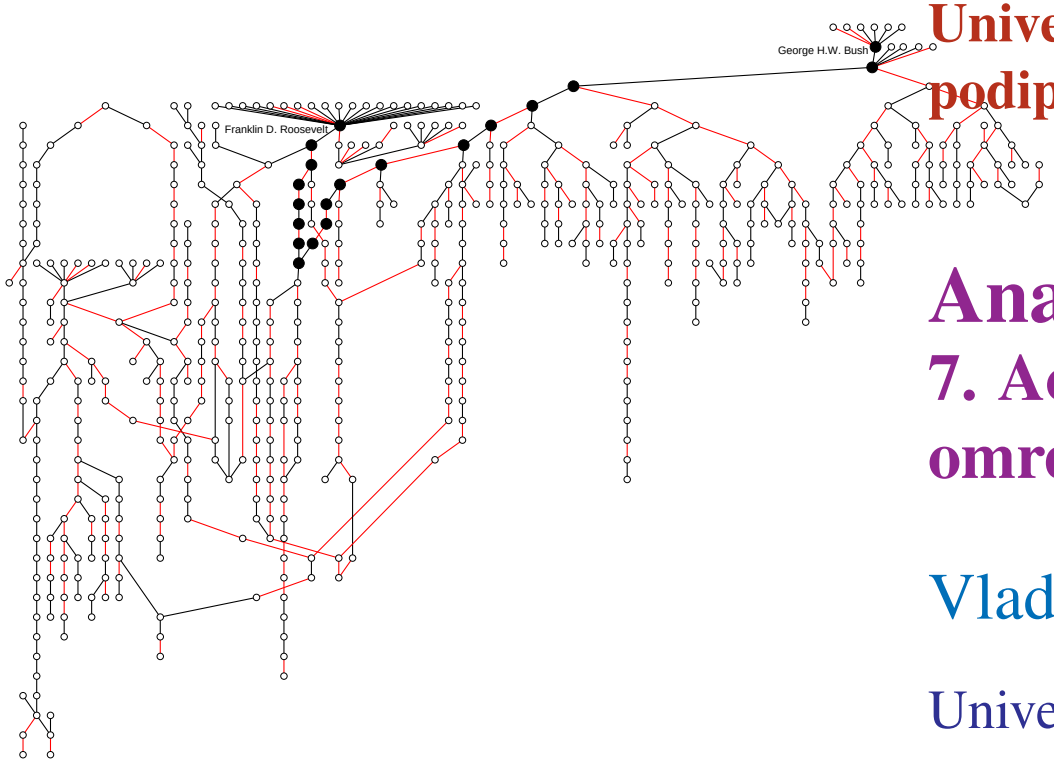


Univerza v Ljubljani
podiplomski študij statistike

Analiza omrežij
7. Aciklična in dvovrstna omrežja

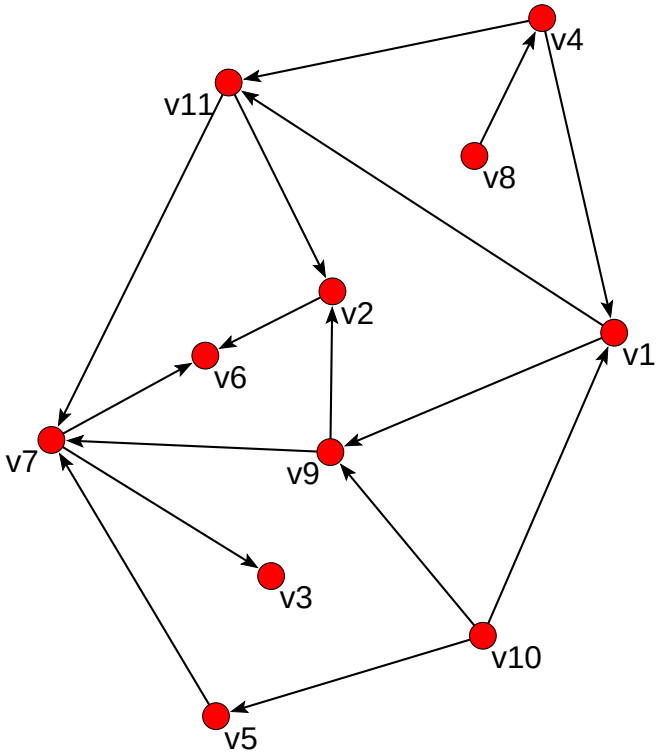
Vladimir Batagelj

Univerza v Ljubljani

Ljubljana, 7. december 2006 / 5. januar 2004

Kazalo

Aciklična omrežja



acyclic.paj

Omrežje $\mathcal{G} = (\mathcal{V}, \mathbf{R})$, $\mathbf{R} \subseteq \mathcal{V} \times \mathcal{V}$ je *acikličnо*, 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 $p : \mathcal{V} \rightarrow \mathbb{R}$ (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.

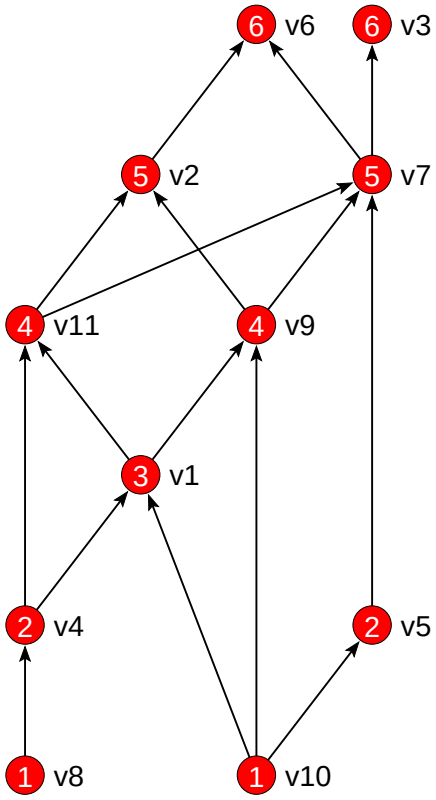
Množica *začetkov* $\mathrm{Min}_{\mathbf{R}}(\mathcal{V}) = \{v : \neg\exists u \in \mathcal{V} : (u, v) \in \mathbf{R}\}$ in množica *koncev* $\mathrm{Max}_{\mathbf{R}}(\mathcal{V}) = \{v : \neg\exists u \in \mathcal{V} : (v, u) \in \mathbf{R}\}$ sta v končnem acikličnem omrežju neprazni.

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

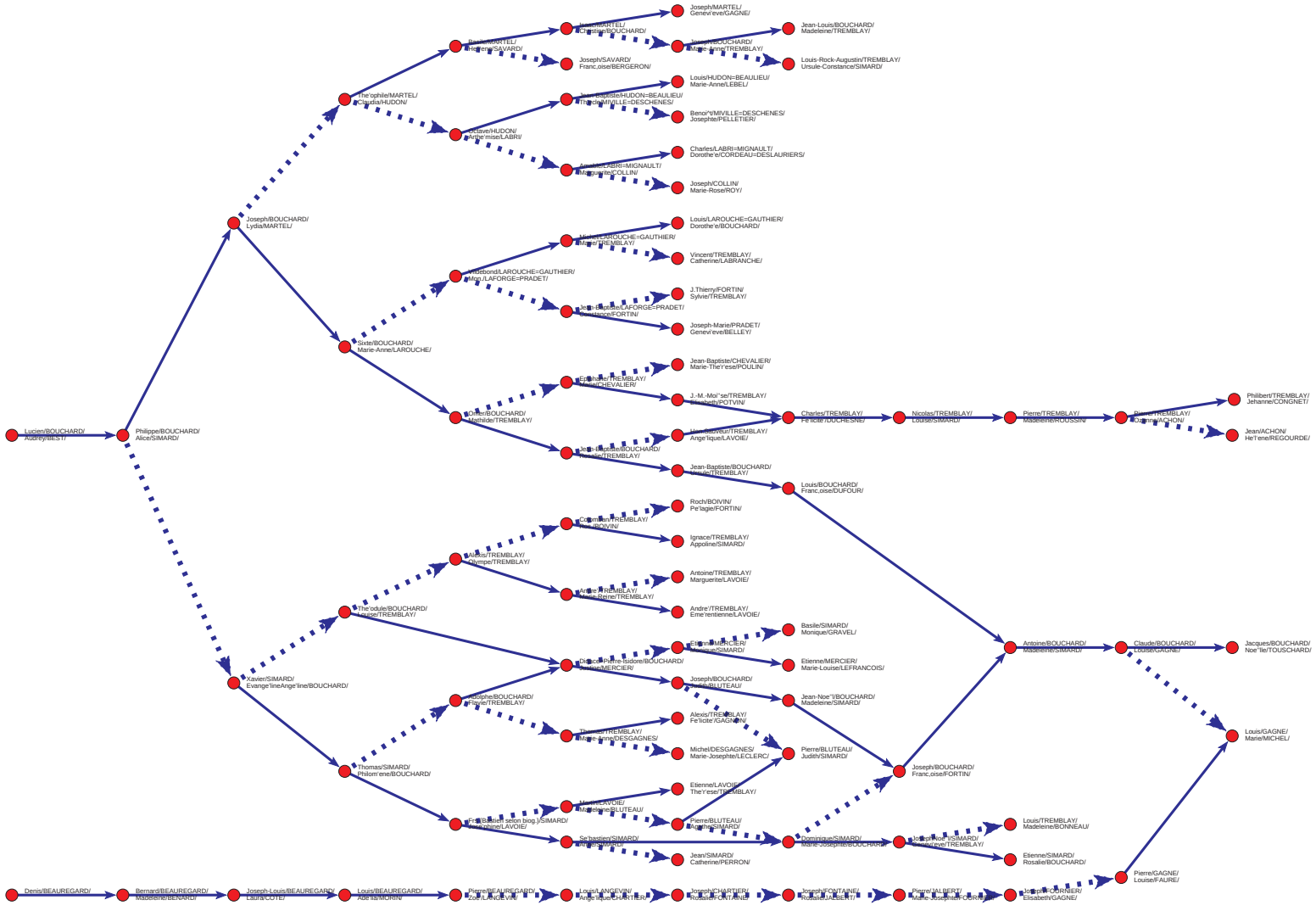


Preslikava $h : \mathcal{V} \rightarrow \mathbb{N}^+$ je *globina*, če so vse razlike na najdaljši poti in vrednost v začetku enake 1.

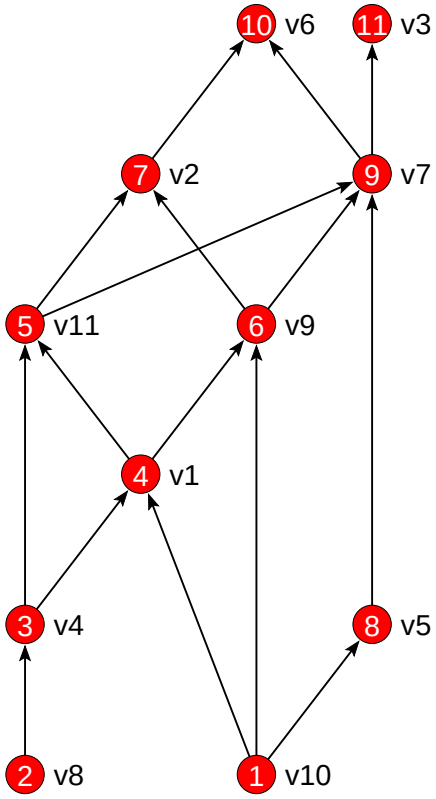
$\mathcal{U} \leftarrow \mathcal{V}; k \leftarrow 0$
while $\mathcal{U} \neq \emptyset$ **do**
  $\mathcal{T} \leftarrow \mathrm{Min}_R(\mathcal{U}); k \leftarrow k + 1$
  for $v \in \mathcal{T}$ **do** $h(v) \leftarrow k$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{T}$

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

Prikaz parnega grafa Bouchardovega rodovnika



Pravilna oštevilčenja



Injektivna, z relacijo $\mathbf{R}$ usklajena preslikava $h : \mathcal{V} \rightarrow 1..|\mathcal{V}|$ je *pravilno oštevilčenje*.

'Topološko urejanje'

$\mathcal{U} \leftarrow \mathcal{V}; k \leftarrow 0$

while $\mathcal{U} \neq \emptyset$ **do**

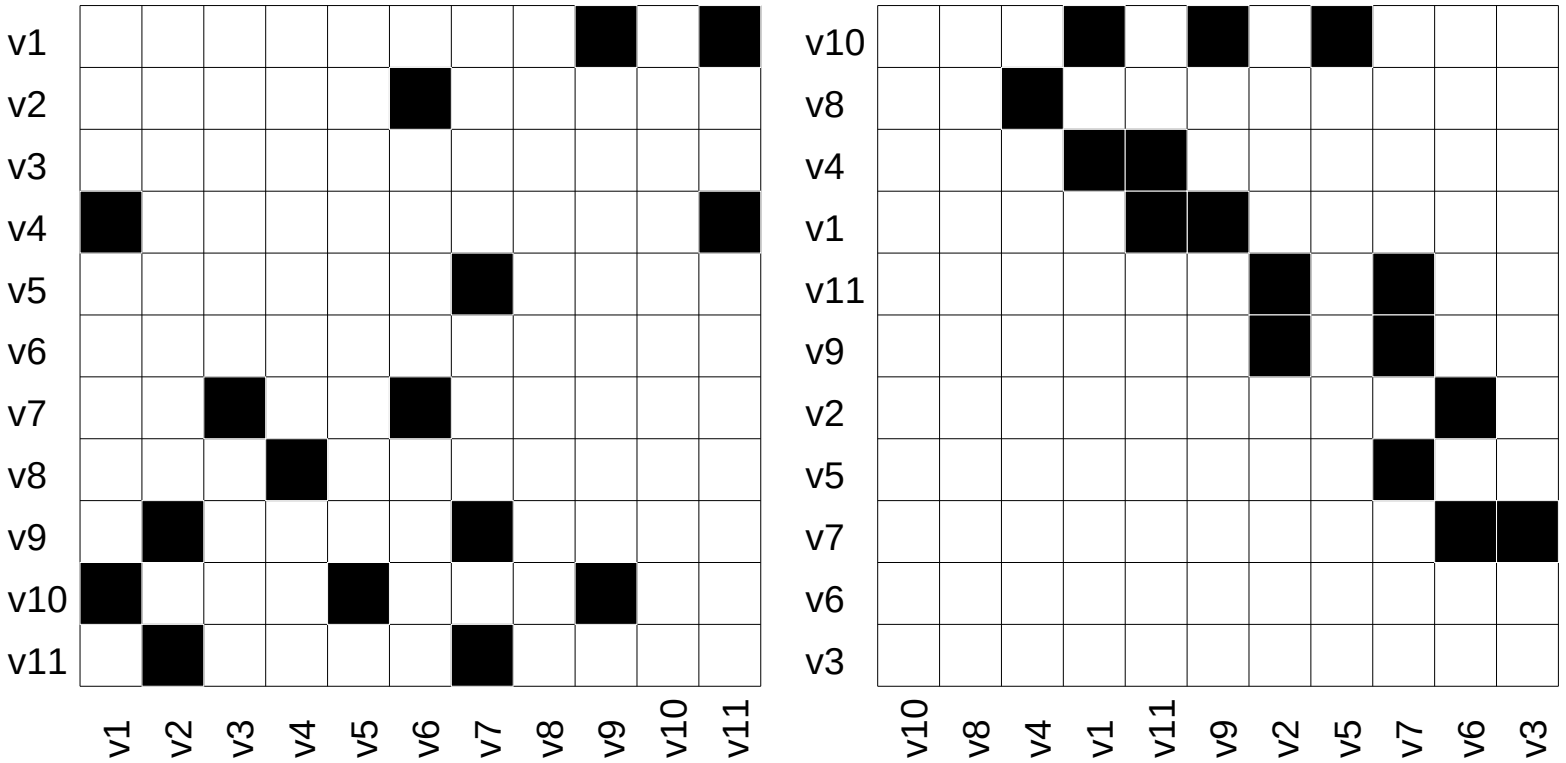
izberi $v \in \mathrm{Min}_{\mathbf{R}}(\mathcal{U}); k \leftarrow k + 1$

$h(v) \leftarrow k$

$\mathcal{U} \leftarrow \mathcal{U} \setminus \{v\}$

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

. . . Pravilna oštevilčenja




```
File/Pajek Project File/Read  [Acyclic.paj]
Net/Partitions/Depth/Acyclic
Partition/Make Permutation
File/Network/Export Matrix to EPS/Using Permutation [a.eps]
```

Pravilna oštevilčenja in izračun vrednosti

Naj bo funkcija $f : \mathcal{V} \rightarrow \mathbb{R}$ 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 oštevilčenje, 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. Točke 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.

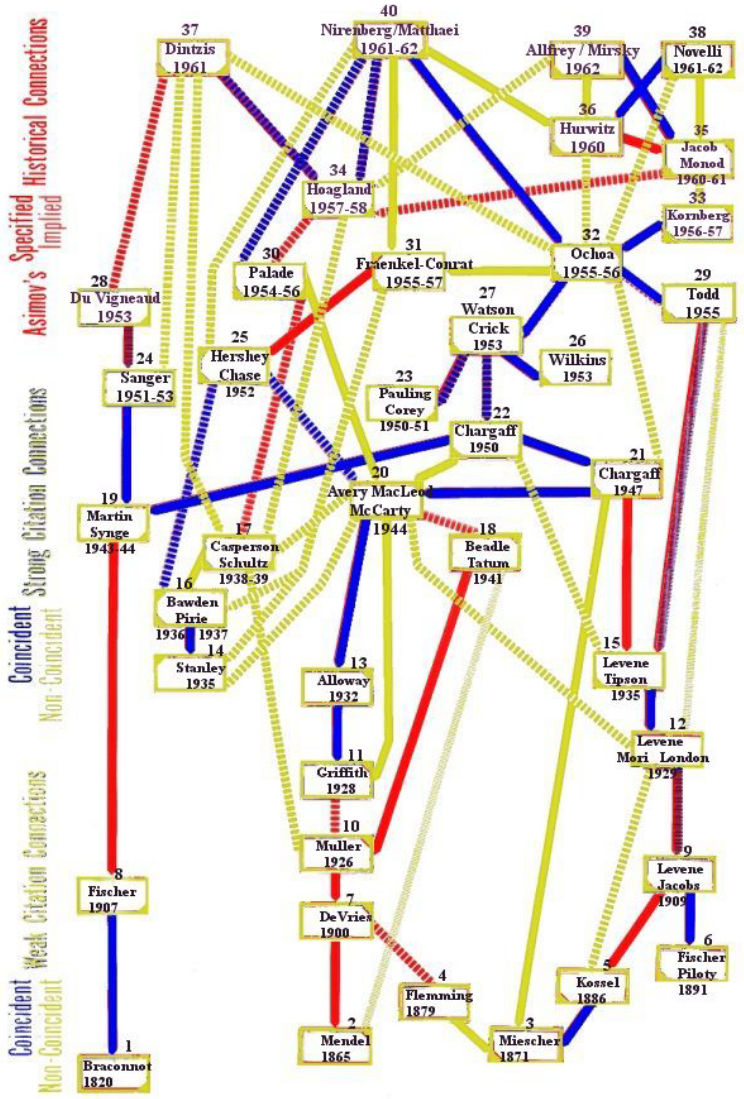
Naj bo $T(v)$ čas najzgodnejšega zaključka vseh opravil v stanju v.

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

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

`Net/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/točk $\mathcal{U}$ (članki, knjige, druga dela, itd.) vpeljemo *relacijo sklicevanja*/množico usmerjenih povezav $\mathbf{R} \subseteq \mathcal{U} \times \mathcal{U}$

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

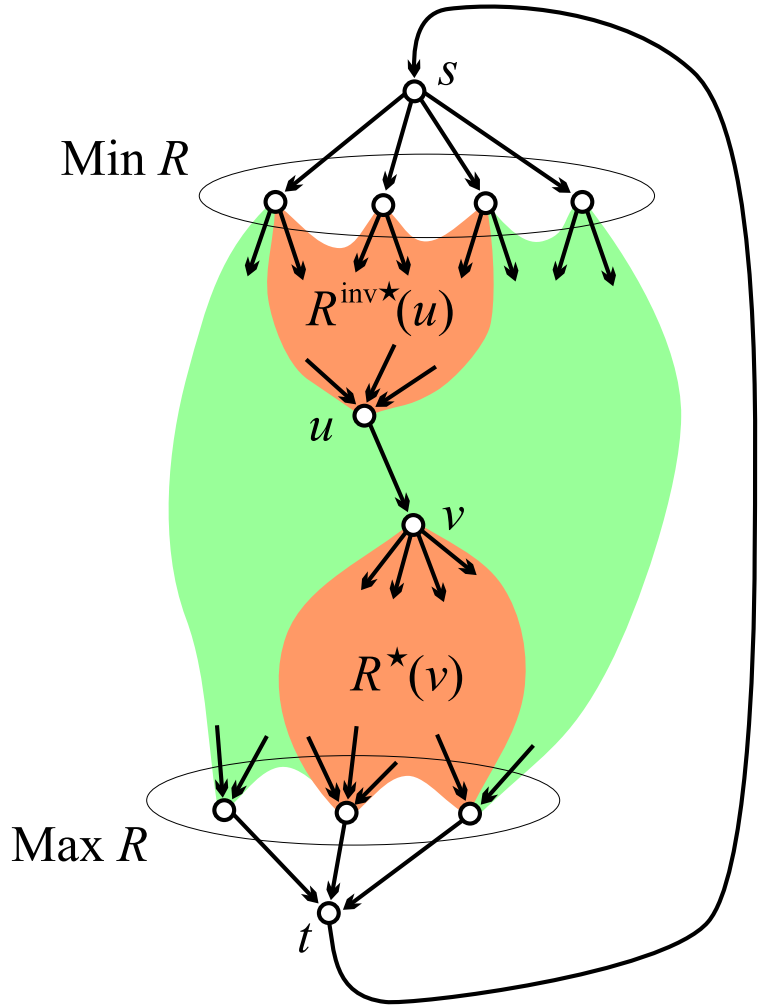
ki določa *omrežje sklicevanj* $\mathcal{N} = (\mathcal{U}, \mathbf{R})$.

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 točke) 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 točko v, in $n^+(v)$ šteje število poti iz točke 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

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

in

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

Ta zveza je osnova za hiter izračun števcev $n(u,v)$ – najprej točke 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 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 točk 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 točkami iz $\mathcal{U}$; za uteži SPNP moramo temu omrežju dodati še povezave iz vseh nemaksimalnih točk iz $\mathcal{U}$ v ponor t.

Točkovne uteži

Količine uporabljene za določitev povezavnih uteži w lahko uporabimo tudi za določitev ustreznih točkovnih 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 točko u.

V programu **Pajek** dobimo hkrati uteži w in lastnost t.

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

Lastnosti uteži SPC

Vrednosti števcev $n(u,v)$ določajo tok po omrežju sklicevanj – zanj velja *Kirchoffov točkovni zakon*: v vsaki točki 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.

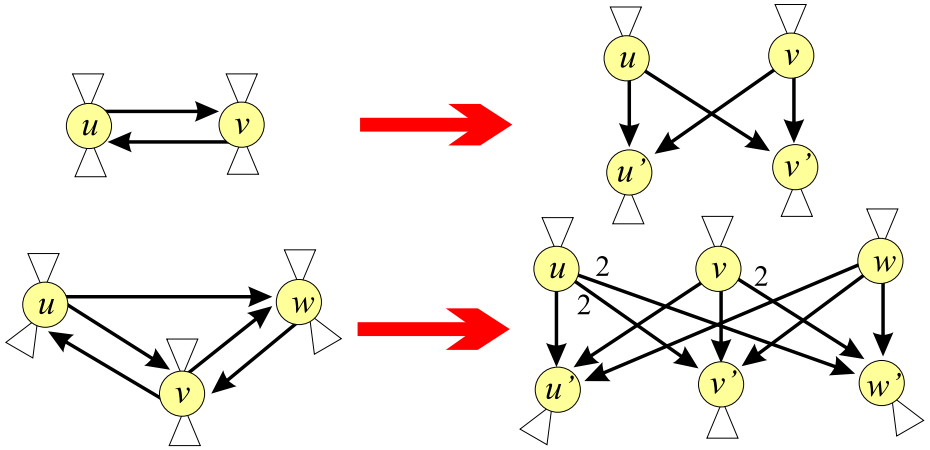
Omrežja sklicevanj s cikli

Če v omrežju obstaja cikel, obstaja med nekaterimi točkami 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 točko, ali
- razklenemo cikle z odstranitvijo nekaj povezav, ali
- z uporabo ustreznih transformacij (glej naslednjo prosojnico).

Transformacija Preprint



Transformacija *preprint* temelji na naslednji zamisli: Vsaka točka (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

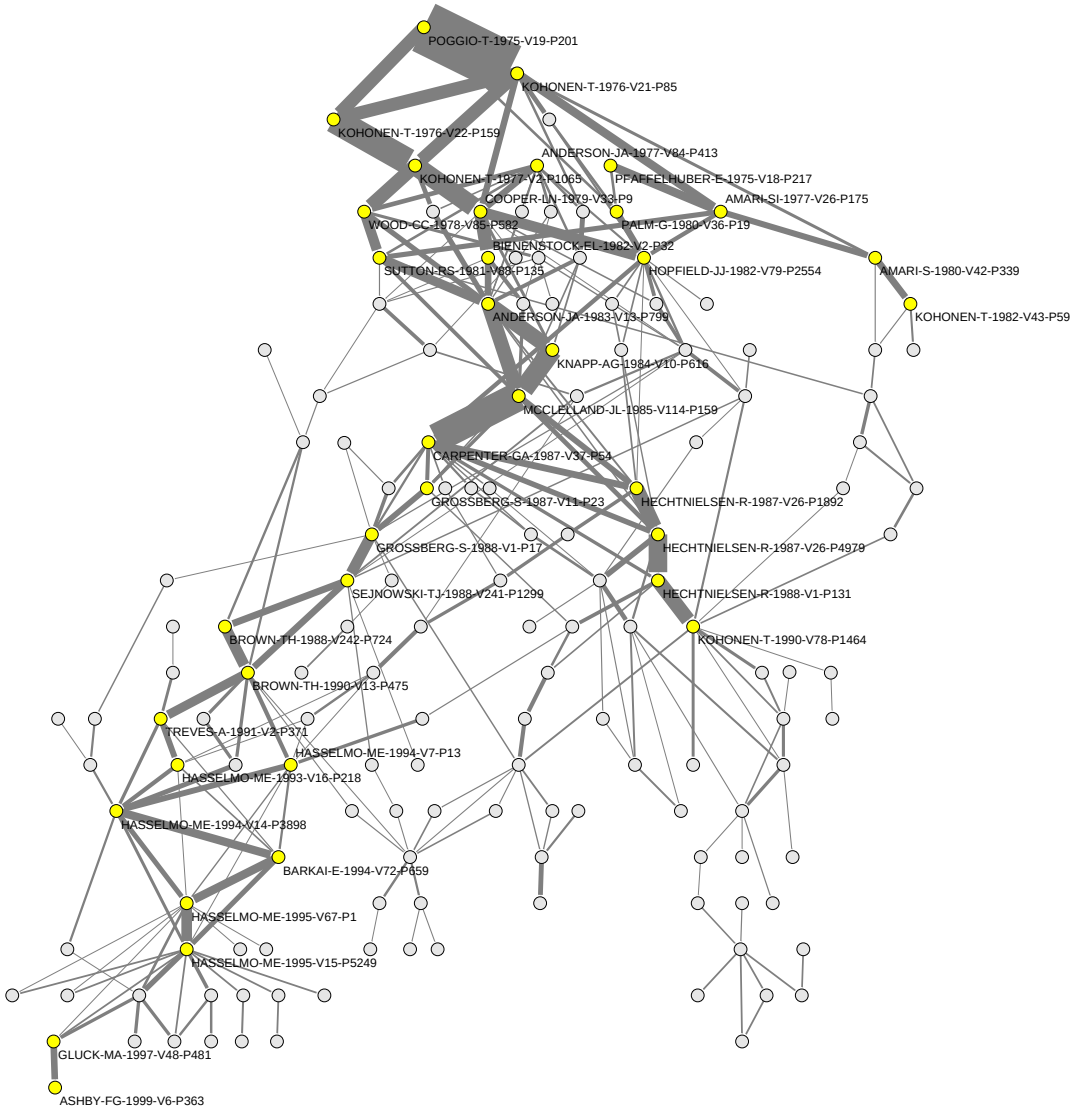
Poklejmo si omrežje sklicevanj za področje ($n = 4470$, $m = 12731$) SOM (*self-organizing maps*). Določimo uteži SPC.

Pri pregledu porazdelitve vrednosti uteži smo izbrali prag 0.007 in določili pripadajoči povezavni prerez. 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.

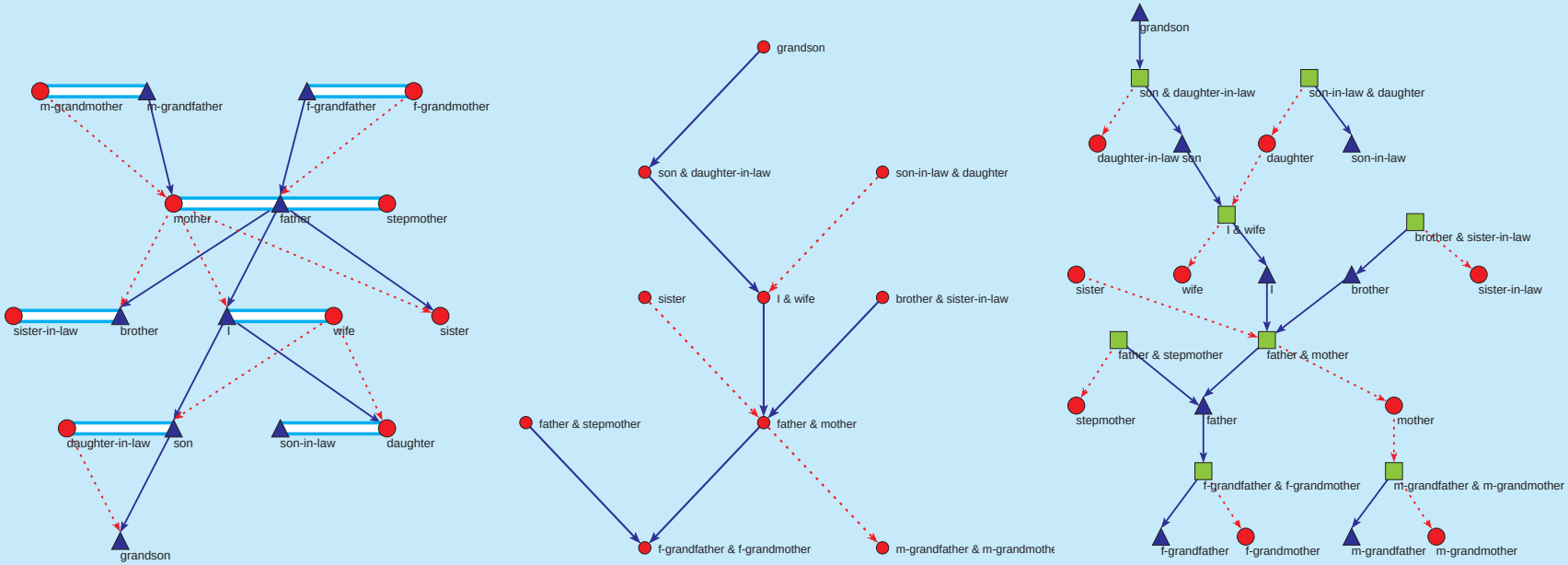
S slike vidimo, da v razvoju področja SOM obstajata vsaj dve smeri.

Omrežje SOM – povezavni prerez 0.007



Rodovniki

Naslednji primer acikličnih omrežij so rodovniki. V poglavju 'Omrežje 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 točk 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 točk.

Za *navaden Orejev rodovnik* velja ($\mathcal{V}$ – točke, $\mathcal{A}$ – 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 ($\mathcal{V}_p$, $\mathcal{A}_p$ – točke in povezave parnega grafa):

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

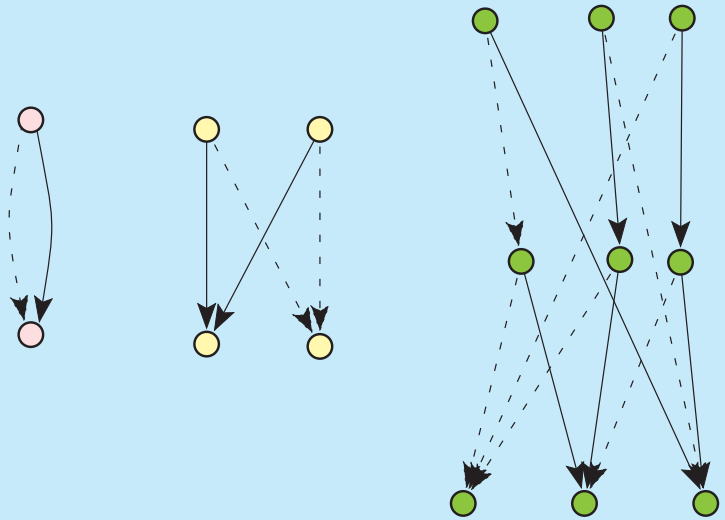
Za dvodelni parni graf pa velja:

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

Število točk in povezav v Orejevih in parnih grafih

data	$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	$\|\mathcal{A}\|$	$\frac{\|\mathcal{L}\|}{\|\mathcal{V}\|}$	$\|\mathcal{V}_i\|$	n_{mult}	$\|\mathcal{V}_p\|$	$\|\mathcal{A}_p\|$	$\frac{\|\mathcal{A}_p\|}{\|\mathcal{V}_p\|}$
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 točk v parnem grafu, m število povezav, k število šibkih komponent, in M število koncev (maksimalnih točk – točk 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

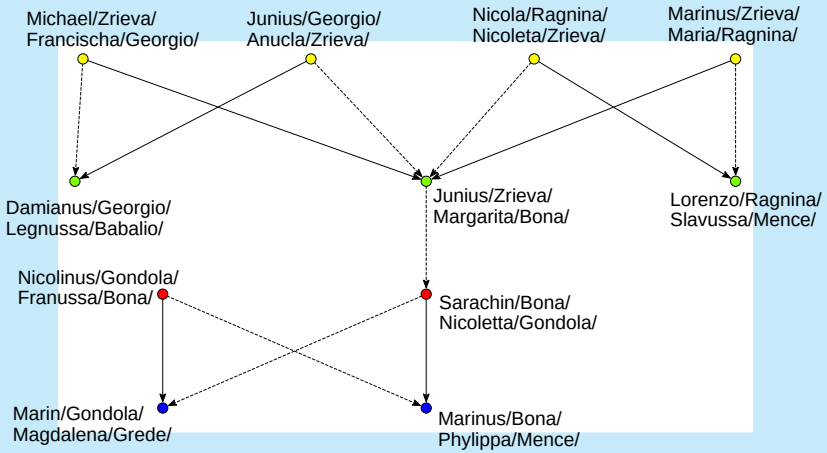
Če se izbrani *vorec* določen z danim (majhnim) grafom ne pojavlja pogosto v redkem grafu, lahko njegove pojavitve razmeroma učinkovito poiščemo s preprostim pregledom vseh možnosti.

Iskanje lahko pospešimo tako, da upoštevamo dodatne lastnosti vzorca:

- točke 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*);
- prva točka iz vzorca pripada dani skupini točk.

Poroke med sorodniki v Ragusi

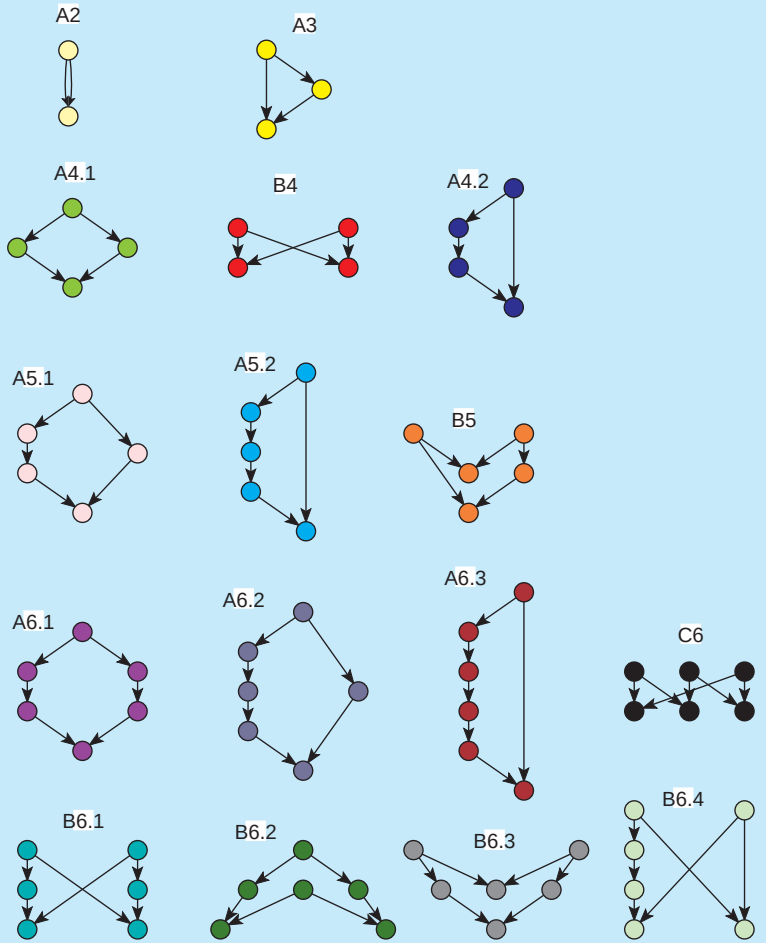
Iskanje vzorcev smo uporabili npr. pri analizi organskih molekul (iskanje ogljikovih obročev) in pri analizi porok med sorodniki v rodovnikih.



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.

`Nets/Fragment (First in Second)`

Poroke med sorodniki (parni grafi na 2 do 6 točkah).



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

- prvi znak – število začetkov: A – en, B – dva, C – tri.
- drugi znak: število točk 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.

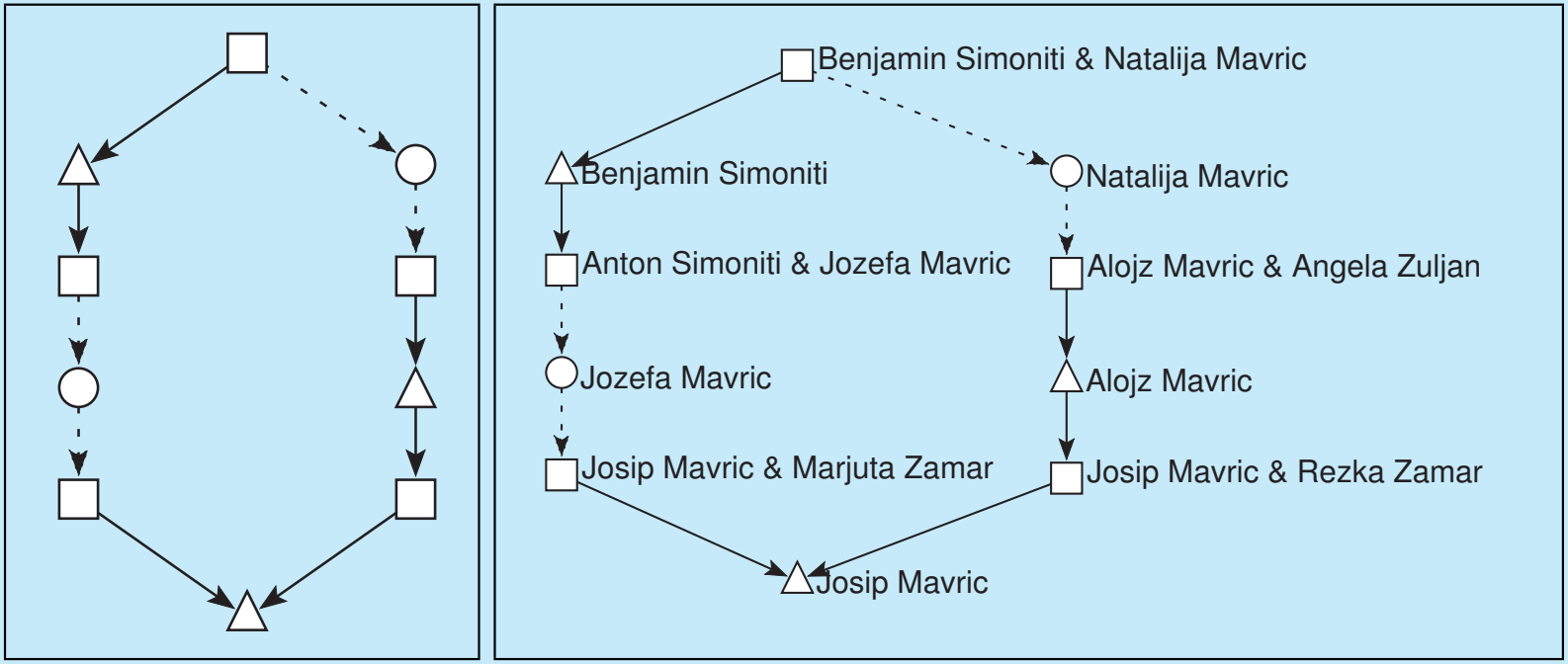
V vseh rodovnikih je število porok med nekrvnimi (vzorci B4, B5, C6, B6.1, B6.2, B6.3 in B6.4) sorodniki veliko večje od števila porok med krvnimi sorodniki. Razlogi za poroke med sorodniki so pogosto pogojeni s prizadevanji za ohranitev premoženja znotraj izbranih rodbin.

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



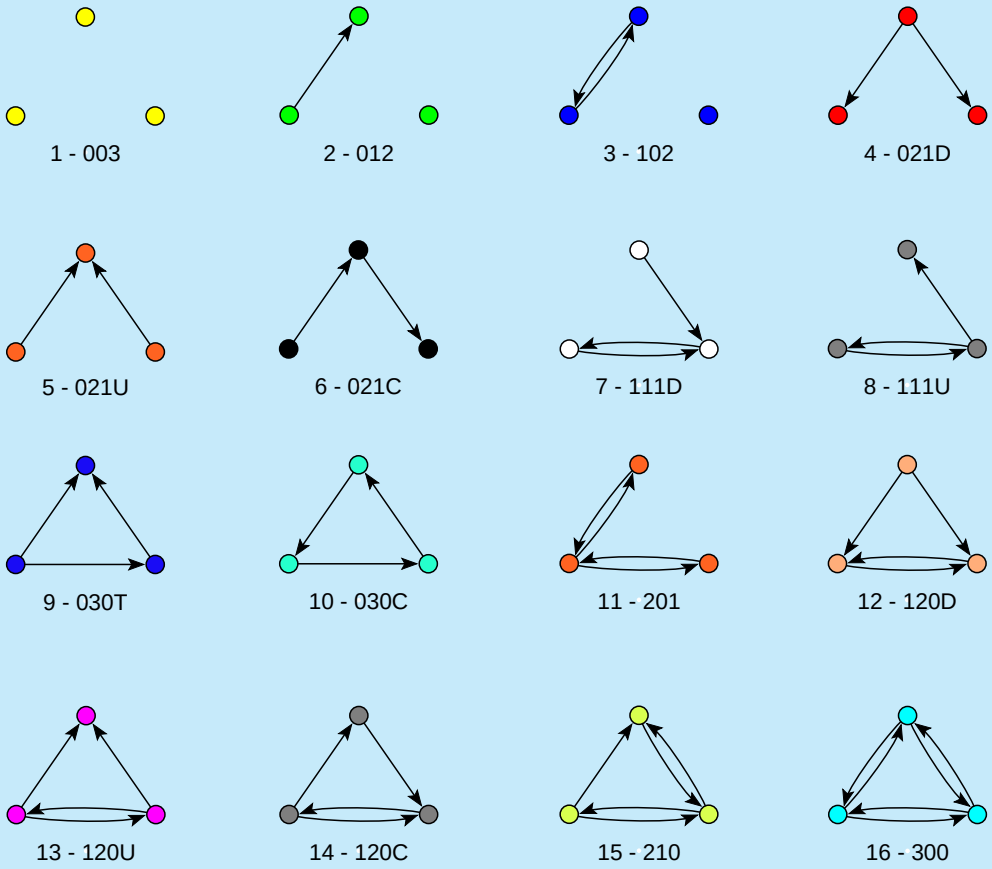
... rodovniki – druga vprašanja

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 rzlika 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 (prverjanje 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 točkami.

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. Točke 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 in lastnosti omrežij

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

Dvovrstna omrežja

Dvovrstno omrežje je omrežje $\mathcal{N} = (\mathcal{U}, \mathcal{V}, \mathcal{A}, w)$, v katerem je množica točk 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 $\mathbf{A} = [a_{uv}]_{\mathcal{U} \times \mathcal{V}}$.

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

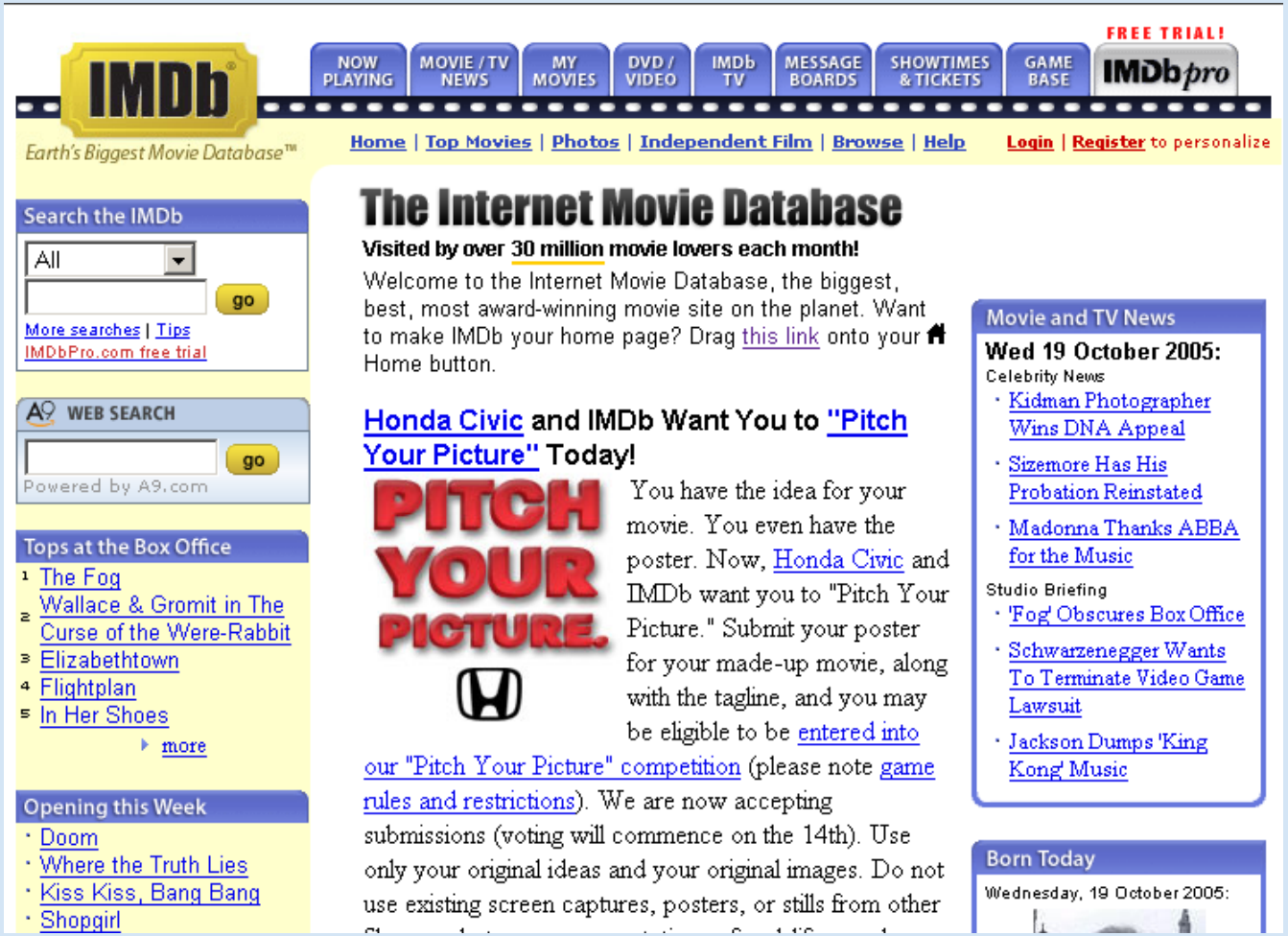
Pristopi k analizi dvovrstnih omrežij

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 $\mathbf{x} = \mathbf{A}^{\mathsf{T}}\mathbf{y}$.

`Net/Vector/Important Vertices/2-Mode: Important Vertices`

Novejša pristopa sta: *dvovrstne sredice* in *4-obroči*. V naslednjem predavanju bomo spoznali še uporabo *razvrščanja* in *bločneh 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 $1324748 = 428440 + 896308$ točk ter 3792390 povezav.

Dvovrstne sredice

Podmnožica točk $C \subseteq \mathcal{V}$ 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 vse točke s trenutno stopnjo manjšo od p, in

iz množice $\mathcal{V}_2$ odstrani vse točke s trenutno stopnjo manjšo od q

until nobena točka ni bila odstranjena;

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 $\mathcal{K}(p, q)$:

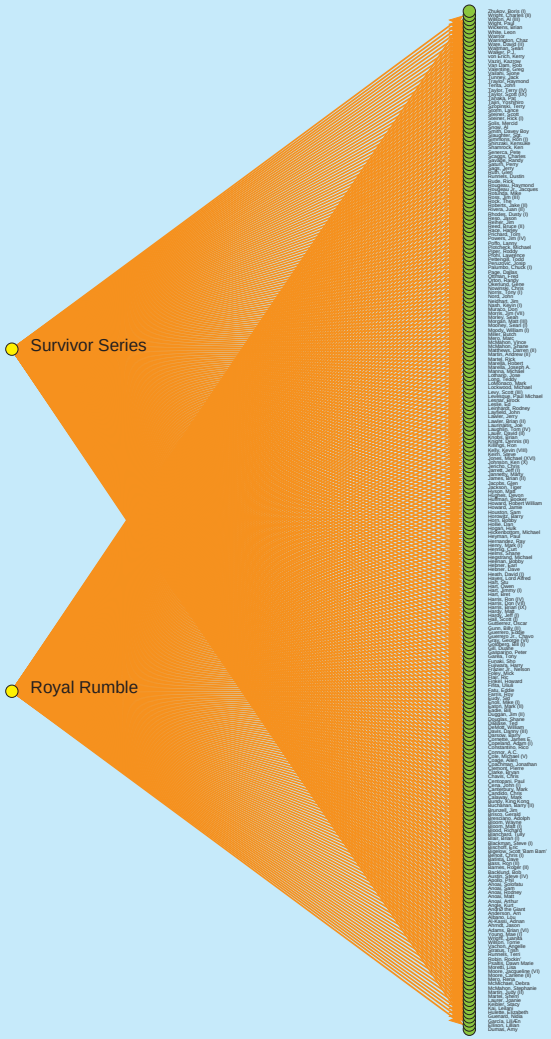
- $n_1 + n_2 \leq$ izbrano zgornjo mejo
- večji skoki z $C(p-1, q)$ ali $C(p, q-1)$ na $C(p, q)$.

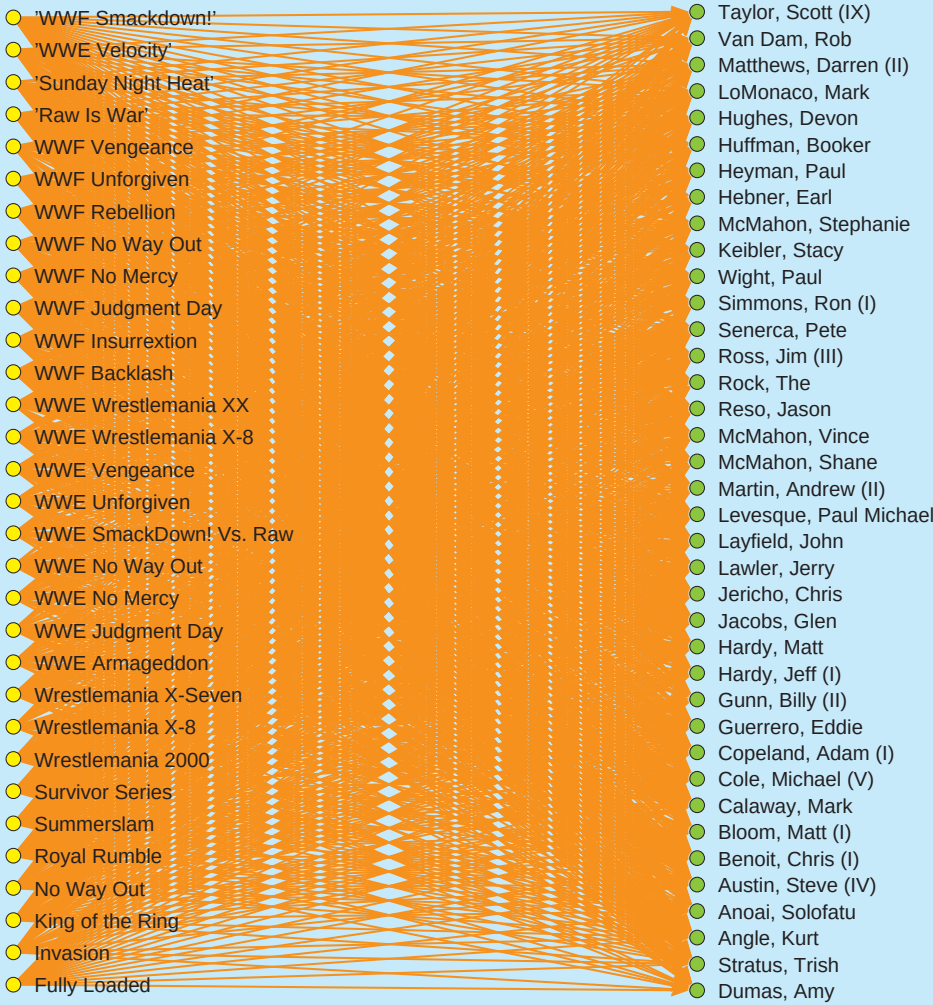
`Net/Partitions/Core/2-Mode Border`

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

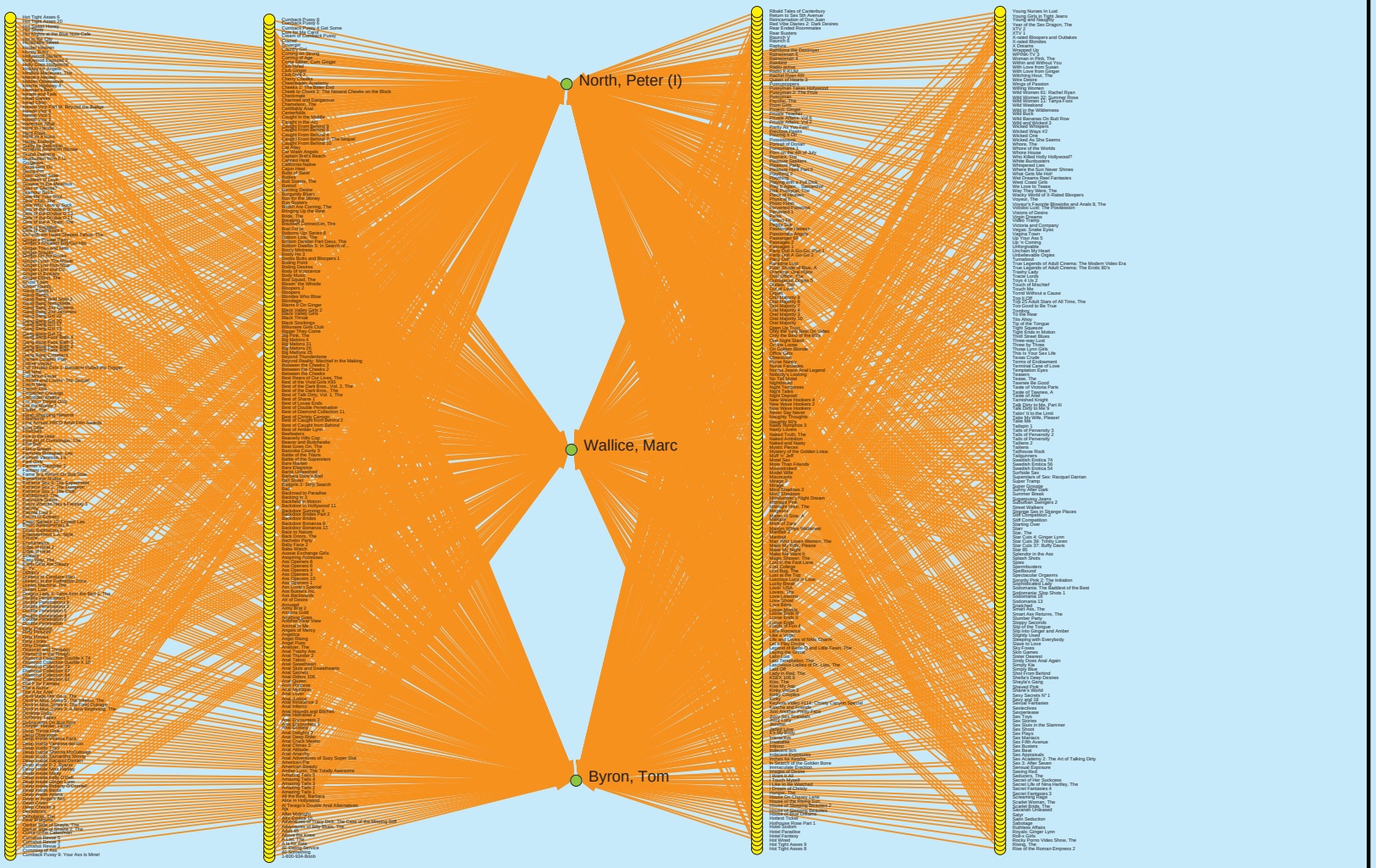
1	1590:	1590	1	22	24:	1854	1153	43	14:	29	83
2	516:	788	3	23	23:	47	56	44	14:	29	83
3	212:	1705	18	24	23:	34	39	45	13:	30	95
4	151:	4330	154	25	22:	42	53	46	13:	29	94
5	131:	4282	209	26	22:	31	38	47	12:	29	101
6	115:	3635	223	27	22:	31	38	48	12:	28	100
7	101:	3224	244	28	20:	36	53	49	12:	26	95
8	88:	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



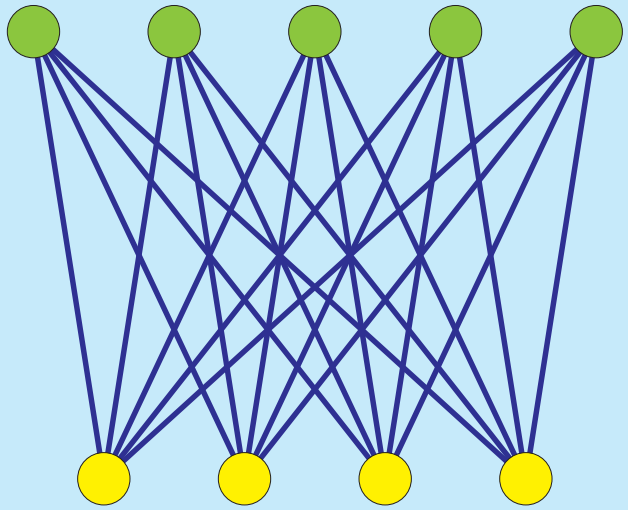


(2,516)-Hard core



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.



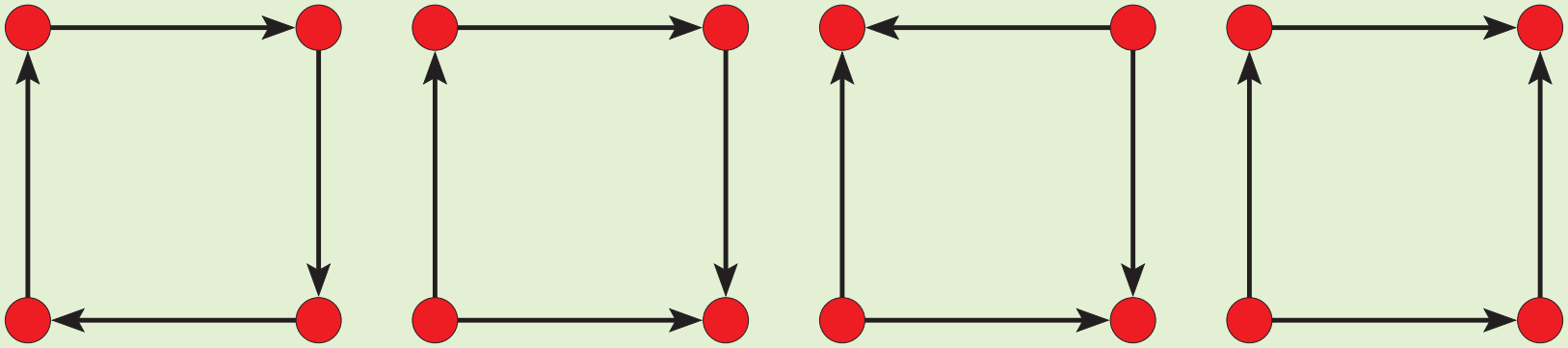
$$w_4(K_{p,q}) = (p-1)(q-1)$$

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

`Net/Count/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.

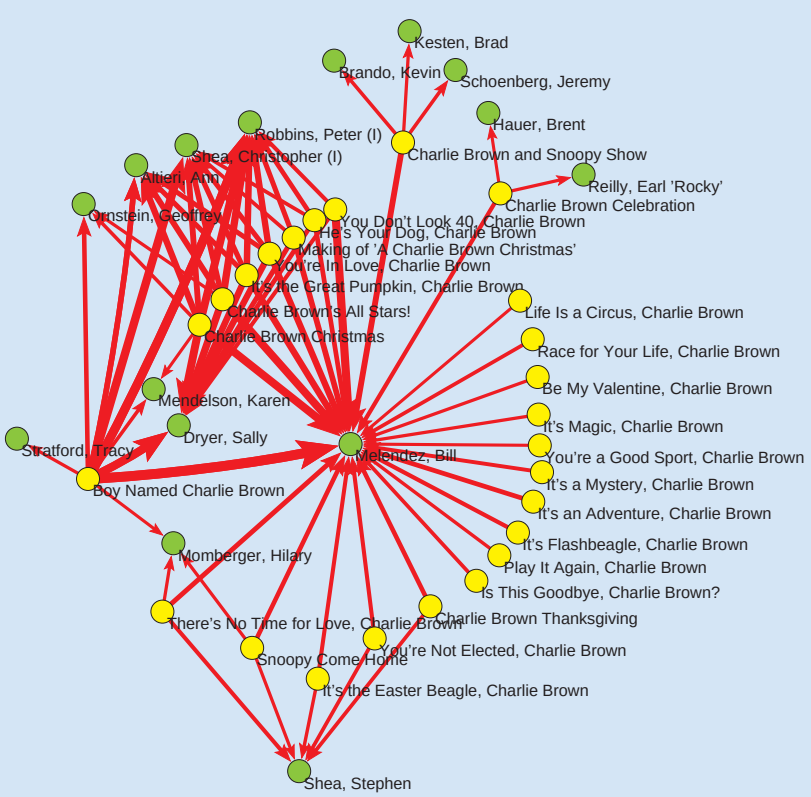
`Net/Count/4-Rings/Directed`

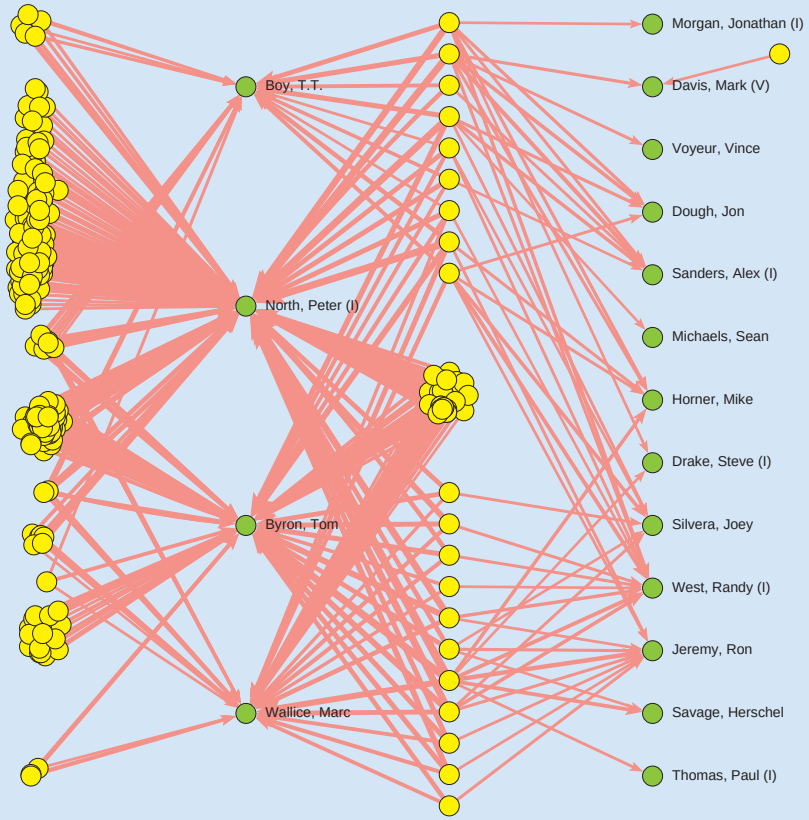
Enostavni povezavni otoki v IMDB za w_4

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

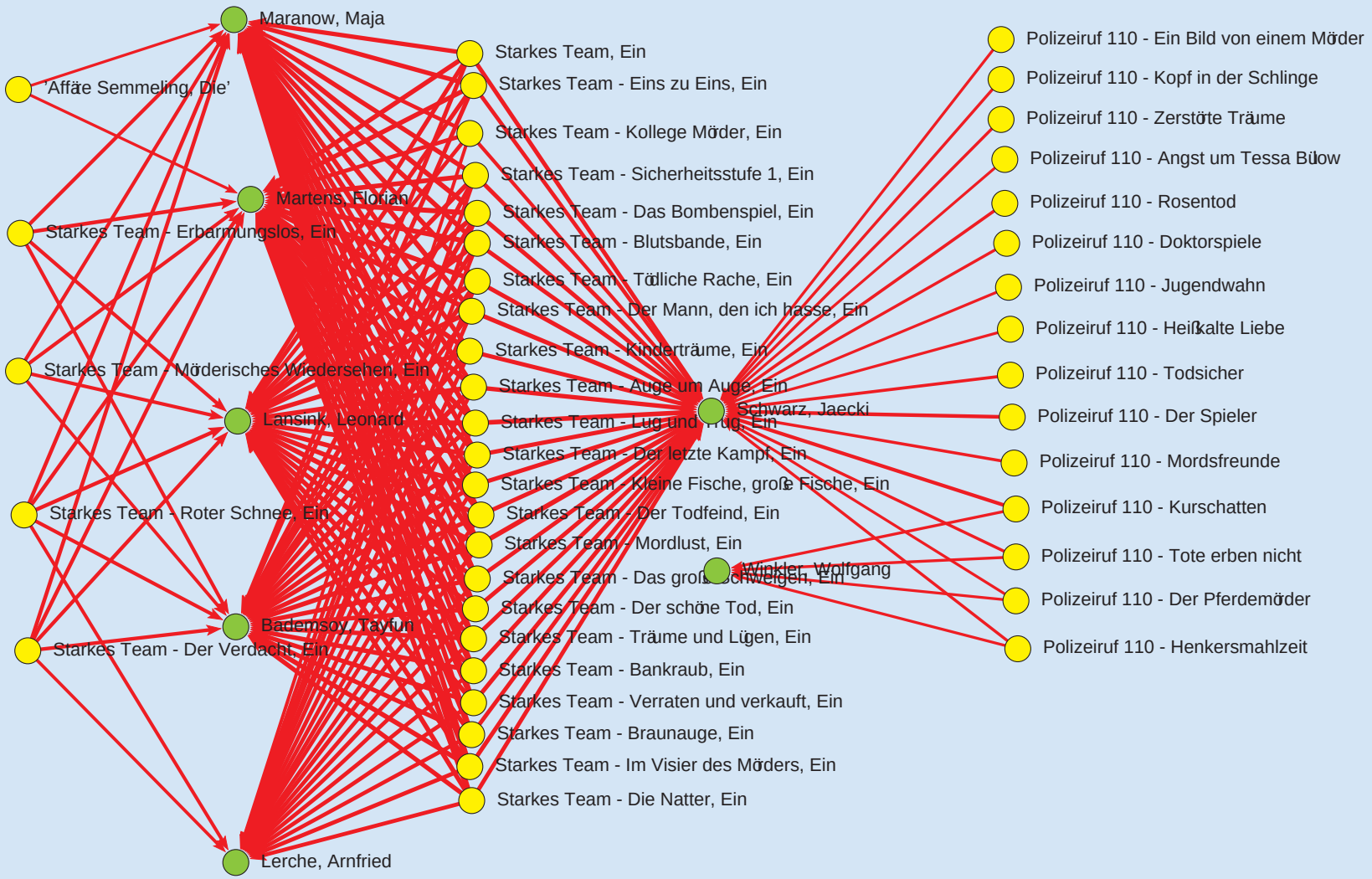
Size	Freq	Size	Freq	Size	Freq	Size	Freq
2	5512	20	19	38	4	59	2
3	1978	21	18	39	3	61	1
4	1639	22	15	40	2	64	1
5	968	23	9	42	2	67	1
6	666	24	13	43	3	70	1
7	394	25	12	45	3	73	1
8	257	26	6	46	4	76	1
9	209	27	6	47	5	82	1
10	148	28	5	48	1	86	1
11	118	29	6	49	2	106	1
12	87	30	3	50	2	122	1
13	55	31	6	51	1	135	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



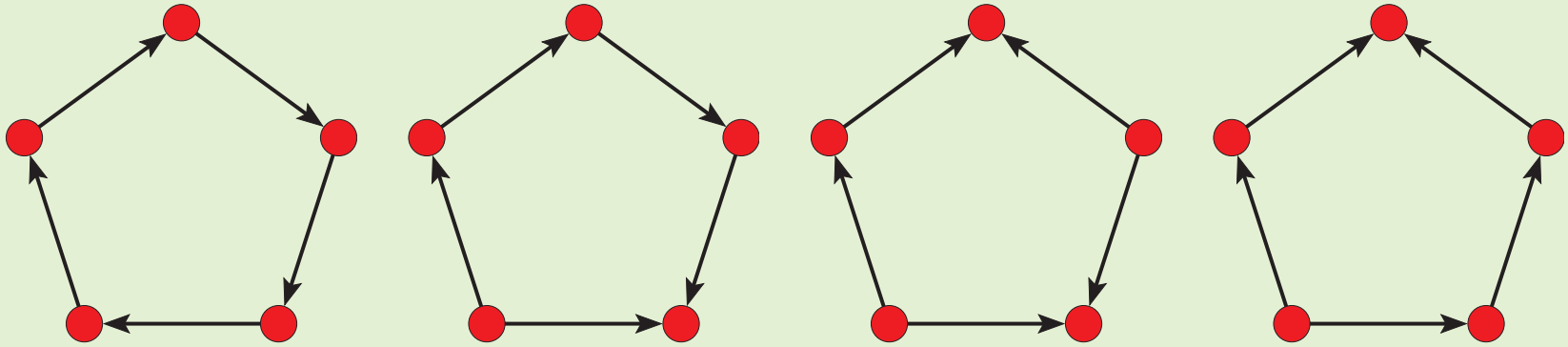


Primer: Otok za w_4 / Polizeiruf 110 in Starkes Team



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.



Množenje omrežij

Naj bo $\mathcal{N} = (\mathcal{I}, \mathcal{J}, \mathcal{E}, w)$ enostavno dvovrstno omrežje nad množicama točk $\mathcal{I}$ 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).

Običajno množenje matrik

Običajni postopek za izračun produkta matrik $\mathbf{C}$

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

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 točke k v omrežju $\mathcal{N}_A$

$N_B(k)$: sosedi točke k v omrežju $\mathcal{N}_B$

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

`Nets/Multiply First * Second`

Zahtevnost hitrega množenja

Bodita $\mathbf{A}$ in $\mathbf{B}$ 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. Tedaj je zahtevnost izračuna produkta enaka

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

Naj bo $\Delta^A_{\mathcal{K}} = \max_{k \in \mathcal{K}} \deg_A(k)$ in $\Delta^B_{\mathcal{K}} = \max_{k \in \mathcal{K}} \deg_B(k)$ ter upoštevajmo znano enakost

$$\sum_{k \in \mathcal{K}} \deg_A(k) = \sum_{i \in \mathcal{I}} \deg_A(i) = |\mathcal{E}_A|$$

Dobimo $C \leq c \cdot \min(|\mathcal{E}_A| \cdot \Delta^B_{\mathcal{K}}, |\mathcal{E}_B| \cdot \Delta^A_{\mathcal{K}})$.

Če ima vsaj eno od redkih omrežij $\mathcal{N}_A$ in $\mathcal{N}_B$ majhno stopnjo na množici $\mathcal{K}$, je tudi produktno omrežje $\mathcal{N}_C$ redko.

Podrobnejša analiza zahtevnosti

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 $\mathcal{K}^* = \mathcal{K}(d^*)$. Potem je $|\mathcal{K}^*| \leq d^*$ in

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

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

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

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

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.

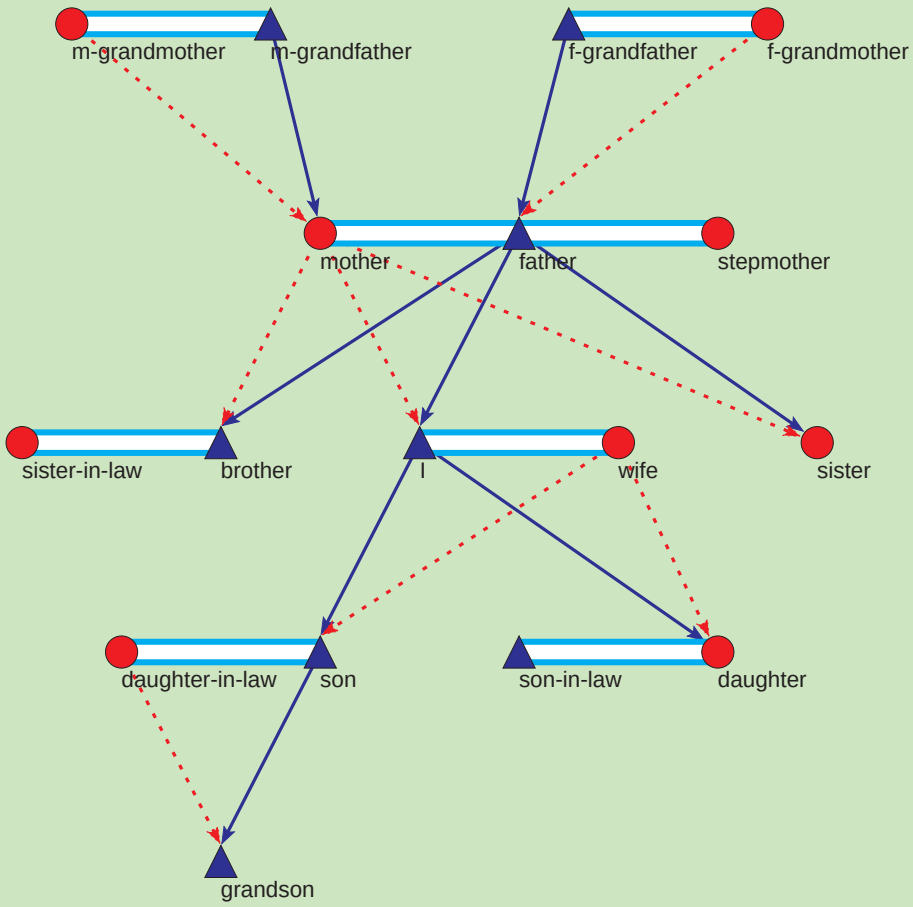
Primer: Sorodstvene vezi

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 s točko,
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

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	Torin	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 20	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. V	61-712	502235	Pozn	POLSKA	45	2	
13	12	6571	135.html	A.BRITO - INDUSTRIA	A.BRI	TO	VIEIRA DE BRITO	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	Københavı
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/FI	53	2	MANNER-
19	18	8152	662.html	AABO AKADEMI UNI	AAB	DEI	BJORKSTRAND,	3,Tykis	20521	EVK1-CT-2002	Turku	SUOMI/FI	53	2	
20	19	8148	959.html	AABO AKADEMI UNI	AAB	Dep	HUPA, Mikko	Domky	20500	502679	Turku	SUOMI/FI	53	2	MANNER-
21	20	8151	233.html	AABO AKADEMI UNI	AAB	DEI	NYBACKA-WILLI	Lemmi	20500	ERK6-CT-1999	Turku	SUOMI/FI	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 $\mathcal{T}$ 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.

. . . 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 $w(u, v) = \text{card}\,\{k \in \mathcal{K} : (q_i(k) = u \wedge q_j(k) = v)\}$.

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 $[\mathbf{q}_i \times \mathbf{q}_j]/\mathbf{q}_s = [\mathbf{q}_i \times \mathbf{q}_s] * [\mathbf{q}_s \times \mathbf{q}_j]$.

Primer: 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 $= \mathbf{P}$
- `country.net` – idents $\times$ countries $= \mathbf{C}$
- `institution.net` – idents $\times$ institutions $= \mathbf{U}$

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

Evropski projekti – množenje omrežij

Ker imajo vsa tri omrežja skupno množico $\mathcal{K}$ =idents, lahko na prej opisani način iz njih z množenjem pridobimo različna 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 $\mathbf{Q} = \mathbf{W}^T \star \mathbf{W}$
- ...

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:

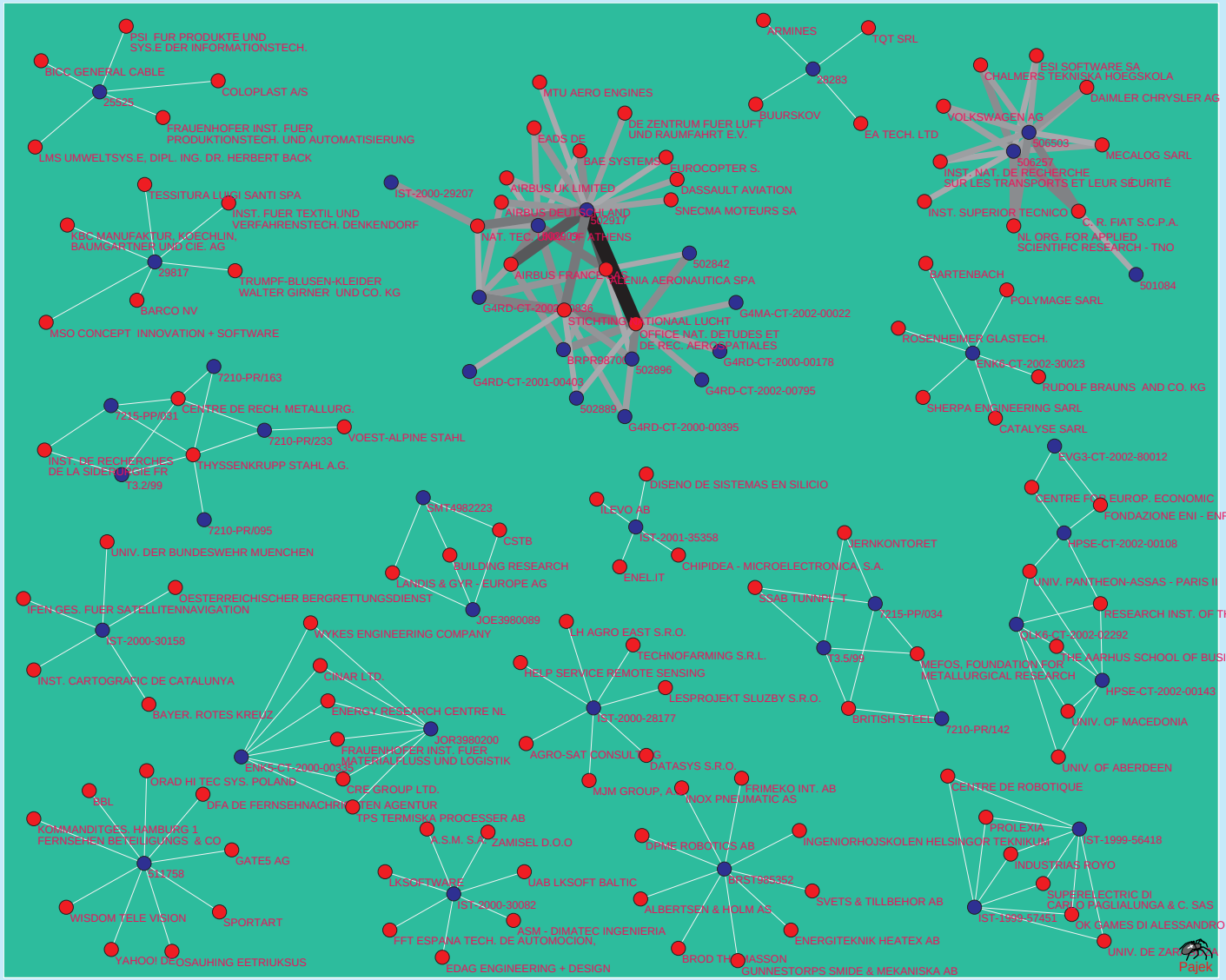
```
Net/Count/4-rings/Undirected
Net/Partitions/Islands/Line Weights[Simple [2,200]
```

Dobili smo 101 otok, 18 med njimi ima velikost vsaj 5 (točk). Najpomembnejša otoka sestavljajo letalske ustanove in avtomobilske ustanove.

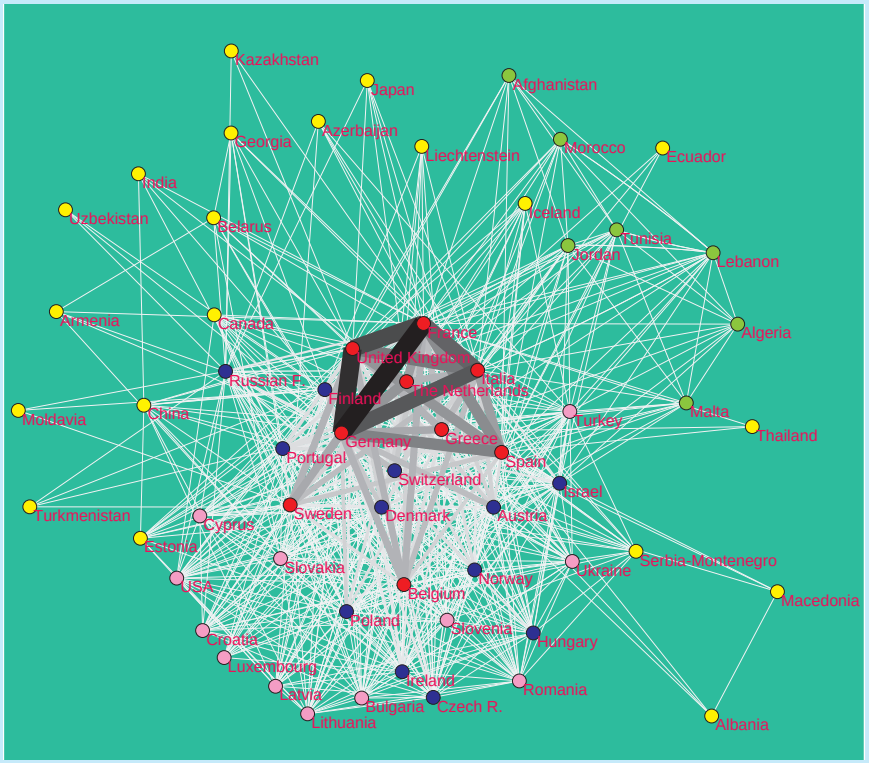
V zapisu nekaterih oznak točk smo uporabili `\n`, ki oznako prelomi.

Za analizo bi lahko uporabili tudi $(p, q)-$sredice.

Analiza omrežja ProjInst.net



Analiza omrežja `Countries.net`



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

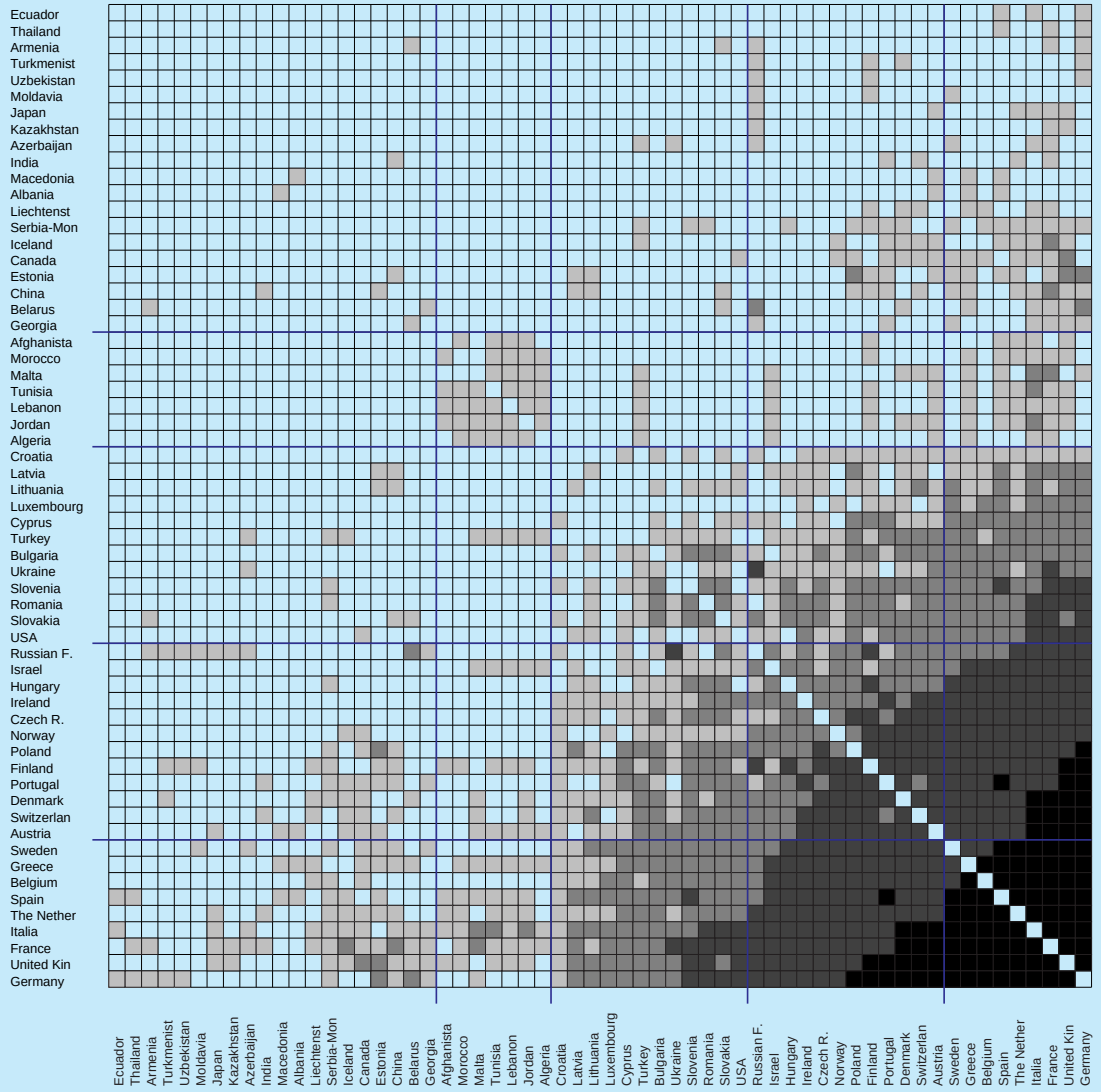
`Net/Transform/Sort lines/Line values/Ascending`

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).

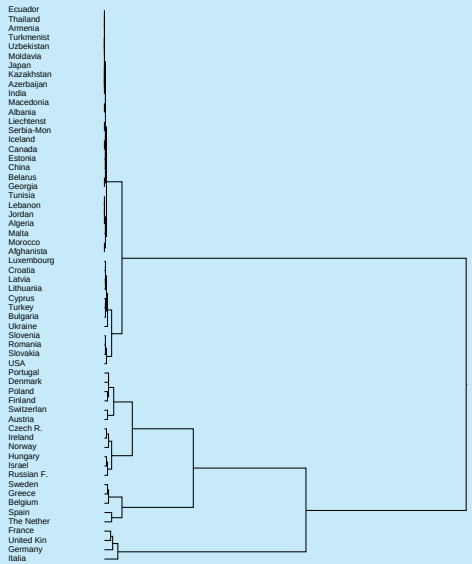
Urejenost lahko še izboljšamo s preurejanjem poddreves v hierarhiji. Kakor vidimo na sliki, dobimo značilno (večslojno) zgradbo središče – obrobje.

Analiza omrežja Countries.net

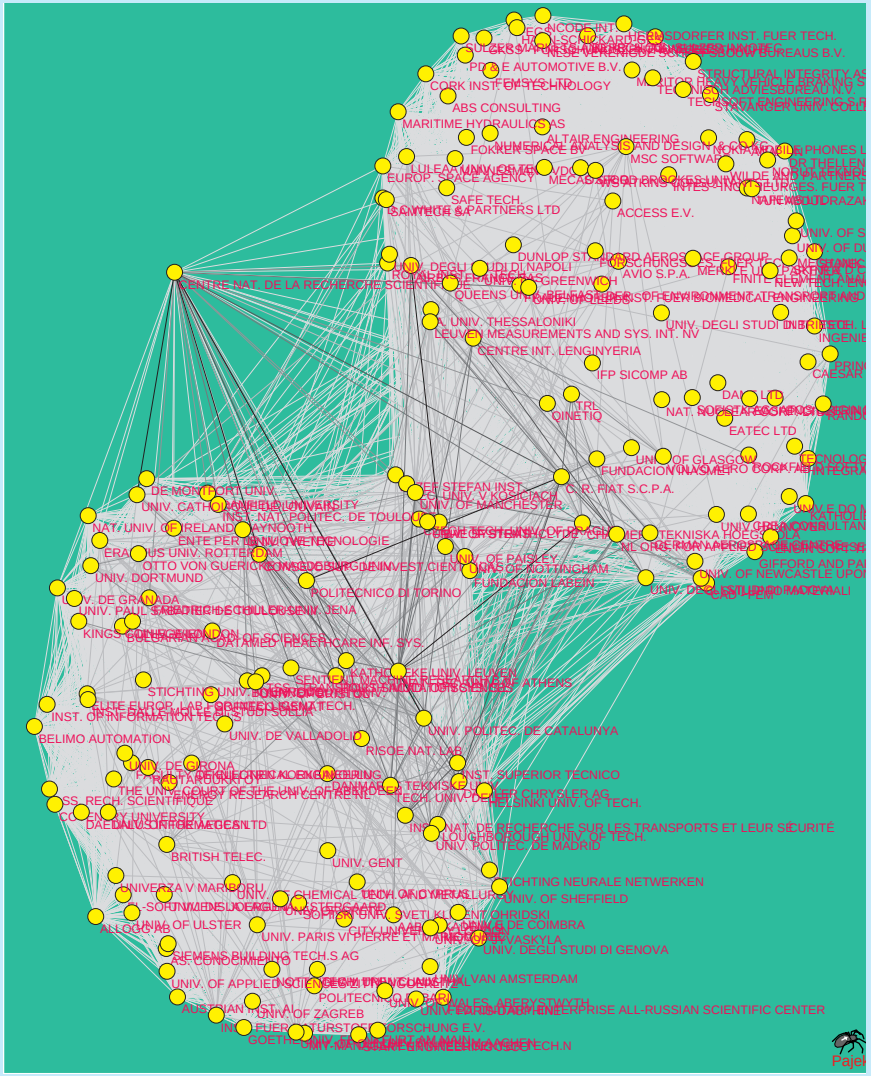
Pajek - shadow [0.00,4.00]



Pajek - Ward [0.00,4785.14]

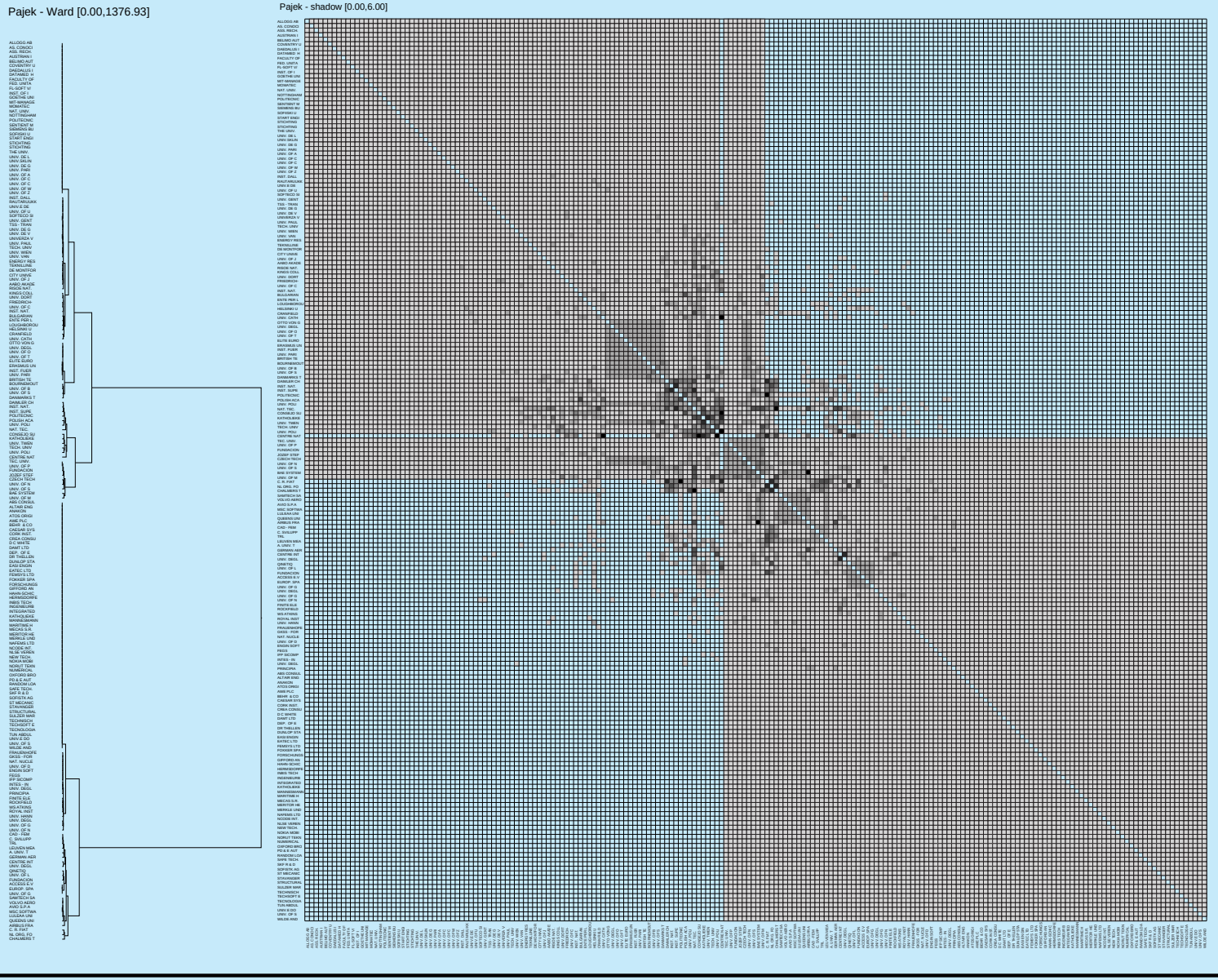


Analiza omrežja Institutions.net



Za določitev najpomembnejših ustanov smo najprej določili sredice vrste p_S in za tako določen vektor določili točkovne 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 odstopajoča točka 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 točk 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 točk.

. . . Normalizacije

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

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

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

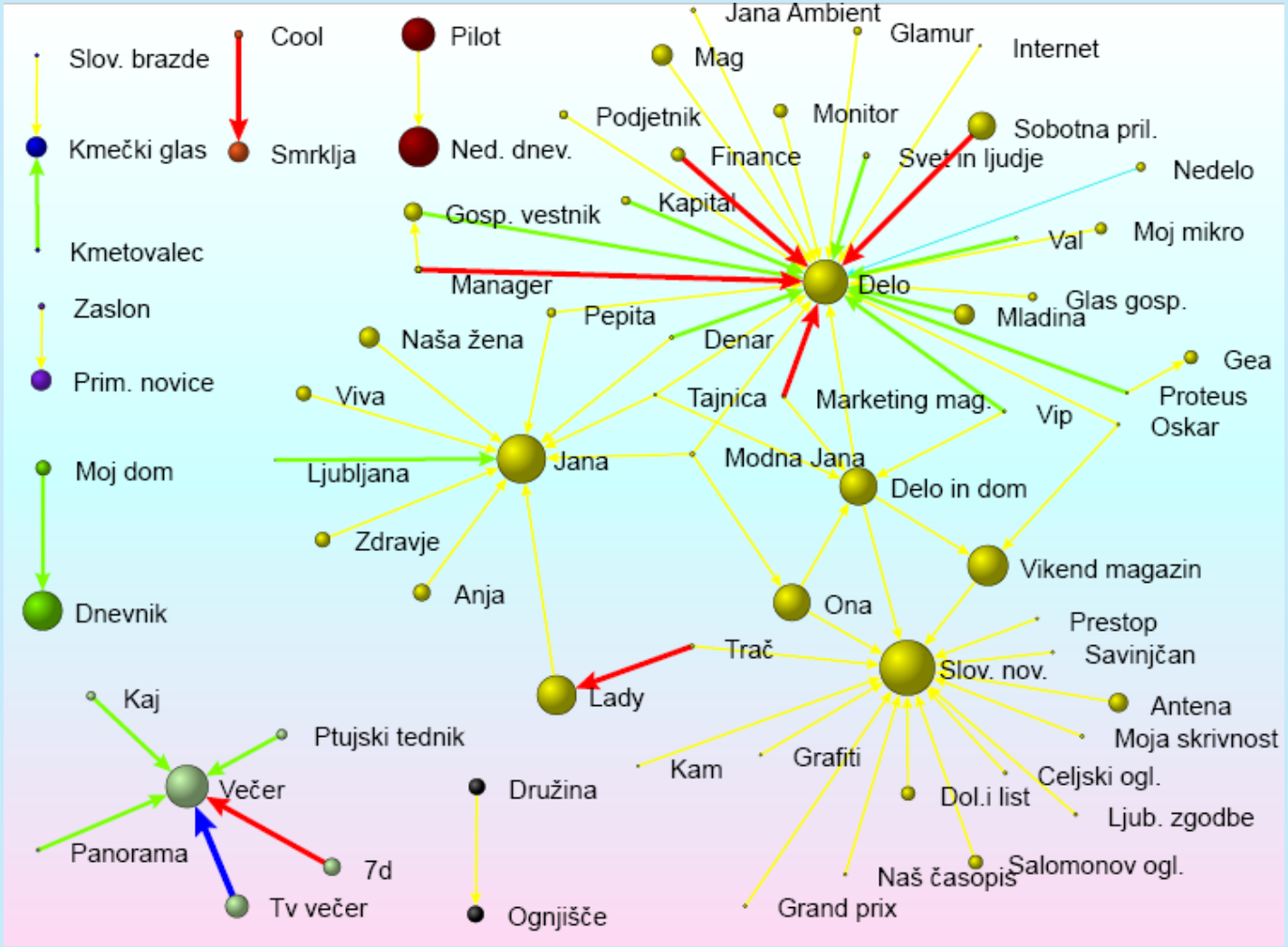
$$\text{MinDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{uu}} & w_{uu} \leq w_{vv} \\ 0 & \text{sicer} \end{cases} \qquad \text{MaxDir}_{uv} = \begin{cases} \frac{w_{uv}}{w_{vv}} & w_{uu} \leq w_{vv} \\ 0 & \text{sicer} \end{cases}$$

Normalizirano omrežje analiziramo z uporabo povezavnih prerezov ali otokov.

`Net/Transform/2-Mode to 1-Mode/Normalize 1-Mode/`

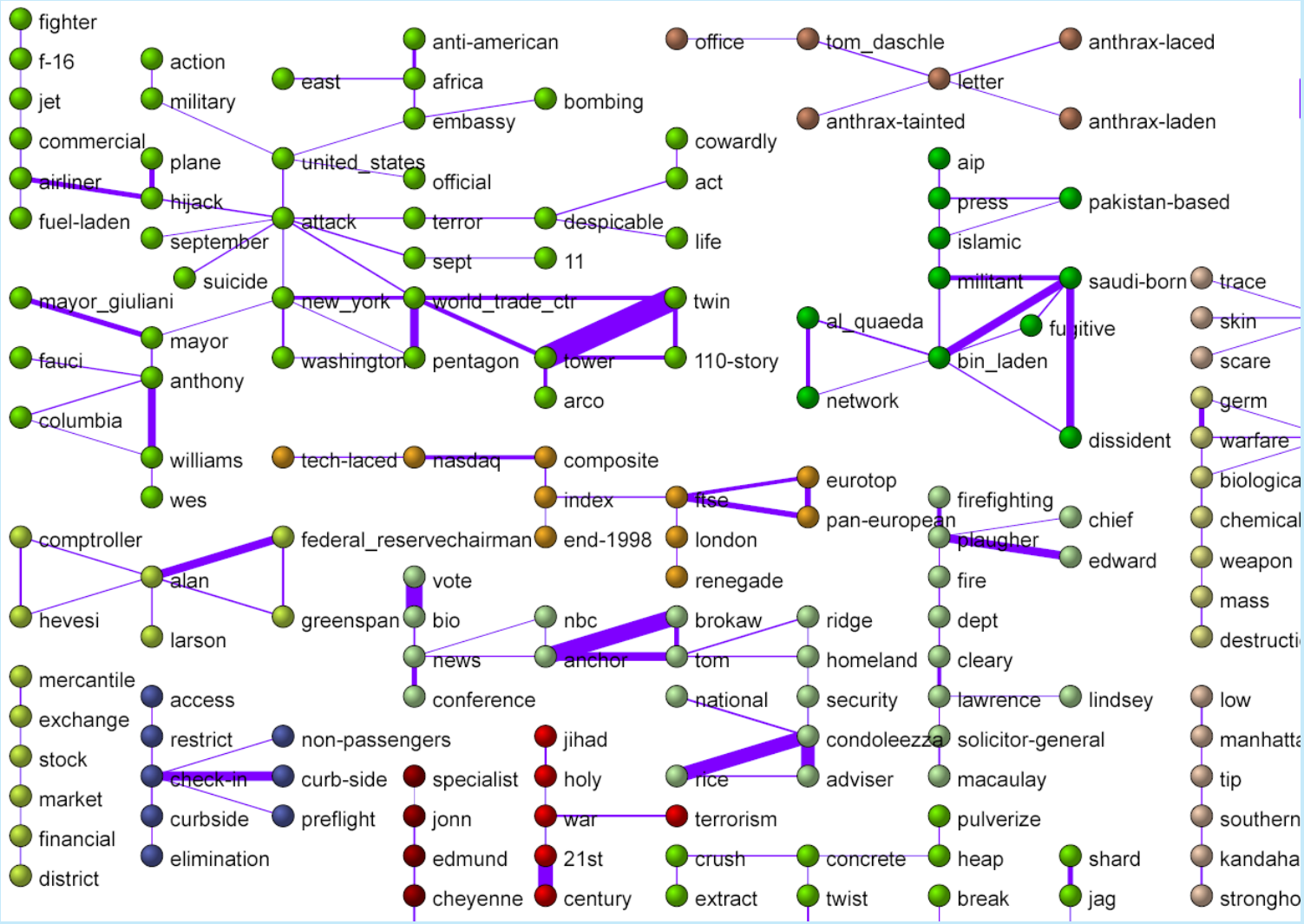
Reuters Terror News: GeoDeg, MaxDir, MinDir.

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