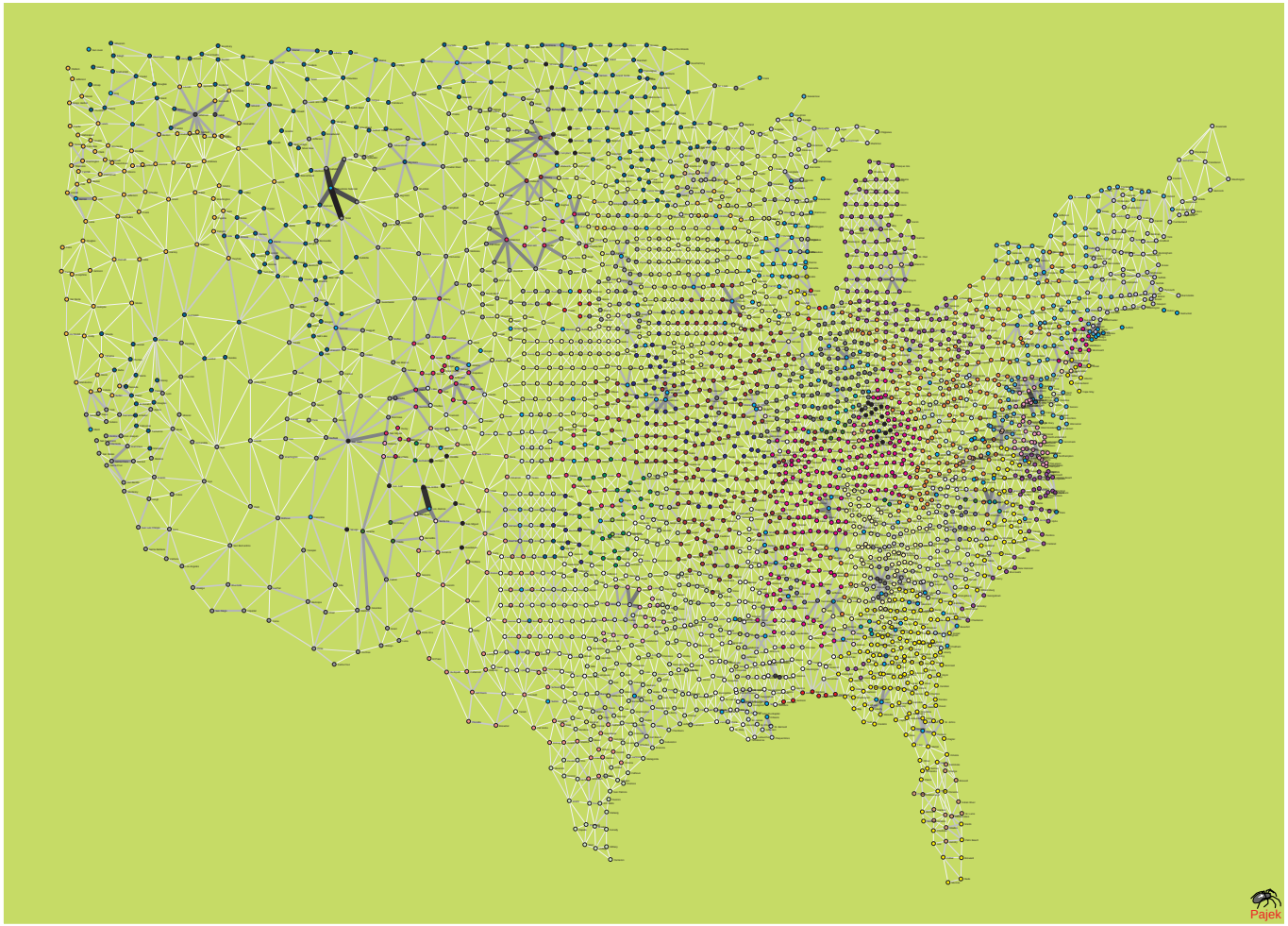
Example: Slovenian communes



Example: US counties $t = 1400$



Example: US counties $t = 200$



Louvain method and VOS

One of the approaches to determine the clustering **C** of a network is to maximize its *modularity*

$$Q(\mathbf{C}) = \sum_{C \in \mathbf{C}} \left(\frac{l(C)}{m} - \left(\frac{d(C)}{2m} \right)^2 \right)$$

where $l(C)$ is the number of edges between vertices belonging to cluster C, and $d(C)$ is the sum of the degrees of vertices from C.

The modularity maximization problem is NP complete.

Louvain method and VOS.

Final Remarks

The current, local optimization based, programs for generalized blockmodeling can deal only with networks with at most some hundreds of units. What to do with larger networks is an open question. For some specialized problems also procedures for (very) large networks can be developed (Doreian, Batagelj, Ferligoj, 1998; Batagelj, Zaveršnik, 2002).

Another interesting problem is the development of *blockmodeling of valued networks* or more general *relational data analysis* (Batagelj, Ferligoj, 2000).

References

- Batagelj V., Ferligoj A. (2000): Clustering relational data. Data Analysis (Eds.: W. Gaul, O. Opitz, M. Schader), Springer, Berlin, 3–15.
- Bruynooghe, M. (1977), Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. Statistique et Analyse des Données, **3**, 24–42.
- Doreian, P., Batagelj, V., Ferligoj, A. (2000), *Symmetric-acyclic decompositions of networks*. *J. classif.*, **17**(1), 3–28.
- Ferligoj A., Batagelj V. (1982), Clustering with relational constraint. *Psychometrika*, **47**(4), 413–426.
- Ferligoj A., Batagelj V. (1983), Some types of clustering with relational constraints. *Psychometrika*, **48**(4), 541–552.
- Murtagh, F. (1985), Multidimensional Clustering Algorithms, *Compstat lectures,* **4**, Vienna: Physica-Verlag.