Photo: V. Batagelj

Analysis of Large Networks with Pajek

Vladimir Batagelj
University of Ljubljana

Workshop

Sunbelt XXVIII, St. Pete Beach, Florida, USA, 22-27 January, 2008

version: January 21, 2008 / 02 : 39

Outline

Networks

A *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of:

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of vertices, $\mathcal{A}$ is the set of arcs, $\mathcal{E}$ is the set of edges, and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of links. $n = \text{card}(\mathcal{V})$, $m = \text{card}(\mathcal{L})$
- $\mathcal{P}$ vertex value functions / *properties*: $p : \mathcal{V} \to A$
- $\mathcal{W}$ line value functions / *weights*: $w : \mathcal{L} \to B$

In November 1996 we started the development of **`Pajek`** – a program, for analysis and visualization of *large networks*. The latest version of **`Pajek`** is freely available, for noncommercial use, at its home page:

`http://vlado.fmf.uni-lj.si/pub/networks/pajek/`

de Nooy, W., Mrvar, A. and Batagelj V.: *Exploratory Social Network Analysis with Pajek*, CUP,2005.

Large Networks

Networks are used in social sciences from thirties (Moreno). Most networks collected till 1990 are *small* (some tens of vertices). Development of IT in nineties enabled collection of *large* networks – several thousands or millions of vertices. Large networks are usually sparse $m \ll n^2$.

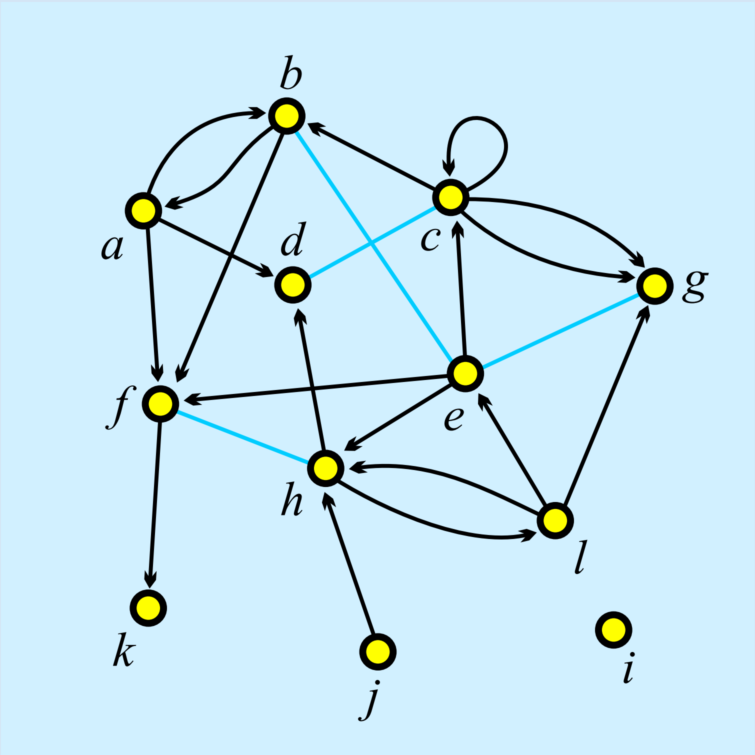
network	size	$n = \|V\|$	$m = \|L\|$	source
ODLIS dictionary	61K	2909	18419	ODLIS online
Citations SOM	168K	4470	12731	Garfield's collection
Molecula 1ATN	74K	5020	5128	Brookhaven PDB
Comput. geometry	140K	7343	11898	BiBTEX bibliographies
English words 2-8	520K	52652	89038	Knuth's English words
Internet traceroutes	1.7M	124651	207214	Internet Mapping Project
Franklin genealogy	12M	203909	195650	Roperld.com gedcoms
World-Wide-Web	3.6M	325729	1497135	Notre Dame Networks
Actors	3.9M	392400	1342595	Notre Dame Networks
US patents	82M	3774768	16522438	Nber
SI internet	38M	5547916	62259968	Najdi Si

Approaches to large networks

In analysis of a *large* network (several thousands or millions of vertices, the network can be stored in computer memory) we can't display it in its totality; also there are only few algorithms available.

To analyze a large network we can use statistical approach or we can identify smaller (sub) networks that can be analyzed further using more sophisticated methods.

Degrees



degree of vertex v, $\deg(v)$ = number of lines with v as end-vertex;

indegree of vertex v, $\text{indeg}(v)$ = number of lines with v as terminal vertex (end-vertex is both initial and terminal);

outdegree of vertex v, $\text{outdeg}(v)$ = number of lines with v as initial vertex.

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

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

Pajek and R

Pajek 0.89 (and later) supports the use of external programs (menu `Tools`). It provides a special support for statistical program R.

In **Pajek** we determine the degrees of vertices and submit them to R

```
info/network/general
Net/Partitions/Degree/All
Partition/Make Vector
Tools/Program R/Send to R/Current Vector
```

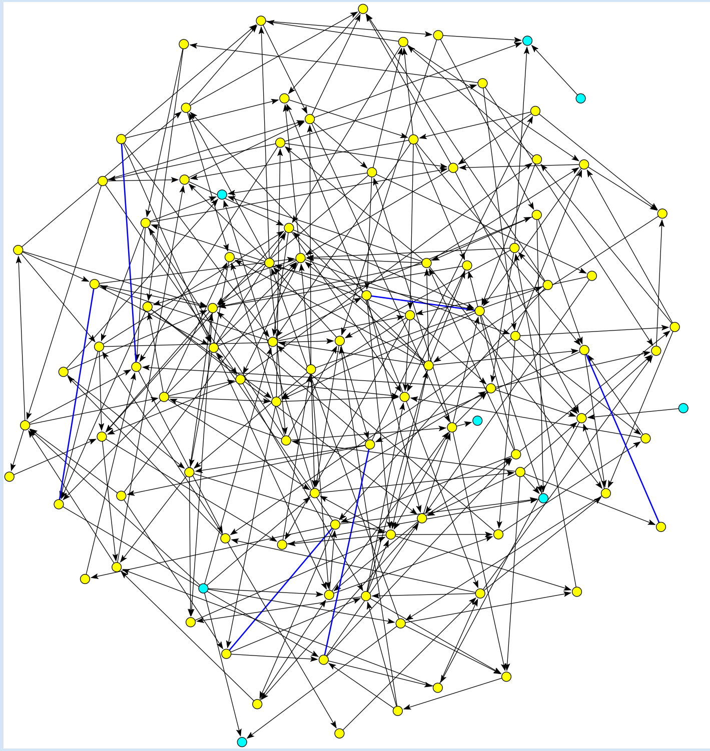
In R we determine their distribution and plot it

```
summary(v2)
t <- tabulate(v2)
c <- t[t>0]
i <- (1:length(t))[t>0]
plot(i,c,log='xy',main='degree distribution',
  xlab='deg',ylab='freq')
```

Attention! The vertices of degree 0 are not considered by `tabulate`. Use

`t <- tabulate(v2+1)`

Erdős and Rényi's random graphs



Erdős and Rényi defined a *random graph* as follows: every possible line is included in a graph with a given probabilty p.

In **Pajek**'s

`Net/Random Network/Erdos-Renyi`

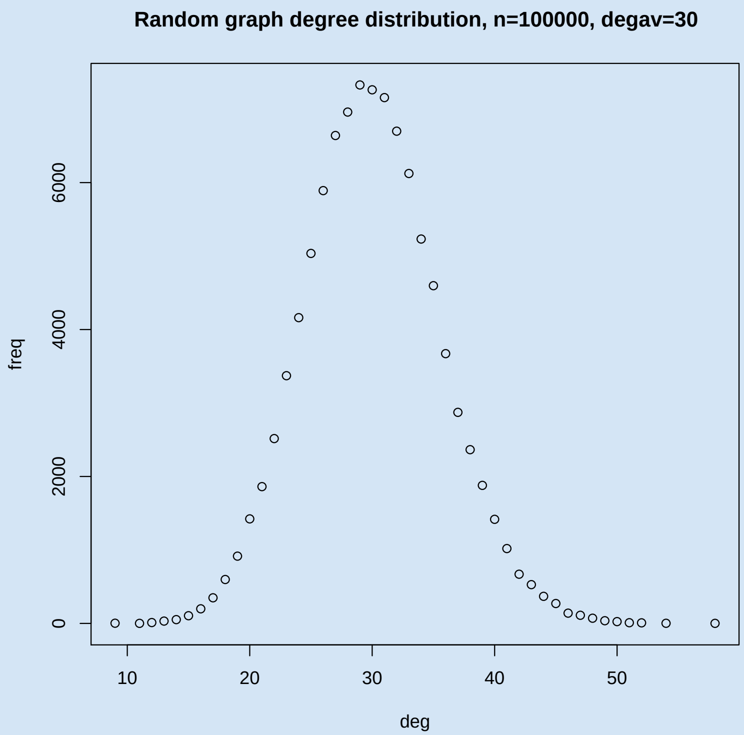
instead of probability p a more intuitive average degree is used

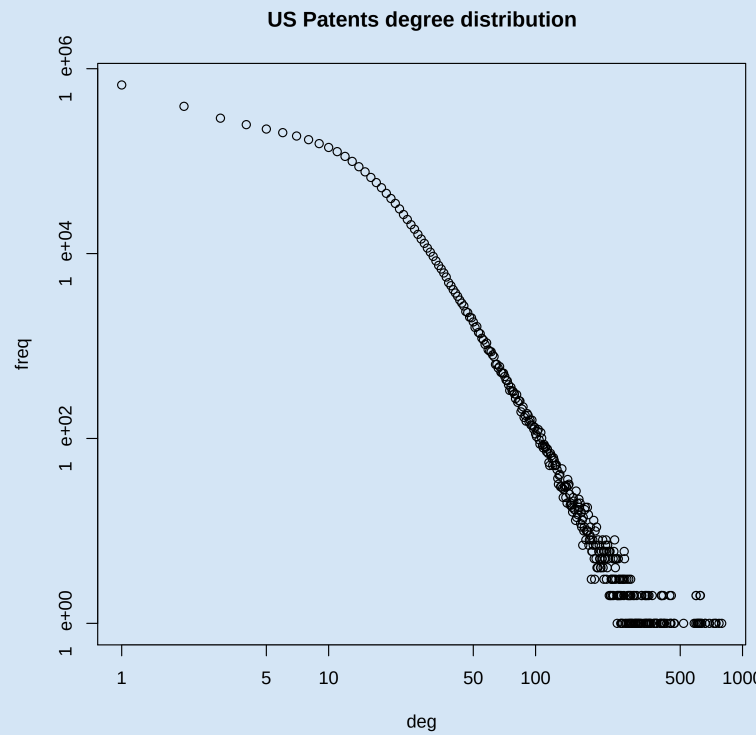
$$\overline{\deg} = \frac{1}{n} \sum_{v \in \mathcal{V}} \deg(v)$$

It holds $p = \frac{m}{m_{max}}$ and, for simple graphs, also $\overline{\deg} = \frac{2m}{n}$.

Random graph in picture has 100 vertices and average degree 3.

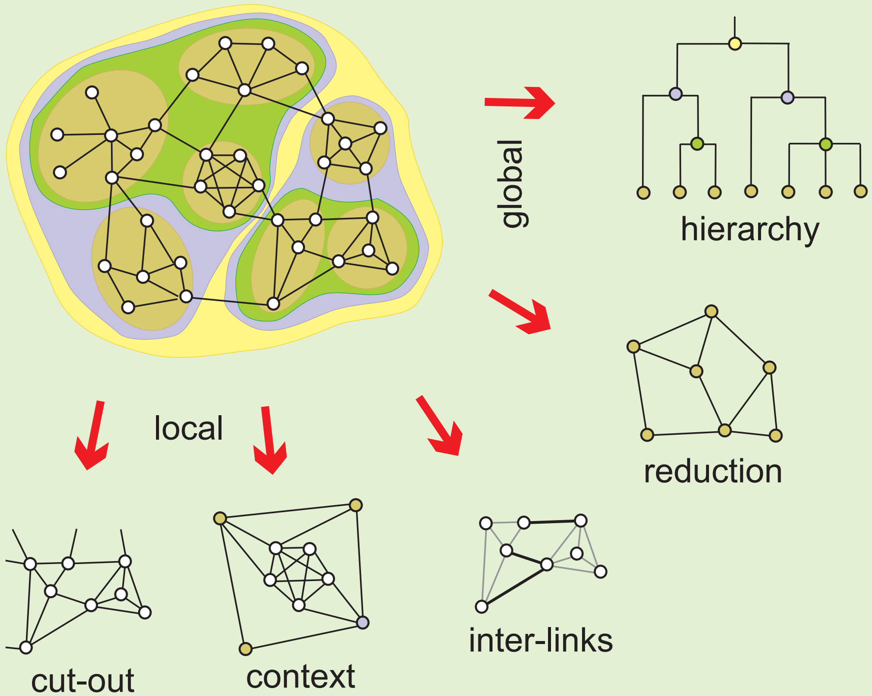
Degree distribution





Real-life networks are usually not random in the Erdős/Rényi sense. The analysis of their distributions gave a new view about their structure – Watts (Small worlds), Barabási (nd/networks, Linked).

Decompositions

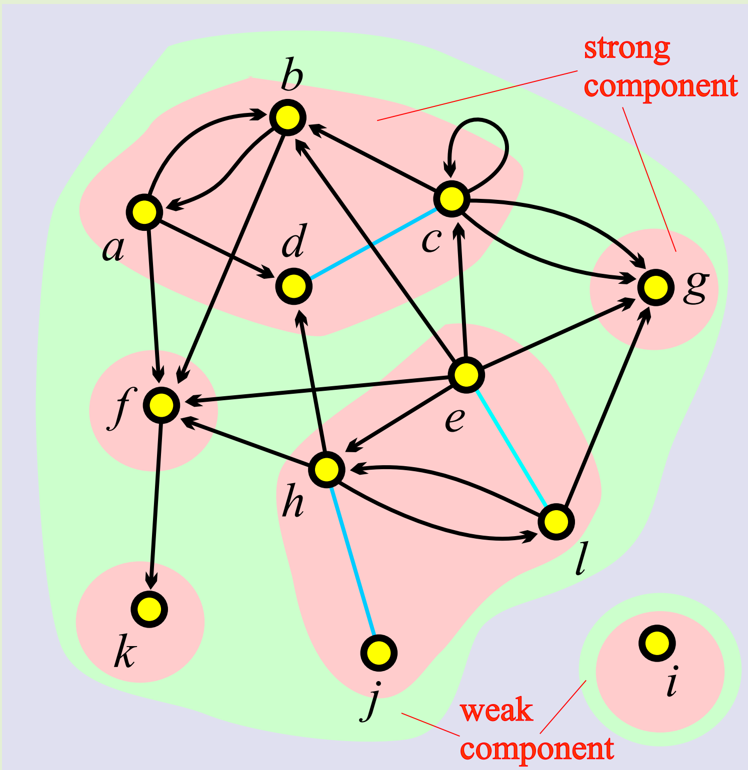


The main goals in the design of **Pajek** are:

- to support abstraction by (recursive) *decomposition* of a large network into several smaller networks that can be treated further using more sophisticated methods;
- to provide the user with some powerful *visualization* tools;
- to implement a selection of efficient *subquadratic* algorithms for analysis of large networks.

With **Pajek** we can: *find* clusters (components, neighbourhoods of 'important' vertices, cores, etc.) in a network, *extract* vertices that belong to the same clusters and *show* them separately, possibly with the parts of the context (detailed local view), *shrink* vertices in clusters and show relations among clusters (global view).

Connectivity



Vertex u is *reachable* from vertex v iff there exists a walk with initial vertex v and terminal vertex u.

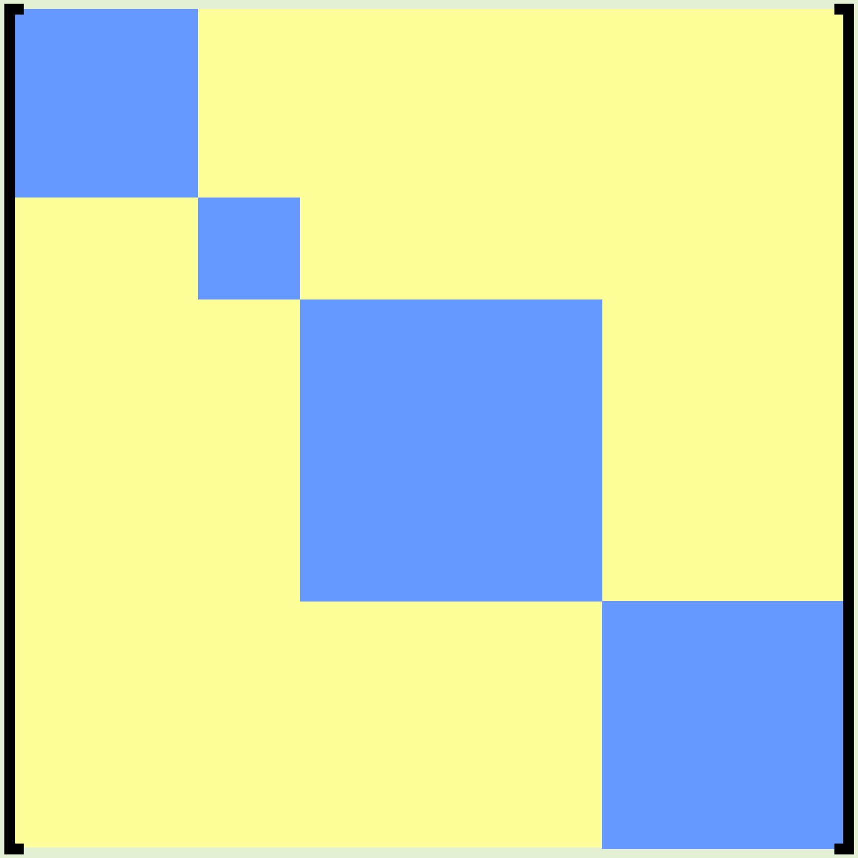
Vertex v is *weakly connected* with vertex u iff there exists a semiwalk with v and u as its end-vertices.

Vertex v is *strongly connected* with vertex u iff they are mutually reachable.

Weak and strong connectivity are equivalence relations.

Equivalence classes induce weak/strong *components*.

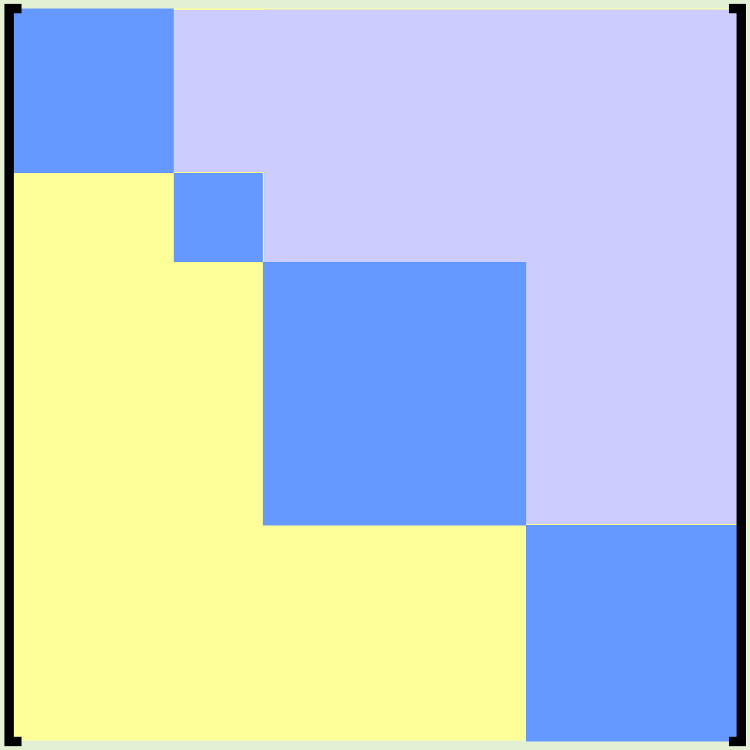
Weak components

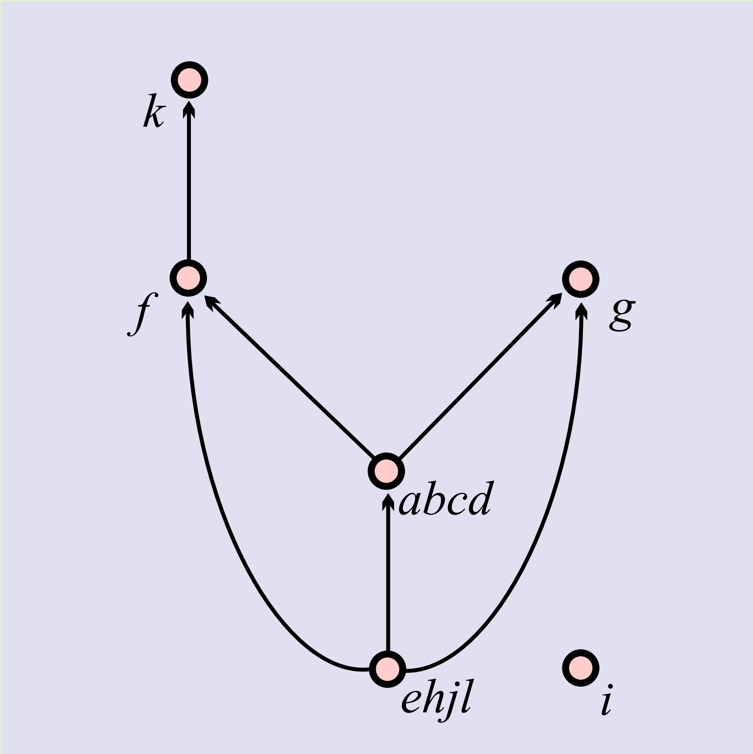


Reordering the vertices of network such that the vertices from the same class of weak partition are put together we get a matrix representation consisting of diagonal blocks – weak components.

Most problems can be solved separately on each component and afterward these solutions combined into final solution.

Reduction (condensation)

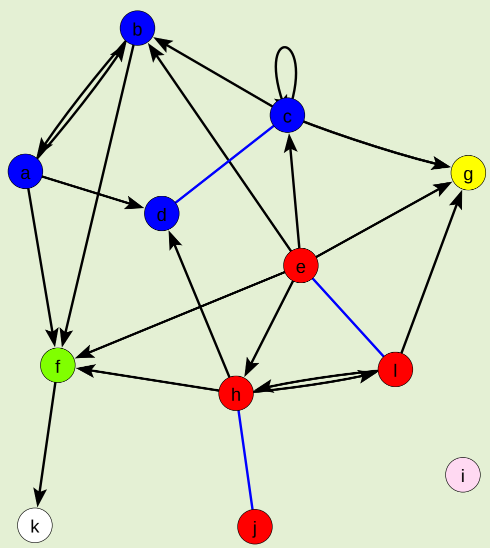


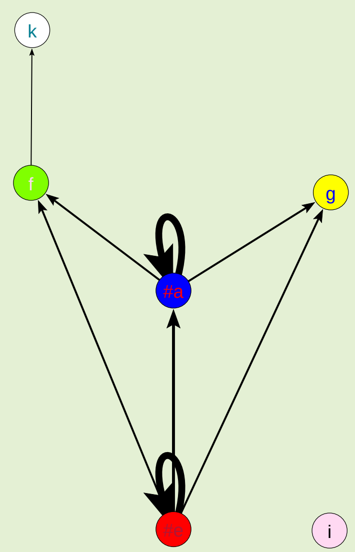


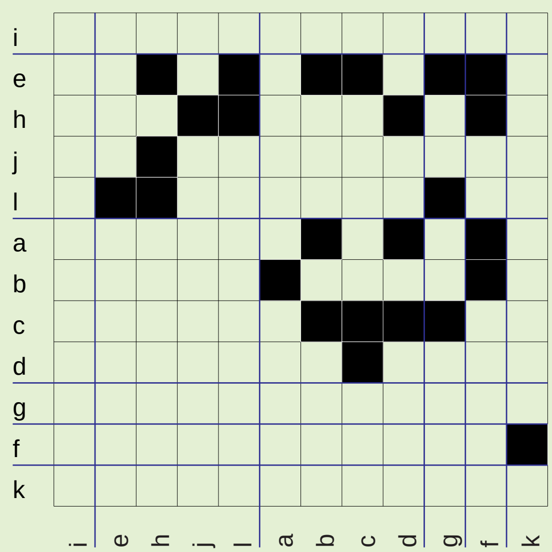
If we shrink every strong component of a given graph into a vertex, delete all loops and identify parallel arcs the obtained *reduced* graph is acyclic. For every acyclic graph an *ordering* / *level* function $i : \mathcal{V} \to \mathbb{N}$ exists s.t. $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$.

Reduction – Example

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







Cuts

The standard approach to find interesting groups inside a network was based on properties/weights – they can be *measured* or *computed* from network structure (for example Kleinberg's hubs and authorities).

The *vertex-cut* of a network $\mathbf{N} = (\mathcal{V}, \mathcal{L}, p)$, $p : \mathcal{V} \to \mathbb{R}$, at selected level t is a subnetwork $\mathbf{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, determined by the set

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

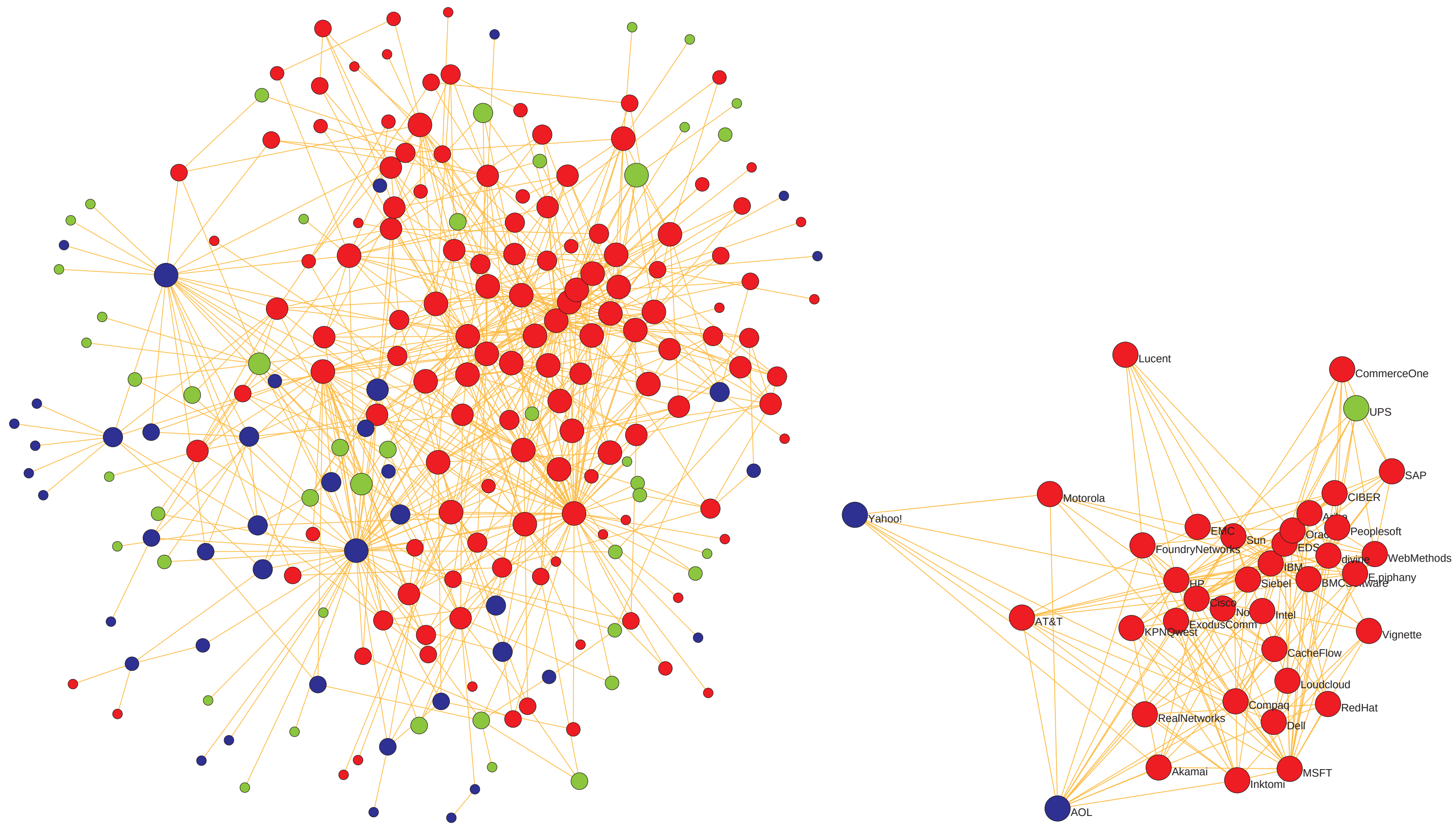
and $\mathcal{L}(\mathcal{V}')$ is the set of lines from $\mathcal{L}$ that have both endpoints in $\mathcal{V}'$.

The *line-cut* of a network $\mathbf{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{V} \to \mathbb{R}$, at selected level t is a subnetwork $\mathbf{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, determined by the set

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

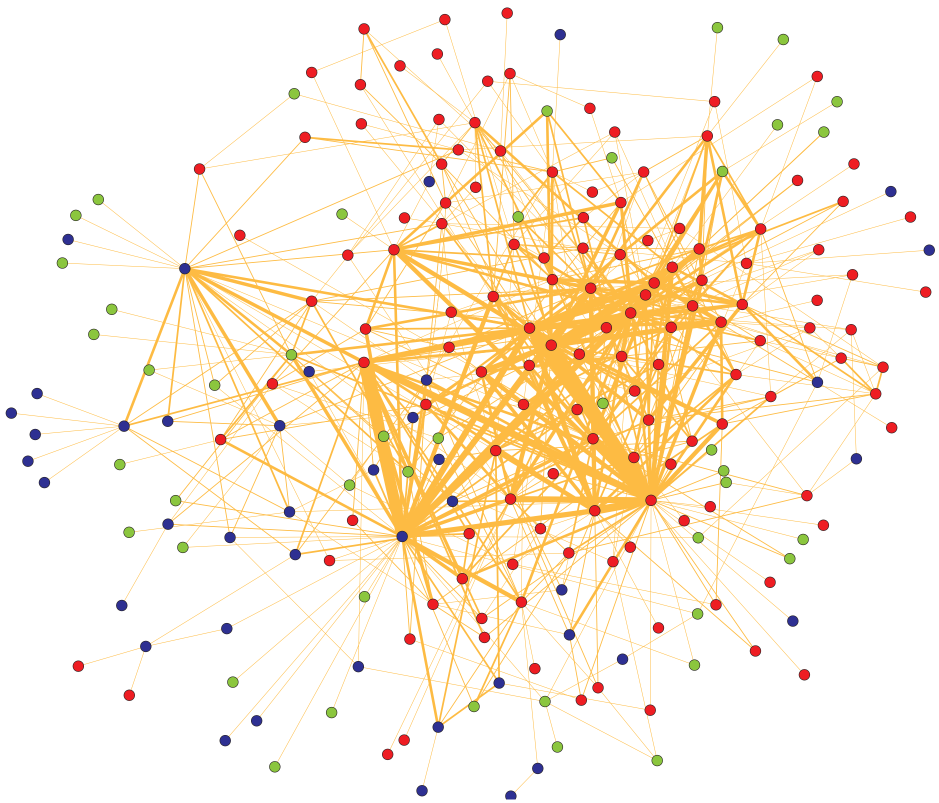
and $\mathcal{V}(\mathcal{L}')$ is the set of all endpoints of the lines from $\mathcal{L}'$.

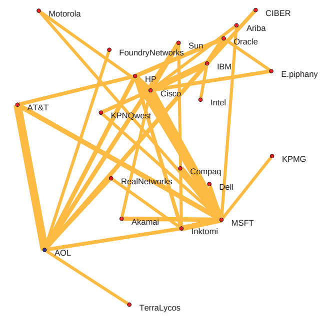
Vertex-cut: Krebs Internet Industries, core=6

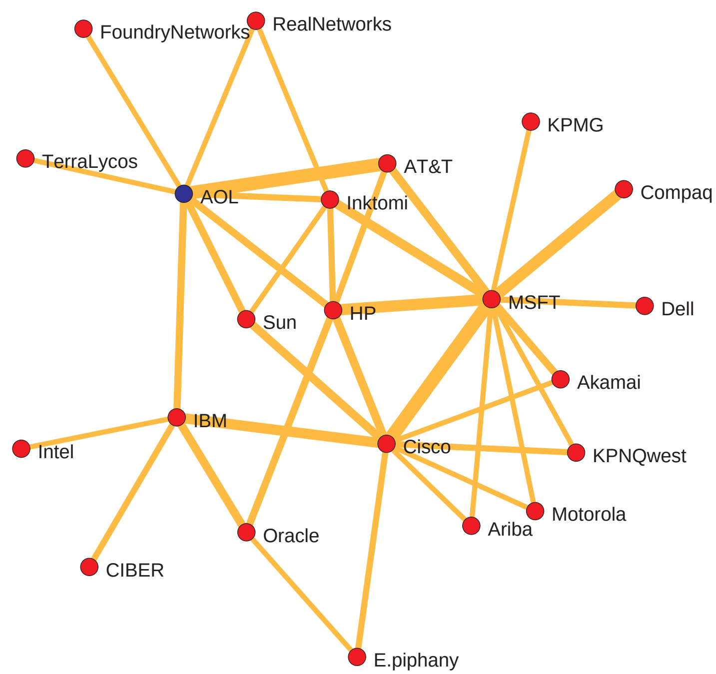


Each vertex represents a company that competes in the Internet industry, 1998 do 2001. $n = 219$, $m = 631$. red – content, blue – infrastructure, green – commerce. Two companies are linked with an edge if they have announced a joint venture, strategic alliance or other partnership.

Line-cut: Krebs Internet Industries, $w_3 \geq 5$







Cuts / Pajek commands

Vertex-cut:

```
File/Pajek Project File/Read  [Krebs.paj]
Net/Partitions/Core/All
Partition/Make Vector
Draw/Draw-Partition-Vector
Layout/Energy/Kamada-Kawai
Operations/Extract from Network/Partition [6]
[select Types ... as First partition]
[select All core ... as Second partition]
Partitions/Extract Second from First [6]
Draw/Draw-Partition
Layout/Energy/Kamada-Kawai
```

Line-cut:

```
[select Krebs ... network]
Net/Count/3-Rings/Undirected
Info/Network/Line Values
Net/Transform/Remove/Lines with Values/lower than [5]
Net/Partitions/Degree/All
Partition/Make Vector
Operations/Extract from Network/Partition [1-*]
[select Types ... as First partition]
[select All Degree ... as Second partition]
Partitions/Extract Second from First [1-*]
Draw/Draw-Partition
Layout/Energy/Kamada-Kawai
```

Simple analysis using cuts

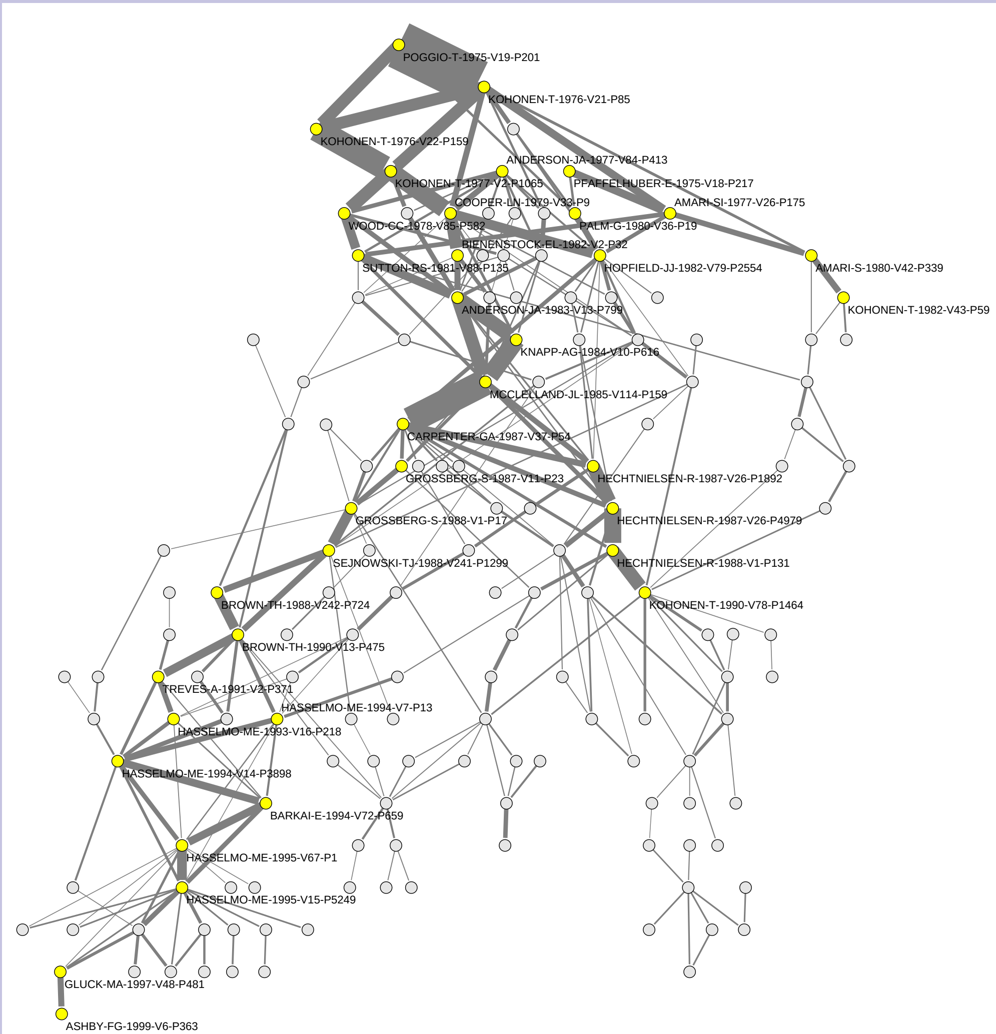
We look at the components of $\mathbf{N}(t)$.

Their number and sizes depend on t. Usually there are many small components. Often we consider only components of size at least k and not exceeding K. The components of size smaller than k are discarded as 'noninteresting'; and the components of size larger than K are cut again at some higher level.

The values of thresholds t, k and K are determined by inspecting the distribution of vertex/arc-values and the distribution of component sizes and considering additional knowledge on the nature of network or goals of analysis.

We developed some new and efficiently computable properties/weights.

Citation weights

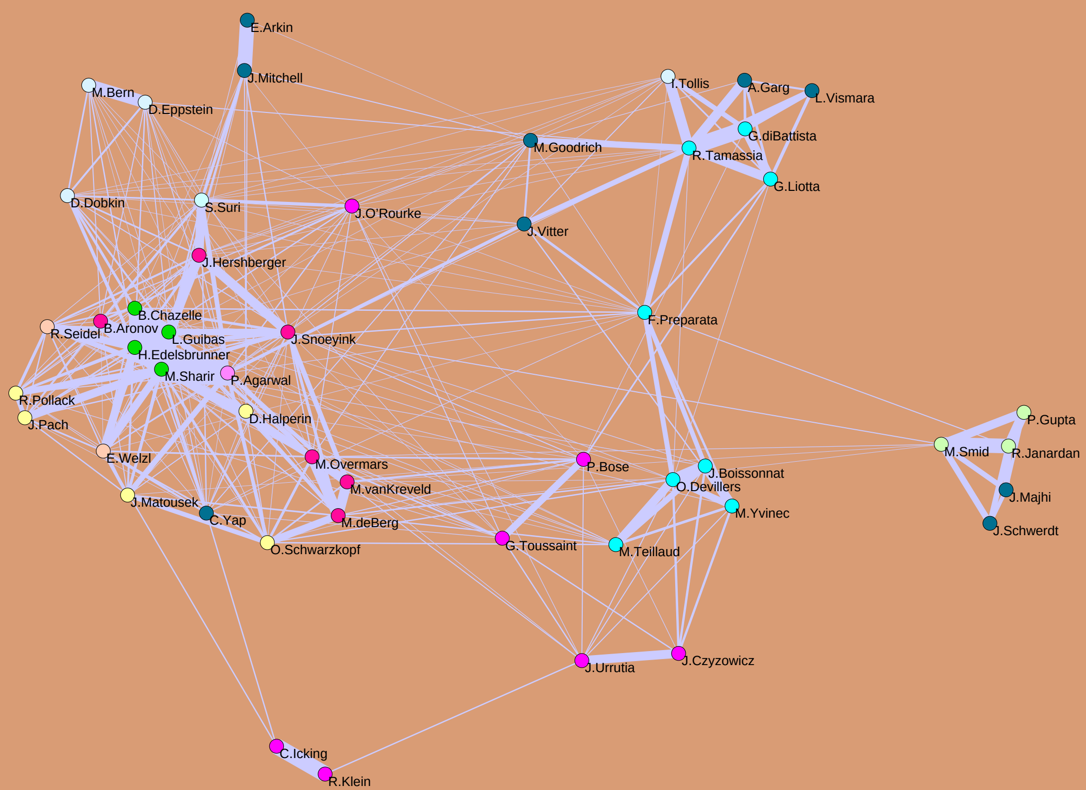


The citation network analysis started in 1964 with the paper of Garfield et al. In 1989 Hummon and Doreian proposed three indices – weights of arcs that are proportional to the number of different source-sink paths passing through the arc. We developed algorithms to efficiently compute these indices.

Main subnetwork (arc cut at level 0.007) of the SOM (selforganizing maps) citation network (4470 vertices, 12731 arcs).

See paper.

Cores and generalized cores



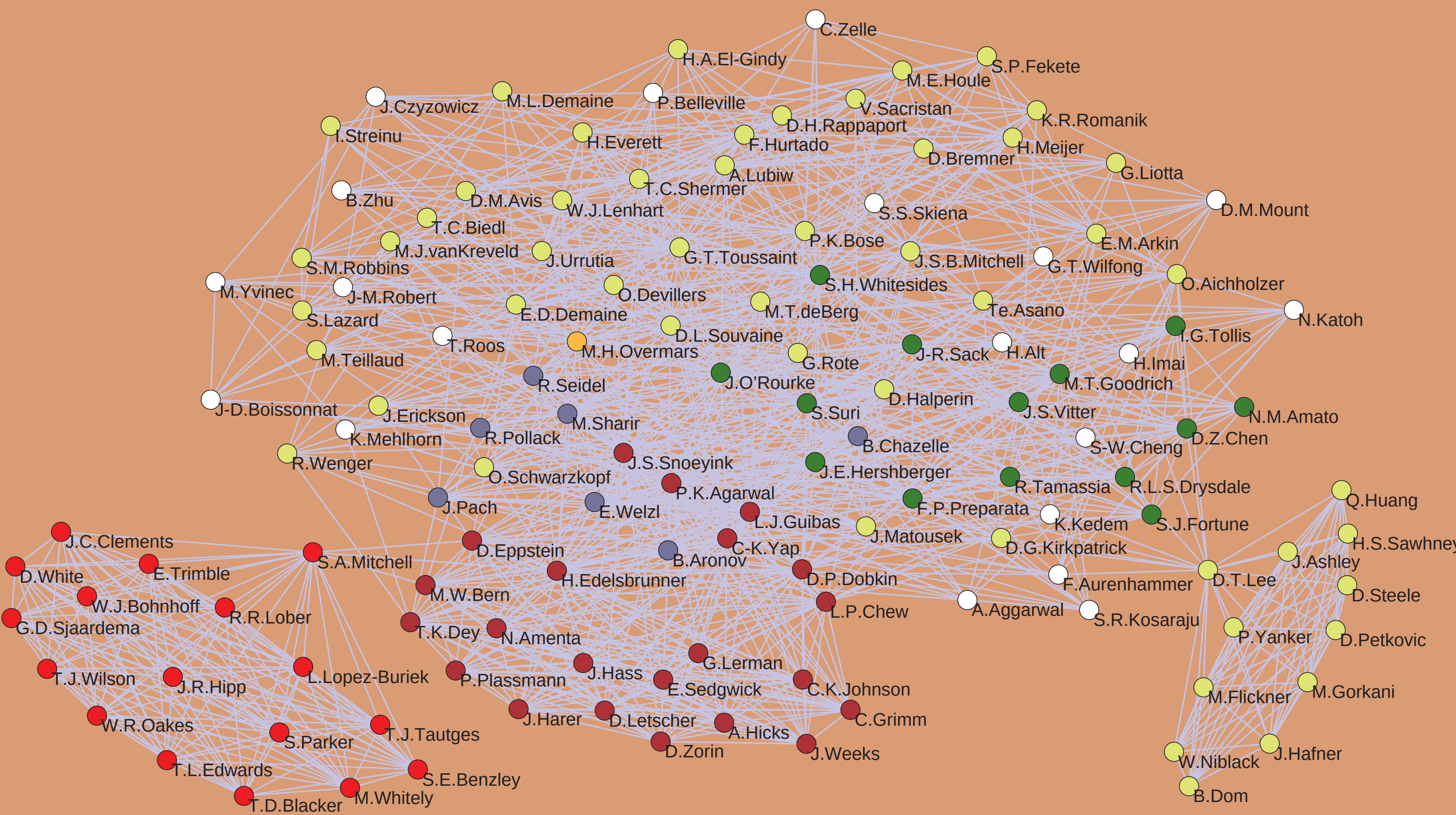
The notion of core was introduced by Seidman in 1983. Vertices belonging to a *k-core* have to be linked to at least k other vertices of the core. A very efficient algorithm exists for determining cores.

The notion of core can be extended to other vertex functions and for several of them the corresponding cores can be efficiently determined.

Figure presents the p_S-core at level 46 of the collaboration network (7343 vertices, 11898 edges, edge weight counts the number of common works) in the field of computational geometry.

See paper.

Cores and generalized cores / Core 10



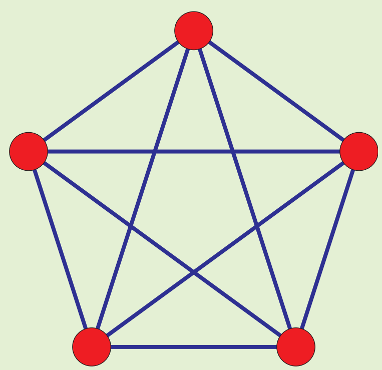
Cores and generalized cores / Pajek commands

```
File/Network/Read  [Geom.net]
Net/Partitions/Core/All
Info/Partition
Operations/Extract from Network/Partition [13-*]
Draw/Draw-Partition
Layout/Energy/Kamada-Kawai
Options/Values of lines/Similarities
Layout/Energy/Kamada-Kawai
Operations/Extract from Network/Partition [21]
Draw
Layout/Energy/Kamada-Kawai
Options/Values of lines/Forget
Layout/Energy/Kamada-Kawai
[select Geom.net]
Net/Vector/PCore/Sum/All
Info/Vector
Vector/Make Partition/by Intervals/Selected Thresholds [45]
Info/Partition
Operations/Extract from Network/Partition [2]
Draw
Options/Values of lines/Similarities
Layout/Energy/Fruchterman-Reingold
```

k-rings

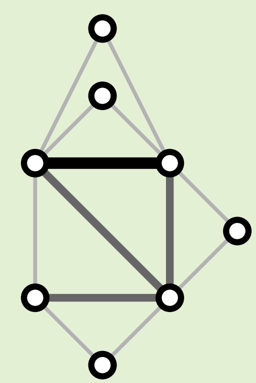
A *k-ring* is a simple closed chain of length k. Using k-rings we can define a weight of edges as

$w_k(e) = \#$ of different k-rings containing the edge $e \in E$



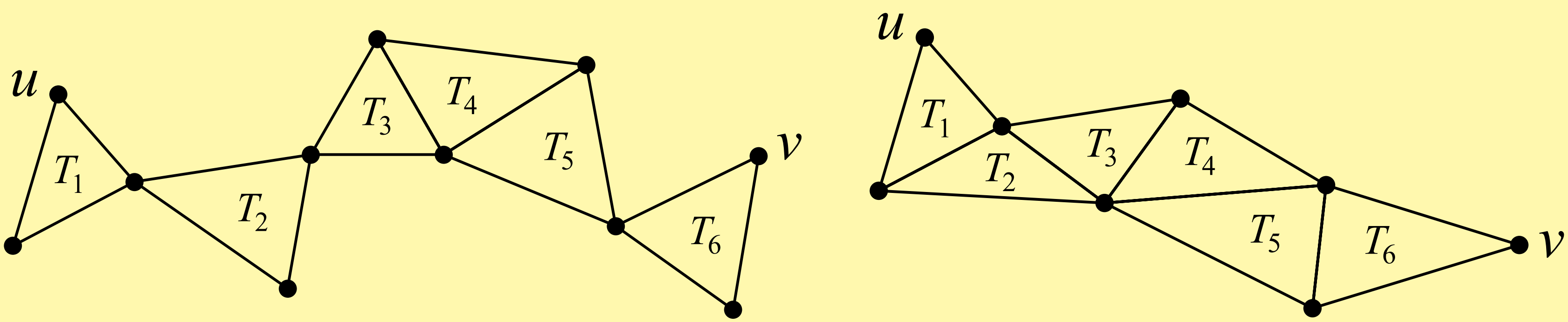
Since for a complete graph K_r, $r \geq k \geq 3$ we have $w_k(K_r) = (r-2)!/(r-k)!$, the edges belonging to cliques have large weights. Therefore these weights can be used to identify the dense parts of a network. For example: all r-cliques of a network belong to $r-2$-edge cut for the weight w_3.

We can assign to a given graph a *triangular network* in which every line of the original graph gets as its weight the number of triangles that contain it. The triangular weights provide us, combined with islands, with a very efficient way to identify dense parts of a graph.



Triangular connectivity

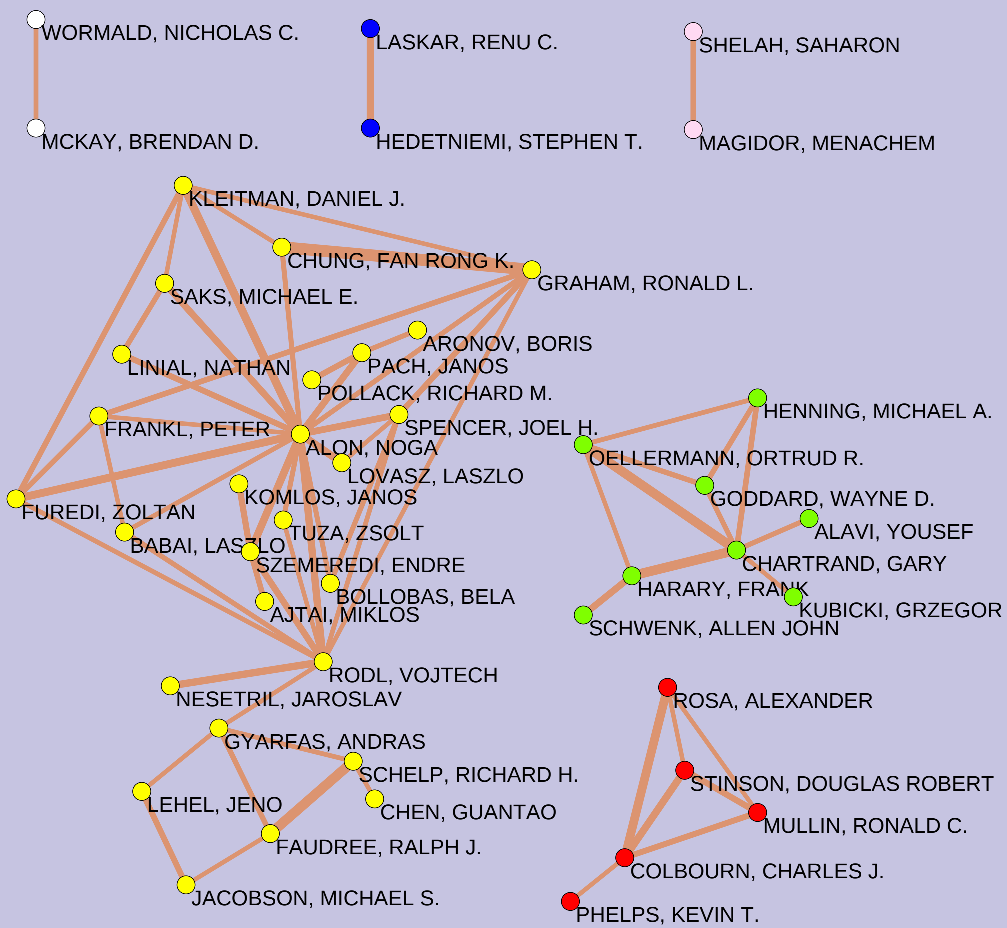
Related to triangular network is the notion of *triangular connectivity*



that can be used to operationalize the notion of strong ties.

These notions can be generalized to short cycle connectivity (see paper).

Edge-cut at level 16 of triangular network of Erdős collaboration graph



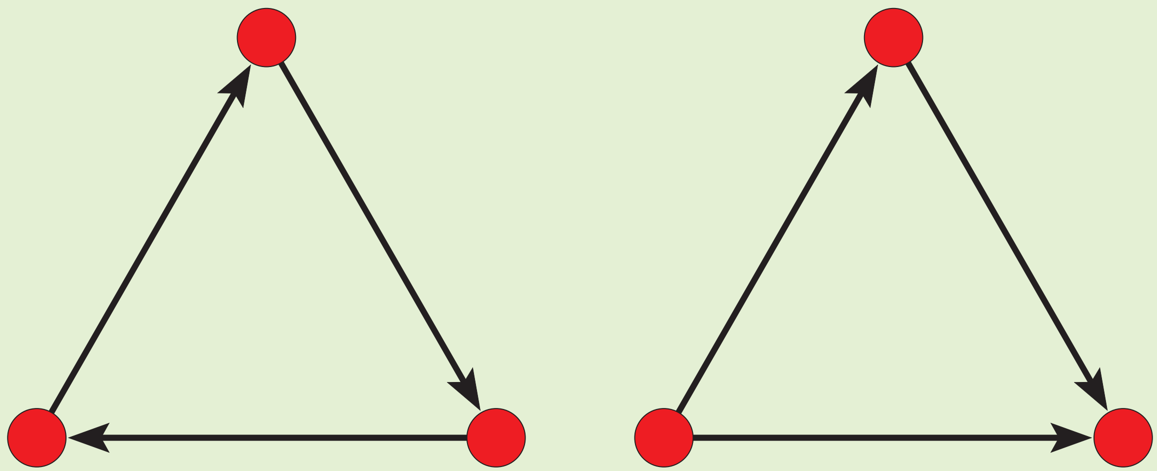
without Erdős,

$n = 6926$,

$m = 11343$

Directed 3-rings

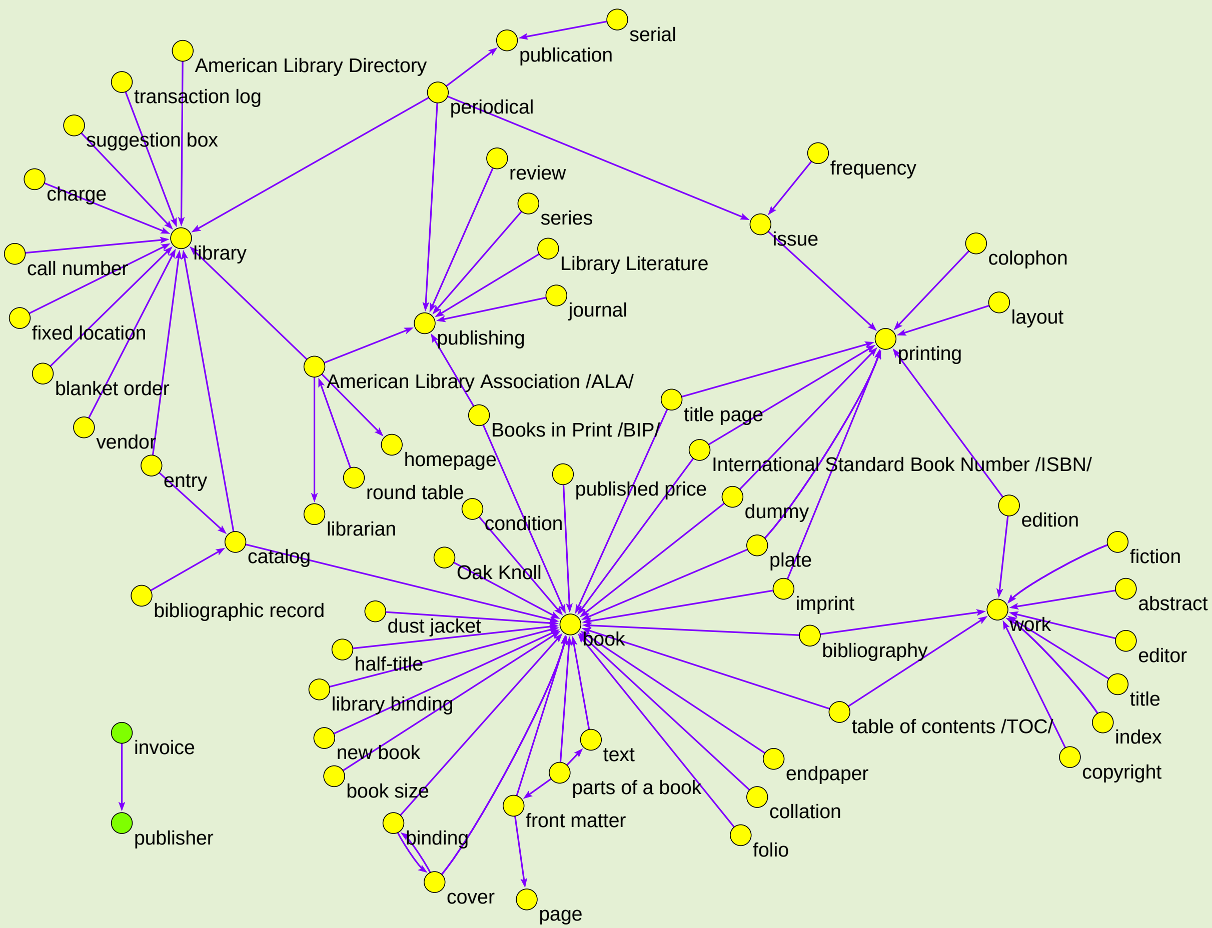
In directed networks there are two types of 3-rings:



cyclic transitive

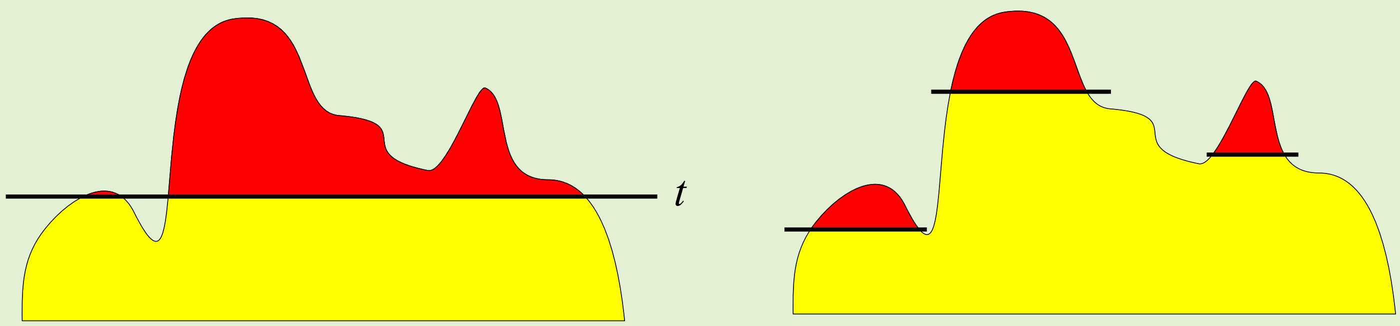
The 3-rings weights were implemented in **`Pajek`** in May 2002.

Edge-cut at level 11 of transitive network of ODLIS dictionary graph



Islands

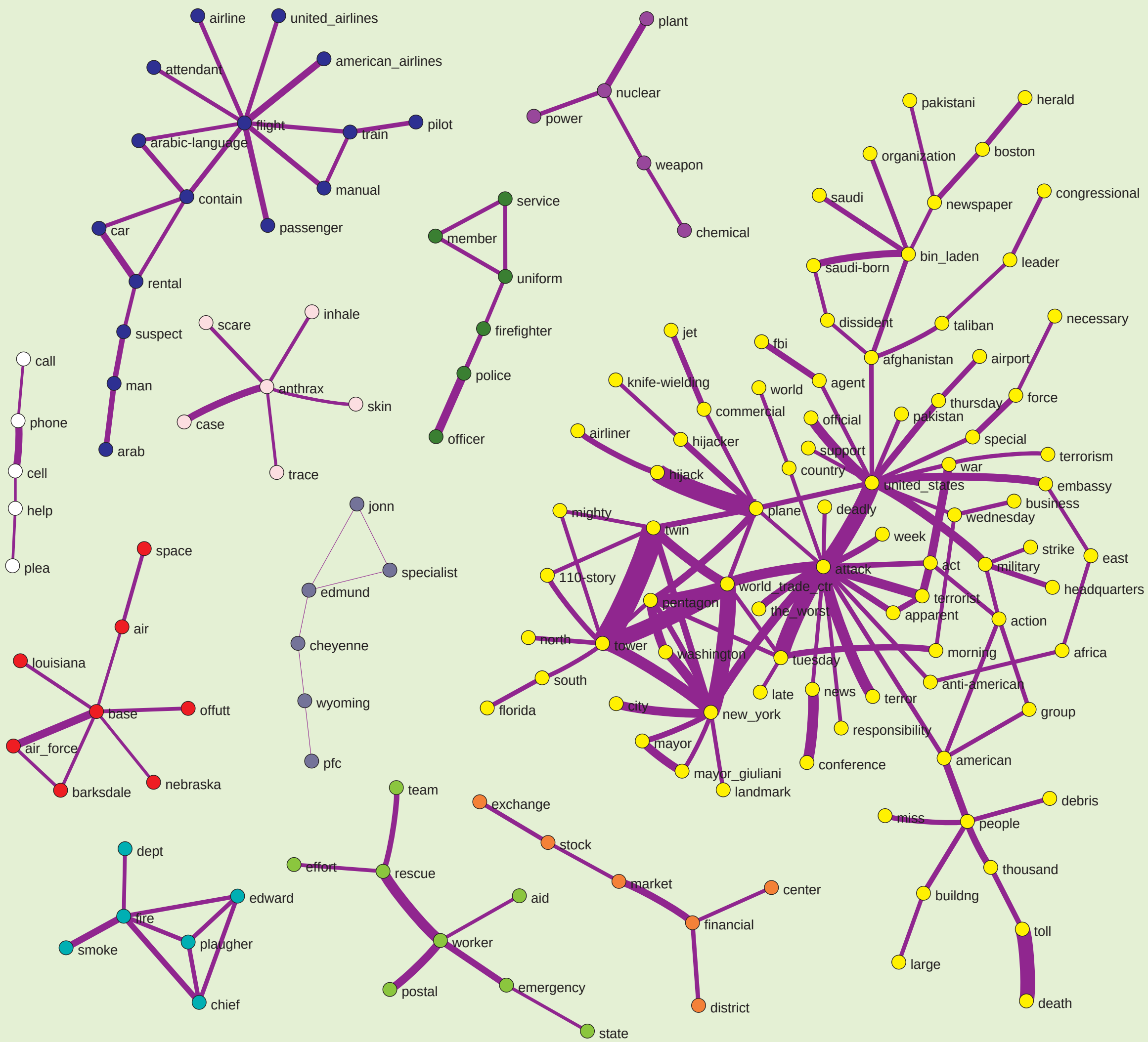
If we represent a given or computed value of vertices / lines as a height of vertices / lines and we immerse the network into a water up to selected level we get *islands*. Varying the level we get different islands. Islands are very general and efficient approach to determine the 'important' subnetworks in a given network.



We developed very efficient algorithms to determine the islands hierarchy and to list all the islands of selected sizes.

See details.

Islands - Reuters terror news



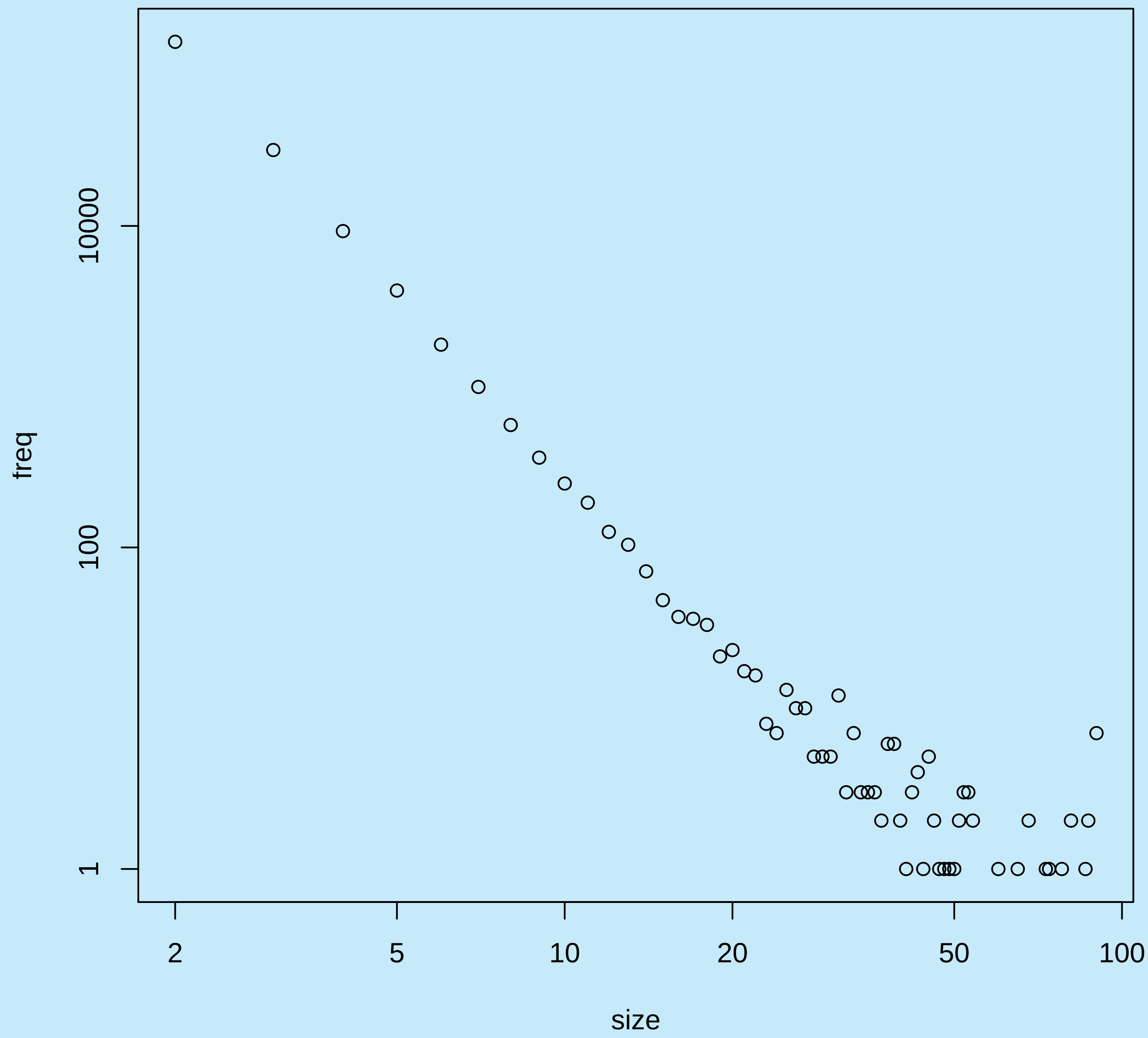
Using CRA S. Corman and K. Dooley produced the *Reuters terror news network* that is based on all stories released during 66 consecutive days by the news agency Reuters concerning the September 11 attack on the US. The vertices of a network are words (terms); there is an edge between two words iff they appear in the same text unit. The weight of an edge is its frequency. It has $n = 13332$ vertices and $m = 243447$ edges.

Islands – US patents

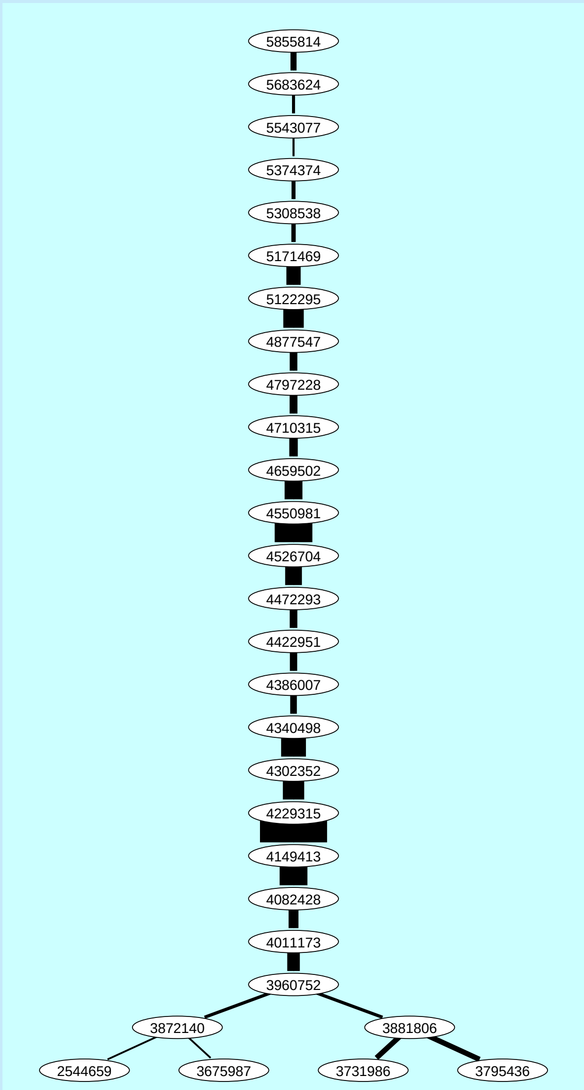
As an example, let us look at Nber network of US Patents. It has 3774768 vertices and 16522438 arcs (1 loop). We computed SPC weights in it and determined all (2,90)-islands. The reduced network has 470137 vertices, 307472 arcs and for different k: C_2 =187610, C_5 =8859,C_{30} =101, C_{50} =30 islands. Rolex

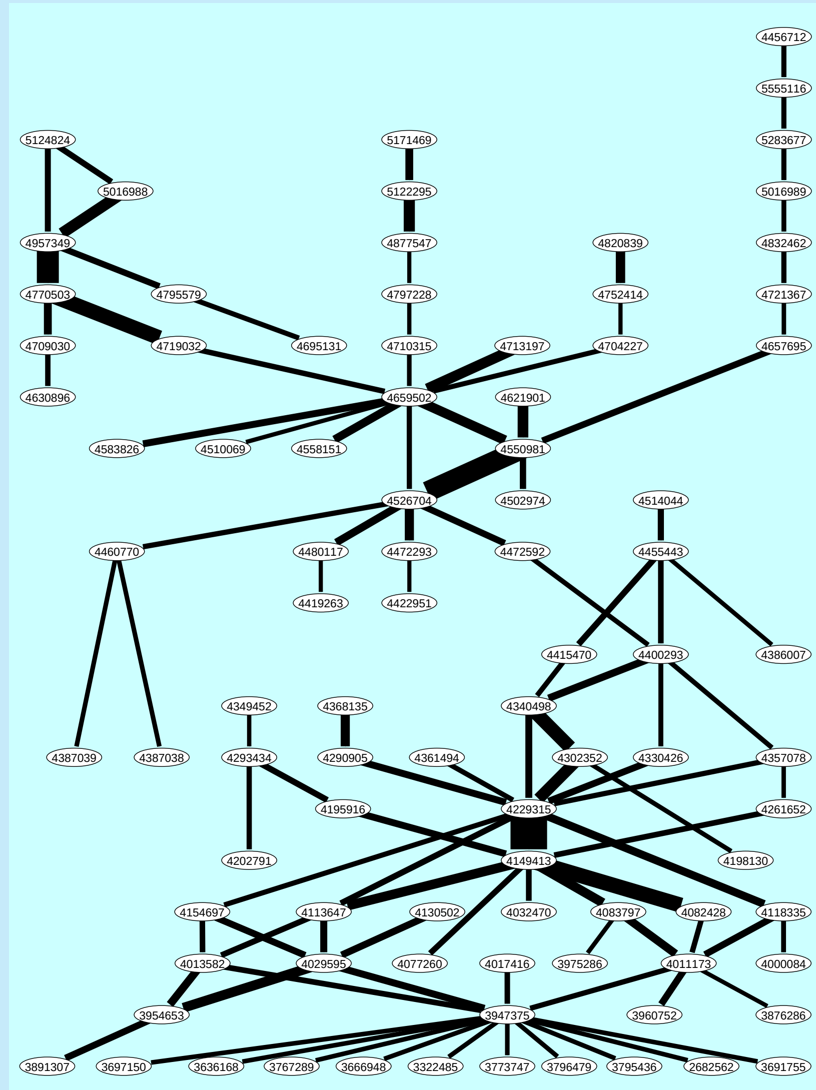
```
 [1]    0 139793  29670 9288 3966 1827 997 578 362 250
[11] 190    125    104   71   47   37  36  33  21  23
[21]  17     16      8    7   13   10  10   5   5   5
[31]  12      3      7    3    3    3   2   6   6   2
[41]   1      3      4    1    5    2   1   1   1   1
[51]   2      3      3   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

Island size distribution



Main path and main island of Patents





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

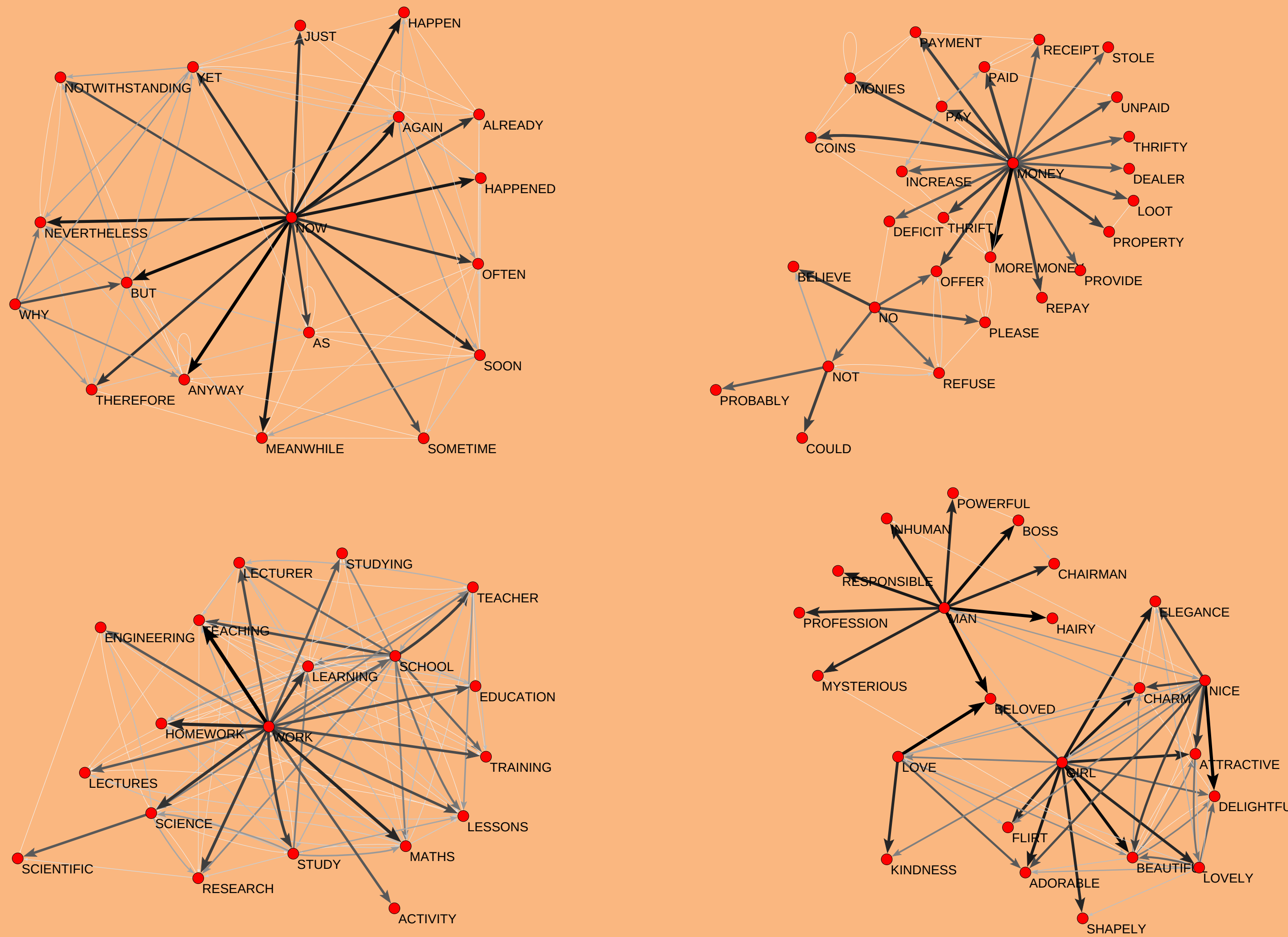
patent	date	author(s) and title
4083797	Apr 11, 1978	Oh. Nematic liquid crystal compositions
4113647	Sep 12, 1978	Coates, et al. Liquid crystalline materials
4118335	Oct 3, 1978	Krause, et al. Liquid crystalline materials of reduced viscosity
4130502	Dec 19, 1978	Eidenschink, et al. Liquid crystalline 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-sub stituted 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

Islands – The Edinburgh Associative Thesaurus

$n = 23219$, $m = 325624$, transitivity weight



Islands / Pajek commands

```
File/Network/Read  [eatRS.net]
Net/Partitions/Islands/Generate Network with Islands [On]
Net/Partitions/Islands/Line Weights Simple [2 50]
Partition/Canonical Partition - Decreasing Frequencies
Info/Partition
Operations/Extract from Network/Partition [1-38]
Draw/Draw-Partition-Vector
Layout/Energy/Kamada-Kawai/Free
[manually distribute components over the available space]
Options/Transform/Fit area
```

The procedure for ’triangular islands’ is similar

```
File/Network/Read  [eatRS.net]
Net/Count/3-Rings/Directed/Transitive
Net/Partitions/Islands/Generate Network with Islands [On]
Net/Partitions/Islands/Line Weights Simple [2 50]
...
```

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

12th Annual Graph Drawing Contest, 2005. The IMDB network is bipartite (2-mode) and has $1324748 = 428440 + 896308$ vertices and 3792390 arcs.

Bipartite cores

The subset of vertices $C \subseteq V$ is a *(p, q)-core* in a bipartite (2-mode) network $N = (V_1, V_2; L)$, $V = V_1 \cup V_2$ iff

a. in the induced subnetwork $K = (C_1, C_2; L(C))$, $C_1 = C \cap V_1$, $C_2 = C \cap V_2$ it holds $\forall v \in C_1 : \deg_K(v) \geq p$ and $\forall v \in C_2 : \deg_K(v) \geq q$;

b. C is the maximal subset of V satisfying condition **a**.

Properties of bipartite cores:

- $C(0, 0) = V$
- $K(p, q)$ is not always connected
- $(p_1 \leq p_2) \wedge (q_1 \leq q_2) \Rightarrow C(p_1, q_1) \subseteq C(p_2, q_2)$
- $\mathcal{C} = \{C(p, q) : p, q \in \mathbb{N}\}$. If all nonempty elements of $\mathcal{C}$ are different it is a lattice.

Algorithm for bipartite cores

To determine a (p, q)-core the procedure similar to the ordinary core procedure can be used:

repeat
  remove from the first set all vertices of degree less than p,
    and from the second set all vertices of degree less than q
until no vertex was deleted

It can be implemented to run in $O(m)$ time.

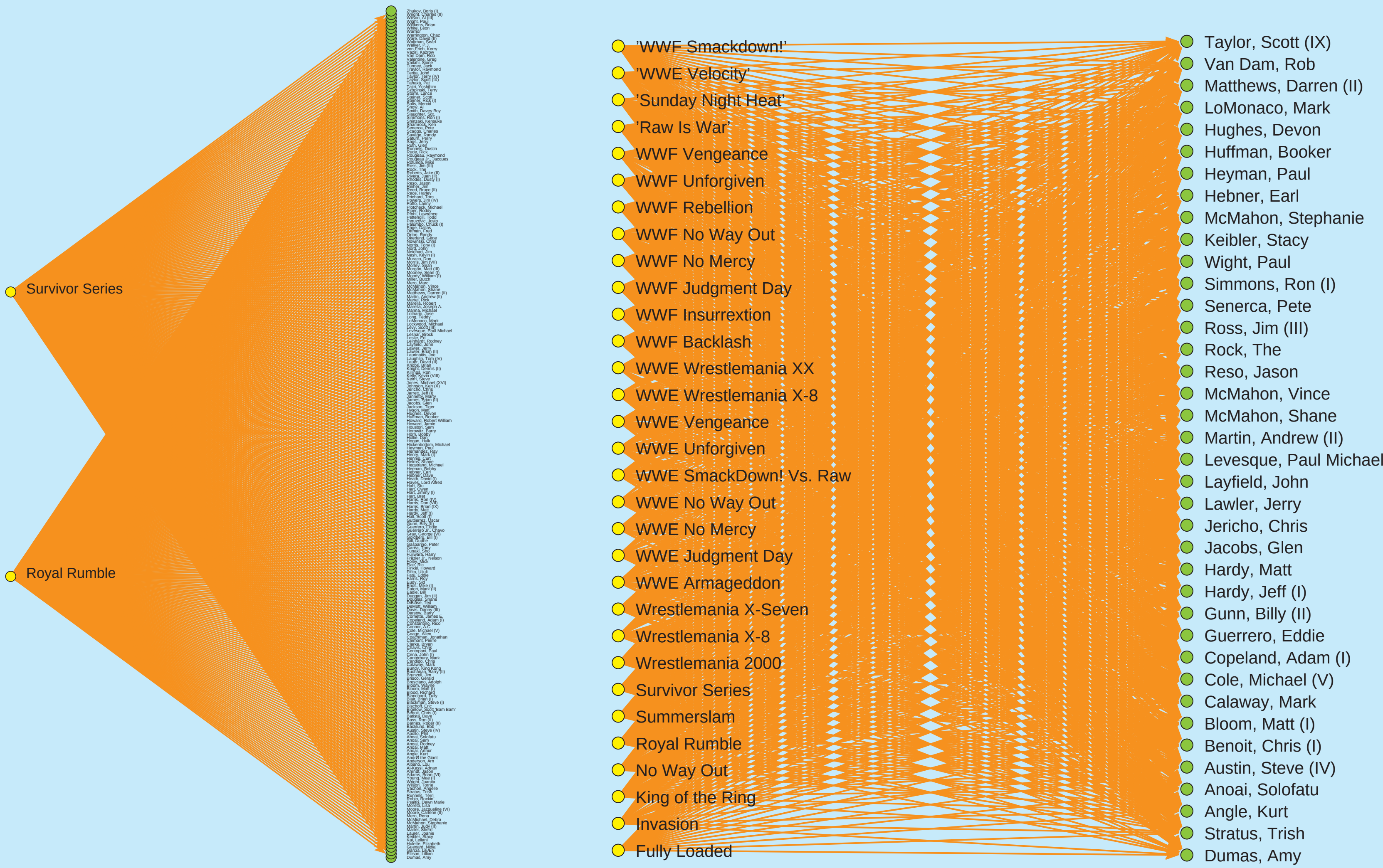
Interesting (p, q)-cores? Table of cores' characteristics $n_1 = |C_1(p, q)|$, $n_2 = |C_2(p, q)|$ and k – number of components in $K(p, q)$:

- $n_1 + n_2 \leq$ selected threshold
- big jumps from $C(p - 1, q)$ and $C(p, q - 1)$ to $C(p, q)$.

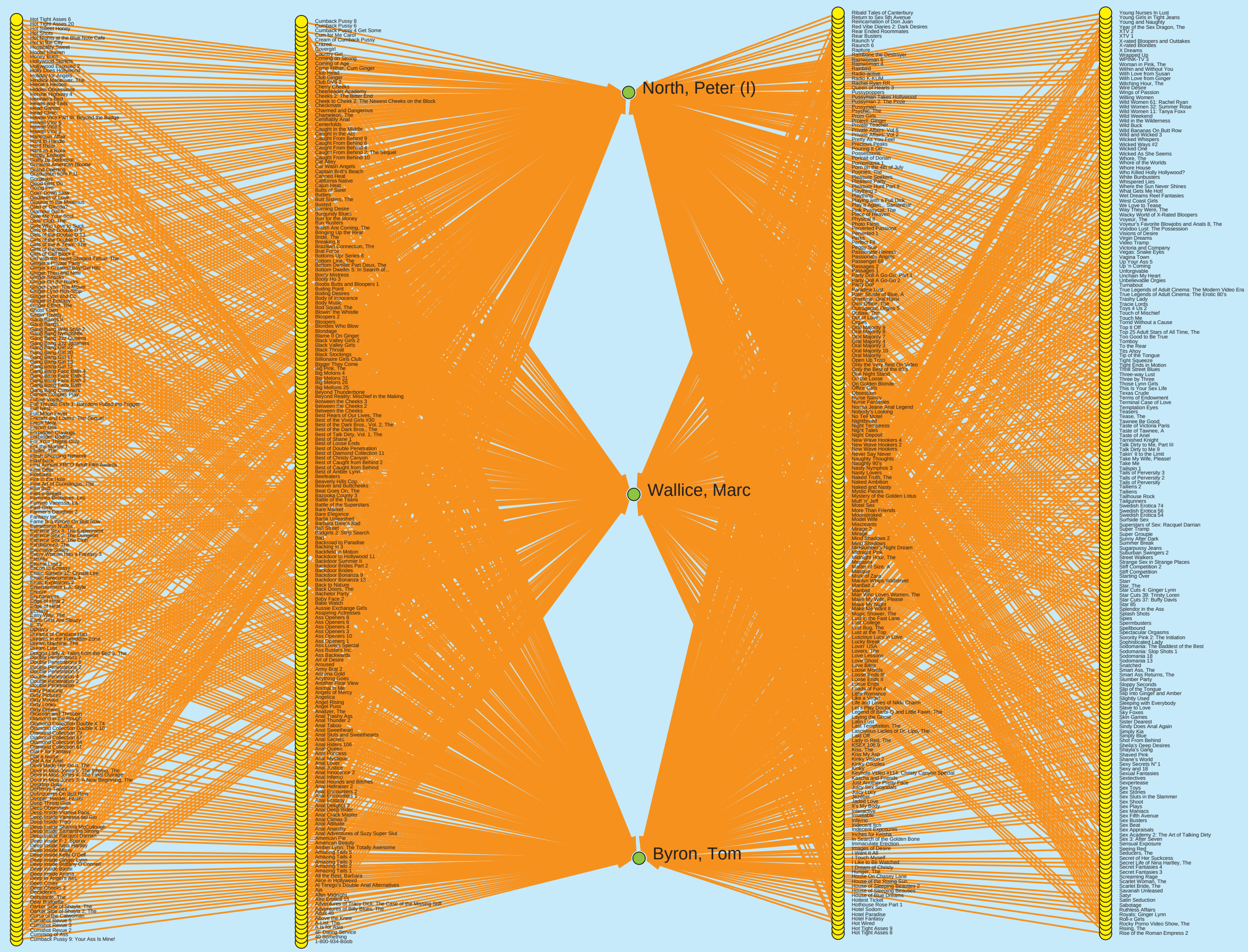
Table $(p, q : n_1, n_2)$ for Internet Movie Database

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)-core and (27,22)-core



(2,516)-Hard core



IMDB cores / Pajek commands

See How to deal with very large networks?

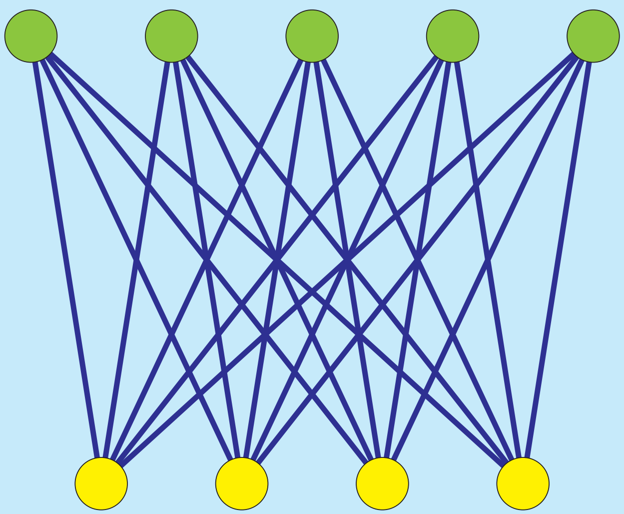
```
Options/Read-Write/Read-Save vertices labels [Off]
Read/Network [IMDB.net]  1:40
Info/Memory
Net/Partitions/Core/2-Mode Review
Net/Partitions/Core/2-Mode [27 22]
Info/Partition
Operations/Extract from Network/Partition [Yes 1]
Net/Partitions/2-Mode
Net/Transform/Add/Vertices Labels from File [IMDB.nam]
Draw/Draw-Partition
Layers/in y direction
Options/Transform/Rotate 2D [90]
```

Different result (because of multiple lines)

```
Net/Components/Weak [2]
Draw/Draw-Partition
Net/Transform/Remove/Multiple lines/Single line
Net/Partitions/Core/2-Mode [27 22]
Operations/Extract from Network/Partition [Yes 1]
Draw/Draw-Partition
```

4-rings and analysis of 2-mode networks

In bipartite (2-mode) network there are no 3-rings. The densest substructures are complete bipartite subgraphs $K_{p,q}$. They contain many 4-rings.

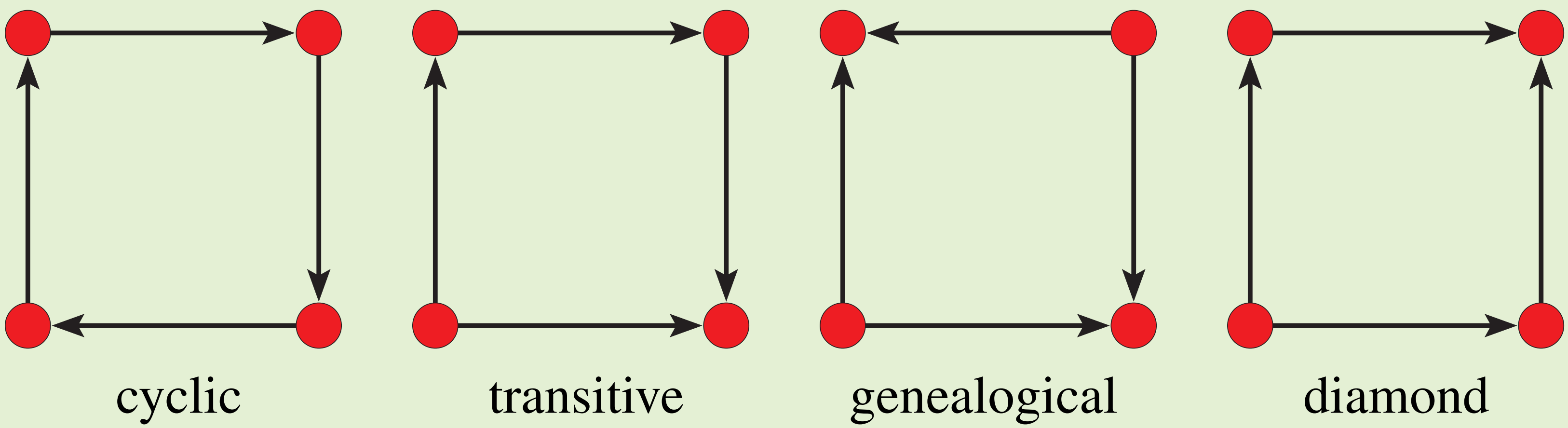


$$w_4(K_{p,q}) = \binom{p}{2}\binom{q}{2}$$

The 4-rings weights were implemented in **Pajek** only recently, in August 2005.

Directed 4-rings

There are 4 types of directed 4-rings:



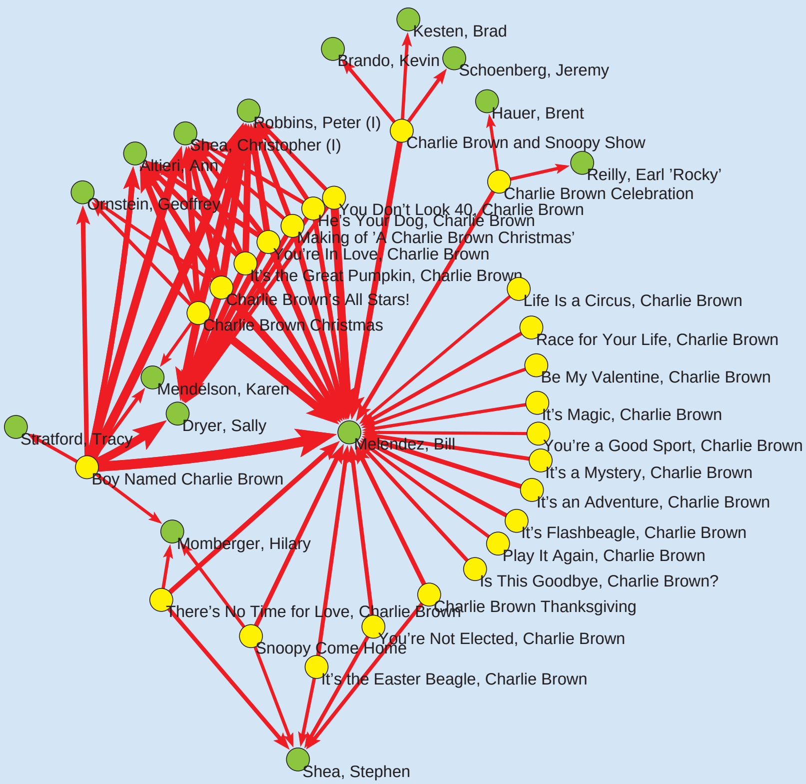
In the case of transitive rings **Pajek** provides a special weight counting on how many transitive rings the arc is a *shortcut*.

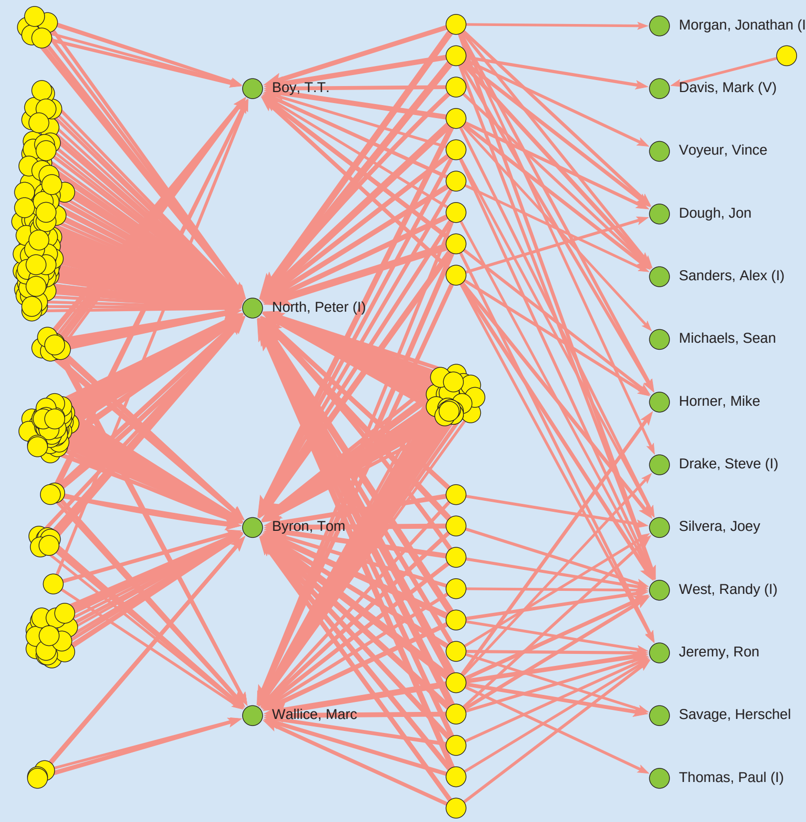
Simple line islands in IMDB for w_4

We obtained 12465 simple line islands on 56086 vertices. Here is their size distribution.

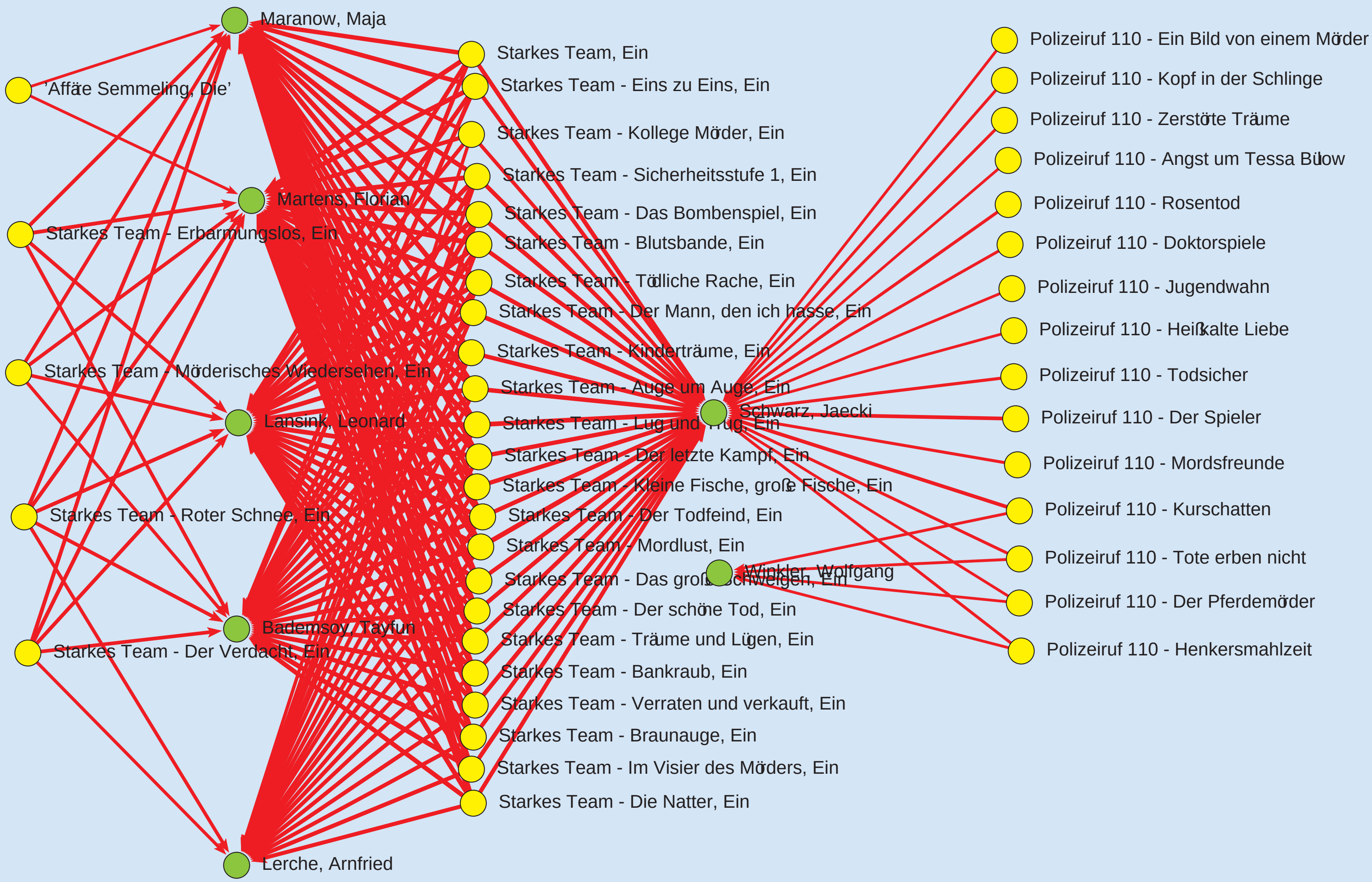
Size	Freq	Size	Freq	Size	Freq	Size	Freq
2	5512	20	19	38	4	59	2
3	1978	21	18	39	3	61	1
4	1639	22	15	40	2	64	1
5	968	23	9	42	2	67	1
6	666	24	13	43	3	70	1
7	394	25	12	45	3	73	1
8	257	26	6	46	4	76	1
9	209	27	6	47	5	82	1
10	148	28	5	48	1	86	1
11	118	29	6	49	2	106	1
12	87	30	3	50	2	122	1
13	55	31	6	51	1	135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

Example: Islands for w_4 / Charlie Brown and Adult



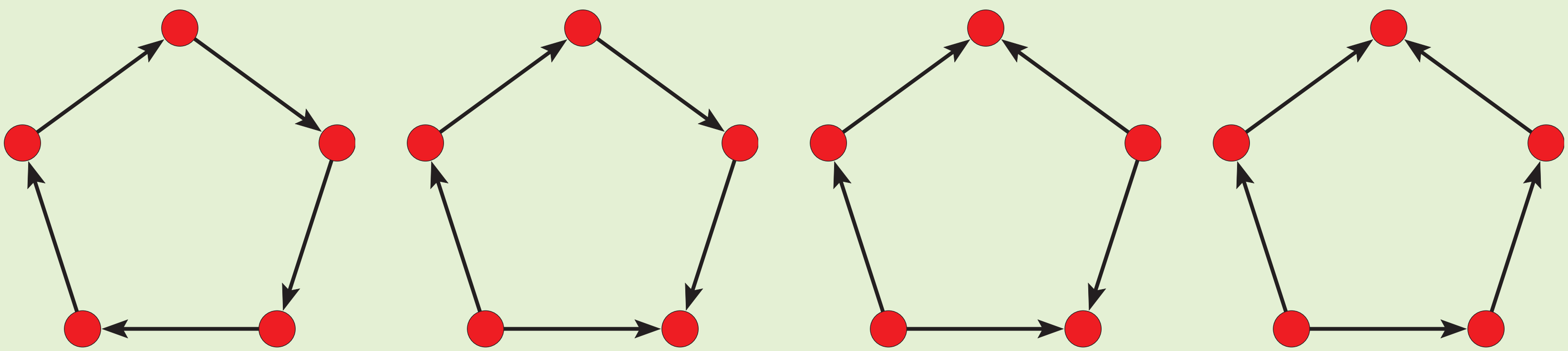


Example: Island for w_4 / Polizeiruf 110 and Starkes Team



5-rings

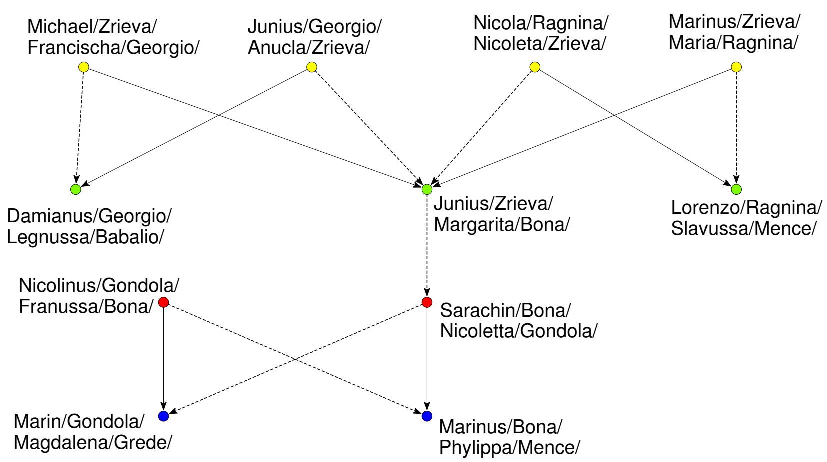
In the future we intend to implement in **Pajek** also weights w_5. Again there are only 4 types of directed 5-rings.



cyclic transitive ???? ????

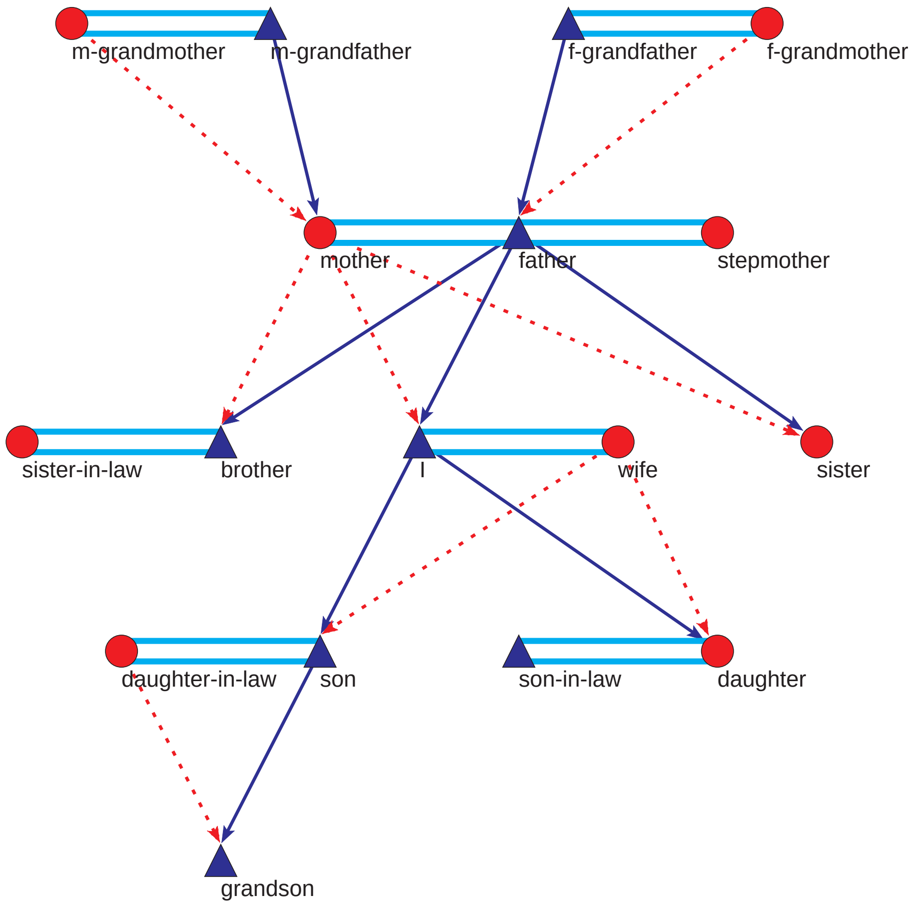
Pattern searching

If a selected *pattern* determined by a given graph does not occur frequently in a sparse network the straightforward backtracking algorithm applied for pattern searching finds all appearences of the pattern very fast even in the case of very large networks. Pattern searching was successfully applied to searching for patterns of atoms in molecula (carbon rings) and searching for relinking marriages in genealogies.



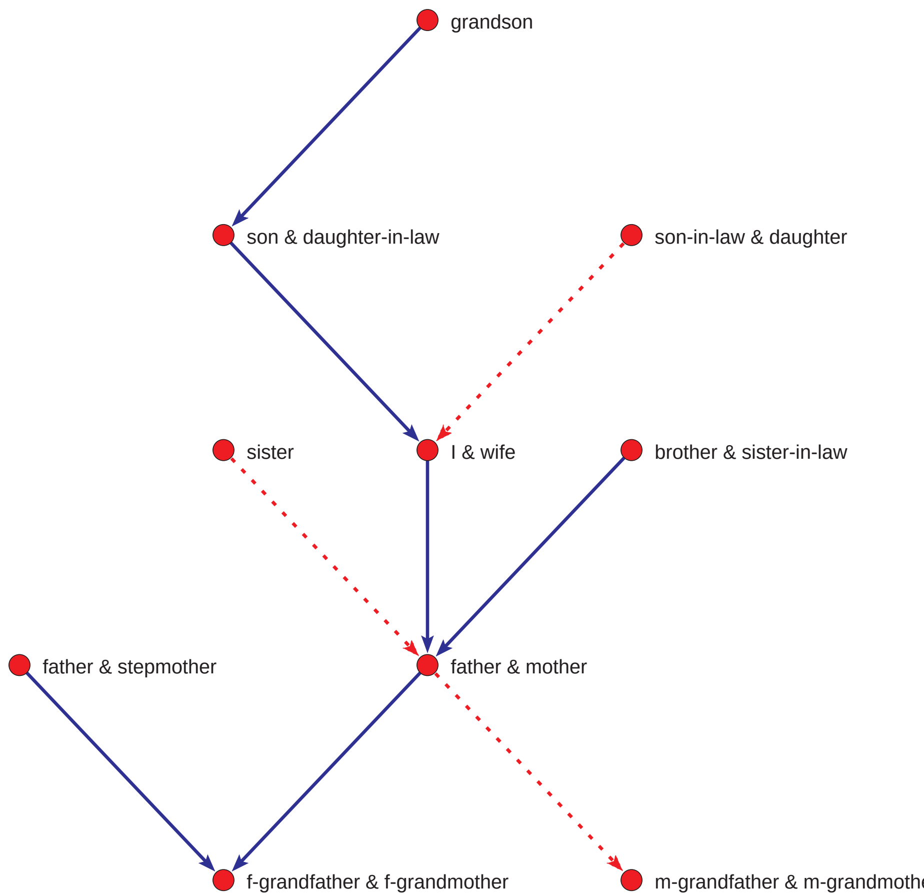
Three connected relinking marriages in the genealogy (represented as a p-graph) of ragusan noble families. A solid arc indicates the _ *is a son of* _ relation, and a dotted arc indicates the _ *is a daughter of* _ relation. In all three patterns a brother and a sister from one family found their partners in the same other family.

Ore-graph



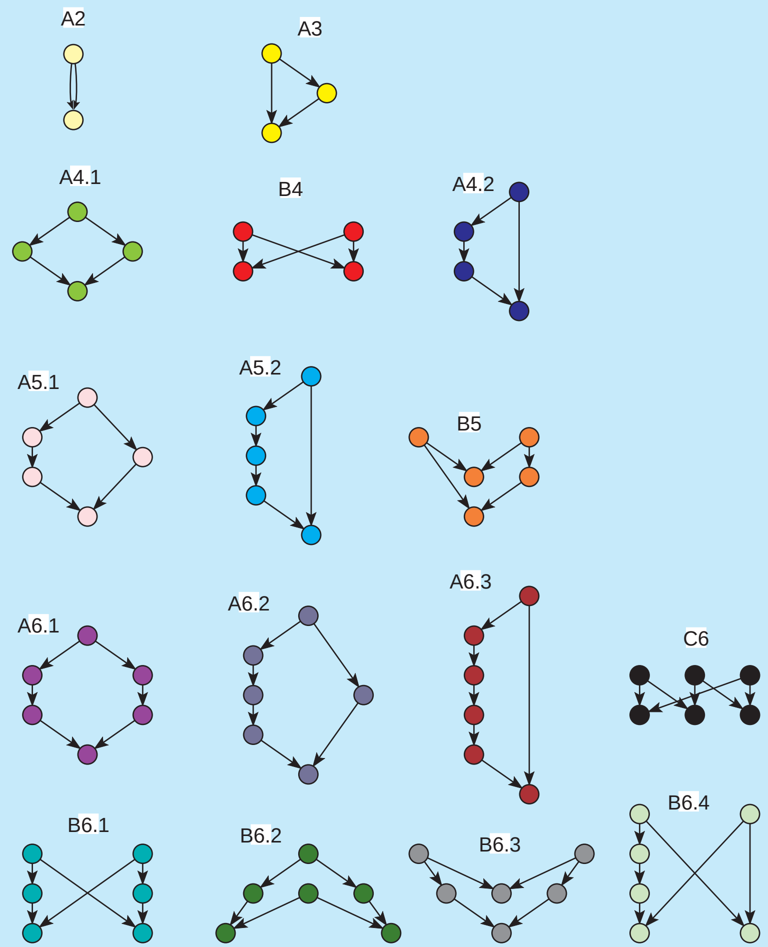
In Ore-graph every person is represented by a vertex, marriages, relation _ ***is a spouse of*** _, are represented with edges and relations _ ***is a mother of*** _ and _ ***is a father of*** _ as arcs pointing from parents to their children.

p-graph



In p-graph vertices represent individuals or couples. In the case that a person is not married yet (s)he is represented by a vertex, otherwise person is represented with the partner in a common vertex. There are only arcs in p-graphs – they point from children to their parents, representing the relations ***FiC*** *_* ***is a daughter of*** *_* and ***MiC*** *_* ***is a son of*** *_*; where ***FiC*** $\equiv$ **f**emale **i**n the **c**ouple; and ***MiC*** $\equiv$ **m**ale **i**n the **c**ouple.

Relinking patterns in p-graphs



All possible relinking marriages in p-graphs with 2 to 6 vertices. Patterns are labeled as follows:

- first character – number of first vertices: A – single, B – two, C – three.
- second character: number of vertices in pattern (2, 3, 4, 5, or 6).
- last character: identifier (if the two first characters are identical).

Patterns denoted by A are exactly the blood marriages. In every pattern the number of first vertices equals to the number of last vertices.

Frequencies normalized with number of couples in p-graph $\times$ 1000.

pattern	Loka	Silba	Ragusa	Turcs	Royal
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

Most of the relinking marriages happened in the genealogy of Turkish nomads; the second is Ragusa while in other genealogies they are much less frequent.

Multiplication of networks

To a simple two-mode *network* $\mathcal{N} = (I, J, E, w)$; where I and J are sets of *vertices*, E is a set of *edges* linking I and J, and $w : E \to \mathbb{R}$ (or some other semiring) is a *weight*; we can assign a *network matrix* $\mathbf{W} = [w_{i,j}]$ with elements: $w_{i,j} = w(i, j)$ for $(i, j) \in E$ and $w_{i,j} = 0$ otherwise.

Given a pair of compatible networks $\mathcal{N}_A = (I, K, E_A, w_A)$ and $\mathcal{N}_B = (K, J, E_B, w_B)$ with corresponding matrices $\mathbf{A}_{I\times K}$ and $\mathbf{B}_{K\times J}$ we call a *product of networks* $\mathcal{N}_A$ and $\mathcal{N}_B$ a network $\mathcal{N}_C = (I, J, E_C, w_C)$, where $E_C = \{(i, j) : i \in I, j \in J, c_{i,j} \neq 0\}$ and $w_C(i, j) = c_{i,j}$ for $(i, j) \in E_C$. The product matrix $\mathbf{C} = [c_{i,j}]_{I\times J} = \mathbf{A} * \mathbf{B}$ is defined in the standard way

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

In the case when $I = K = J$ we are dealing with ordinary one-mode networks (with square matrices).

Fast sparse matrix multiplication

The standard matrix multiplication has the complexity $O(|I| \cdot |K| \cdot |J|)$ – it is (usually) too slow to be used for large networks.

For sparse large networks we can multiply faster considering only nonzero elements:

for k **in** 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 $c_{i,j} := a_{i,k} * b_{k,j}$

$N_A(k)$: neighbors of vertex k in network A

$N_B(k)$: neighbors of vertex k in network B

In general the multiplication of large sparse networks is a 'dangerous' operation since the result can 'explode' – it is not sparse.

Complexity of fast sparse matrix multiplication

Let $\mathbf{A}$ and $\mathbf{B}$ be matrices of networks $\mathcal{N}_A = (\mathcal{I}, \mathcal{K}, \mathcal{E}_A, w_A)$ and $\mathcal{N}_B = (\mathcal{K}, \mathcal{J}, \mathcal{E}_B, w_B)$.

Assume that the body of the loops can be computed in the constant time c. Then we can prove:

If at least one of the sparse networks $\mathcal{N}_A$ and $\mathcal{N}_B$ has small maximal degree on K then also the resulting product network $\mathcal{N}_C$ is sparse.

And after more detailed complexity analysis:

Let $d_{min}(k) = \min(\deg_A(k), \deg_B(k))$, $\Delta_{min} = \max_{k\in\mathcal{K}} d_{min}(k)$, $d_{max}(k) = \max(\deg_A(k), \deg_B(k))$, $\mathcal{K}(d) = \{k \in \mathcal{K} : d_{max}(k) \geq d\}$, $d^* = \text{argmin}_d(|\mathcal{K}(d)| \leq d)$ and $K^* = K(d^*)$.

If for the sparse networks $\mathcal{N}_A$ and $\mathcal{N}_B$ the quantities Δ_{min} and d^* are small then also the resulting product network $\mathcal{N}_C$ is sparse.

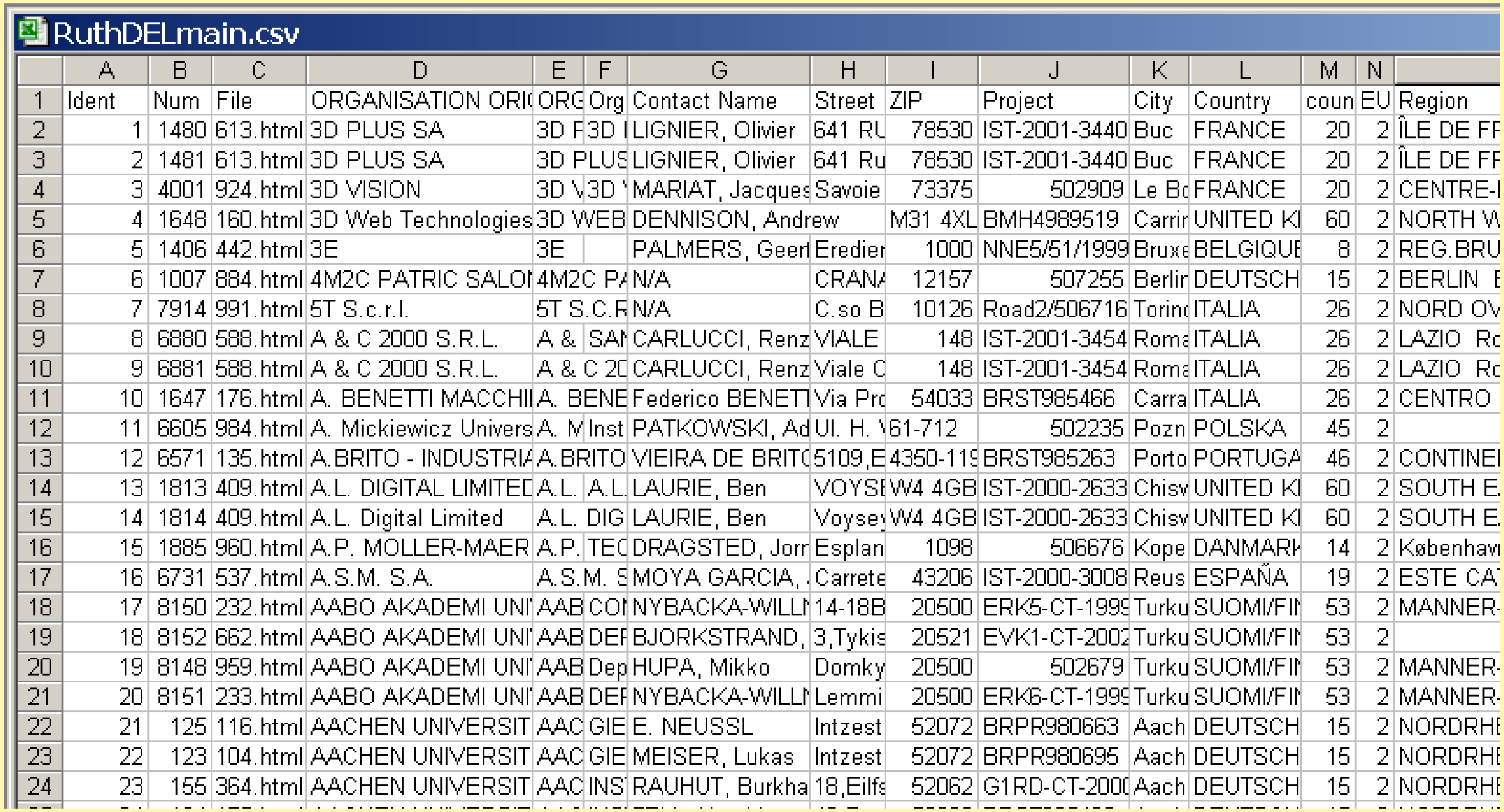
2-mode network analysis by conversion to 1-mode network

Often we transform a 2-mode network into an ordinary (1-mode) network $\mathbf{N}_1 = (\mathcal{U}, \mathcal{E}_1, w_1)$ or/and $\mathbf{N}_2 = (\mathcal{V}, \mathcal{E}_2, w_2)$, where $\mathcal{E}_1$ and w_1 are determined by the matrix $\mathbf{A}^{(1)} = \mathbf{A}\mathbf{A}^T$, $a^{(1)}_{uv} = \sum_{z\in\mathcal{V}} a_{uz} \cdot a^T_{zv}$. Evidently $a^{(1)}_{uv} = a^{(1)}_{vu}$. There is an edge $\{u, v\} \in \mathcal{E}_1$ in $\mathbf{N}_1$ iff $N(u) \cap N(v) \neq \emptyset$. Its weight is $w_1(u, v) = a^{(1)}_{uv}$.

The network $\mathbf{N}_2$ is determined in a similar way by the matrix $\mathbf{A}^{(2)} = \mathbf{A}^T\mathbf{A}$.

The networks $\mathbf{N}_1$ and $\mathbf{N}_2$ are analyzed using standard methods.

Networks from data tables



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 FR
3	2	1481	613.html	3D PLUS SA	3D PLUS		LIGNIER, Olivier	641 Ru	78530	IST-2001-3440	Buc	FRANCE	20	2	ÎLE DE FR
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 SALON	4M2C PA		N/A	CRANA	12157	507255	Berlin	DEUTSCH	15	2	BERLIN B
8	7	7914	991.html	5T S.c.r.l.	5T S.C.R		N/A	C.so B	10126	Road2/506716	Torino	ITALIA	26	2	NORD OV
9	8	6880	588.html	A & C 2000 S.R.L.	A &	SAN	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 MACCHI	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. W	61-712	502235	Pozn	POLSKA	45	2	
13	12	6571	135.html	A.BRITO - INDUSTRIA	A.BRITO		VIEIRA DE BRITO	5109,E	4350-119	BRST985263	Porto	PORTUGA	46	2	CONTINE
14	13	1813	409.html	A.L. DIGITAL LIMITED	A.L.	A.L.	LAURIE, Ben	VOYSE	W4 4GB	IST-2000-2633	Chisw	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	Chisw	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	Københavr
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 CAT
18	17	8150	232.html	AABO AKADEMI UNI	AAB	CON	NYBACKA-WILLN	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-WILLN	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

A *data table* $\mathcal{T}$ is a set of *records* $\mathcal{T} = \{T_k : k \in K\}$, where K is the set of *keys*. A record has the form $T_k = (k, q_1(k), q_2(k), \ldots, q_r(k))$ where $q_i(k)$ is the value of the *property* (attribute) $\mathbf{q}_i$ for the key k.

. . . Networks from data tables

Suppose that the property $\mathbf{q}$ has the range Q. If Q is finite (it can always be transformed in such set by partitioning the set Q and recoding the values) we can assign to the property $\mathbf{q}$ a two-mode network $K \times \mathbf{q} = (K, Q, E, w)$ where $(k, v) \in E$ iff $q(k) = v$, and $w(k, v) = 1$.

Also, for properties $\mathbf{q}_i$ and $\mathbf{q}_j$ we can define a two-mode network $\mathbf{q}_i \times \mathbf{q}_j = (Q_i, Q_j, E, w)$ where $(u, v) \in E$ iff $\exists k \in K : (q_i(k) = u \wedge q_j(k) = v)$, and $w(u, v) = \mathrm{card}(\{k \in K : (q_i(k) = u \wedge q_j(k) = v)\})$.

We define $[\mathbf{q}_i \times \mathbf{q}_j]^T = \mathbf{q}_j \times \mathbf{q}_i$.

It holds $\mathbf{q}_i \times \mathbf{q}_j = [K \times \mathbf{q}_i]^T * [K \times \mathbf{q}_j] = [\mathbf{q}_i \times K] * [K \times \mathbf{q}_j]$.

We can join a pair of properties $\mathbf{q}_i$ and $\mathbf{q}_j$ also with respect to the third property $\mathbf{q}_s$: we get a two-mode network $[\mathbf{q}_i \times \mathbf{q}_j]/\mathbf{q}_s = [\mathbf{q}_i \times \mathbf{q}_s] * [\mathbf{q}_s \times \mathbf{q}_j]$.

EU projects on simulation

For the meeting ***The Age of Simulation*** at Ars Electronica in Linz, January 2006 a dataset of EU projects on simulation was collected by FAS research, Vienna and stored in the form of Excel table (`RuthDELmain.csv`).

The rows are the projects participants (idents) and colomns correspond to different their properties. We produced from this table three two-mode networks using Jürgen Pfeffer's **`Text2Pajek`** program:

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

$|\text{idents}| = 8869$, $|\text{projects}| = 933$, $|\text{institutions}| = 3438$, $|\text{countries}| = 60$.

EU projects – network multiplication

Since all three networks have the common set (idents) we can derive from them using *network multiplication* several interesting networks:

- `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 $\mathbf{Q} = \mathbf{W}^T \star \mathbf{W}$
- ...

Analysis of `ProjInst.net`

For identifying important parts of `ProjInst.net` we first computed the 4-rings weights and in the obtained network we determined the line islands

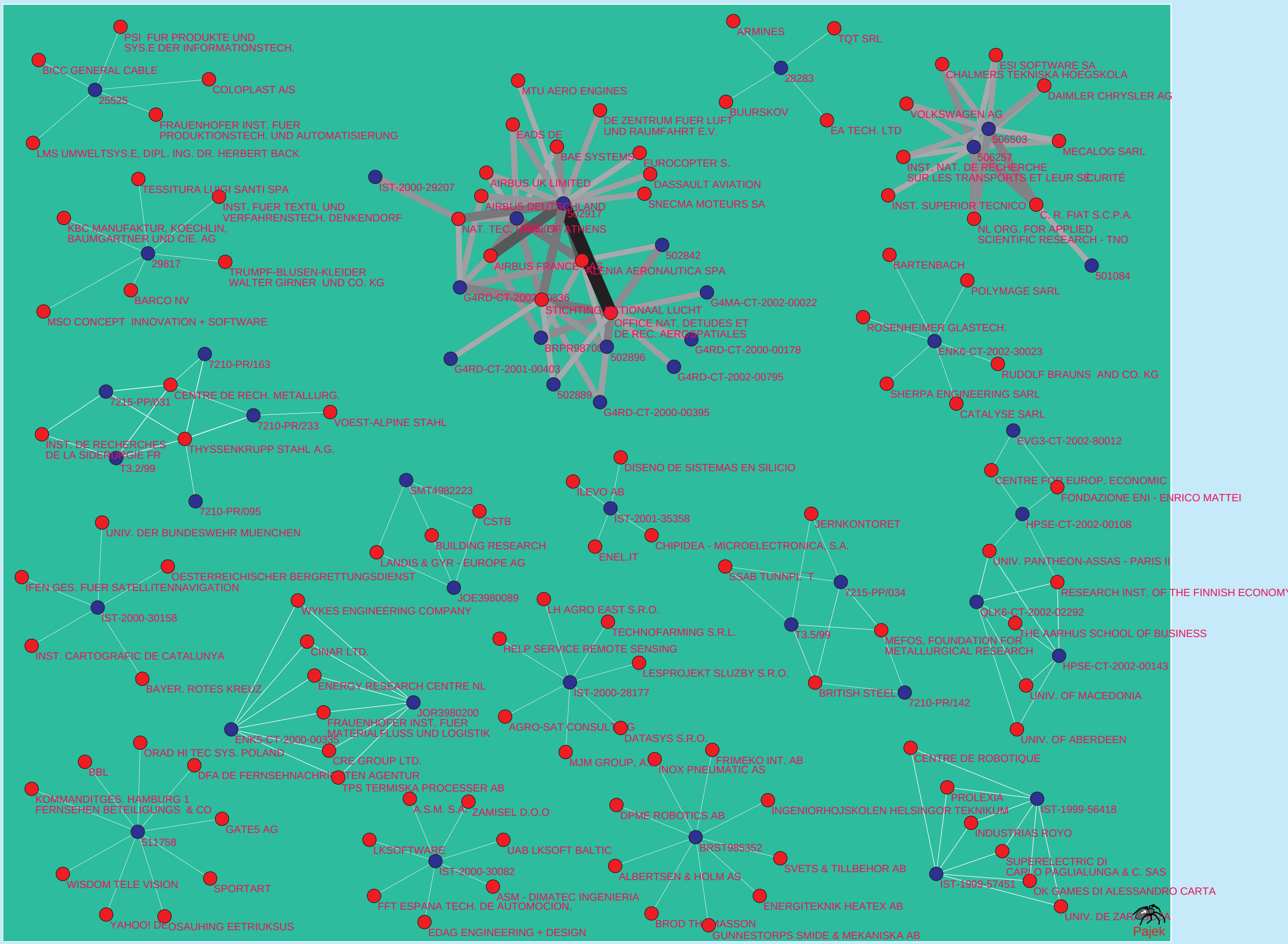
```
Net/Count/4-rings/Undirected
Net/Partitions/Islands/Line Weights[Simple [2,200]
```

We obtain 101 islands. We extracted 18 islands of the size at least 5. There are two most important islands: aviation companies and car companies.

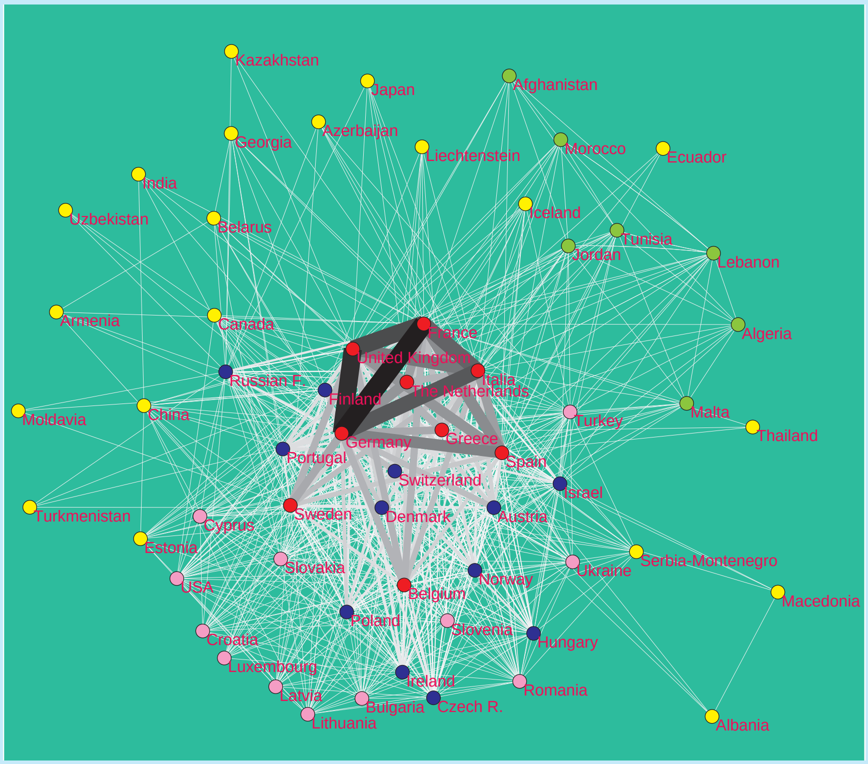
In labels we used a new option `\n`.

For analysis of two-mode networks we can use also $(p, q)-$cores.

Analysis of `ProjInst.net`



Analysis of `Countries.net`



To obtain picture in which the stronger lines cover weaker lines we have to sort them

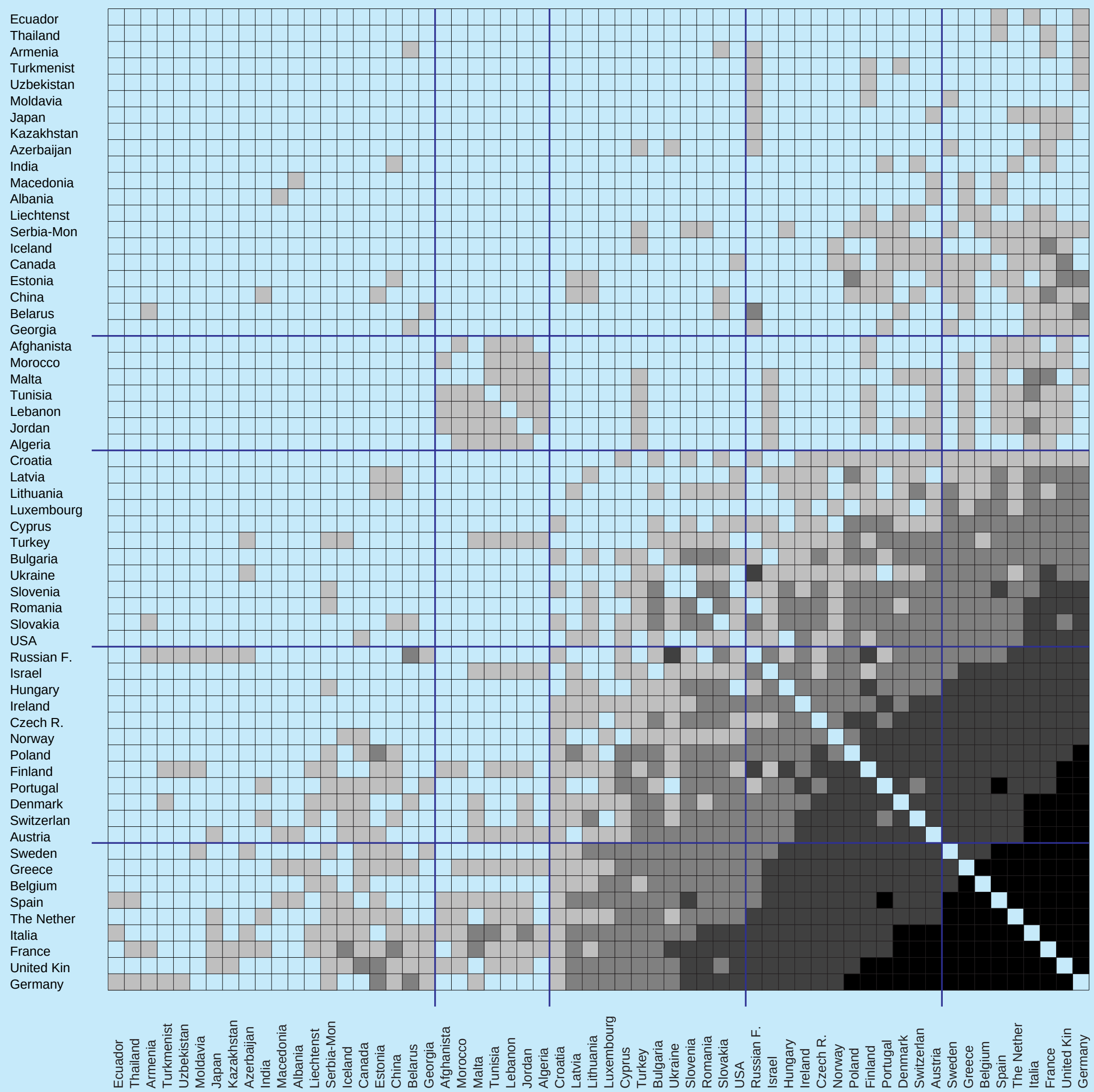
`Net/Transform/Sort lines/Line values/Ascending`

For dense (sub)networks we get better visualization by using matrix display. In this case we also recoded values (2,10,50). To determine clusters we used Ward's clustering procedure with dissimilarity measure d_5 (corrected Euclidean distance).

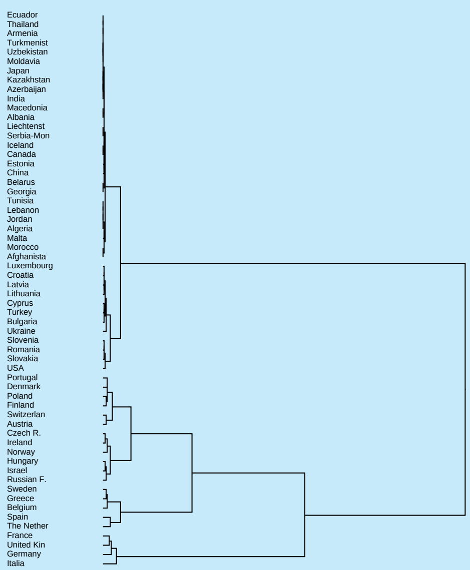
The permutation determined by hierarchy can often be improved by changing the positions of clusters – for the New Year 2006 Andrej added this option in Pajek. We get a typical center-periphery structure.

Analysis of `Countries.net`

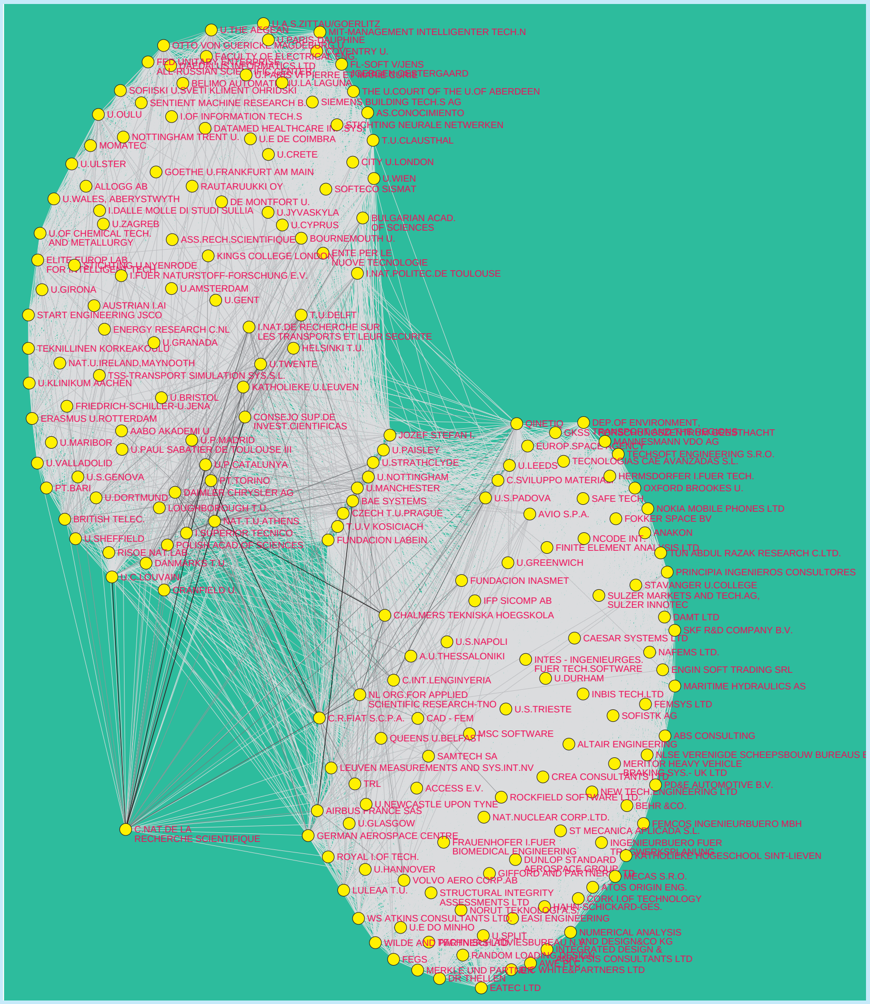
Pajek - shadow [0.00,4.00]



Pajek - Ward [0.00,4785.14]

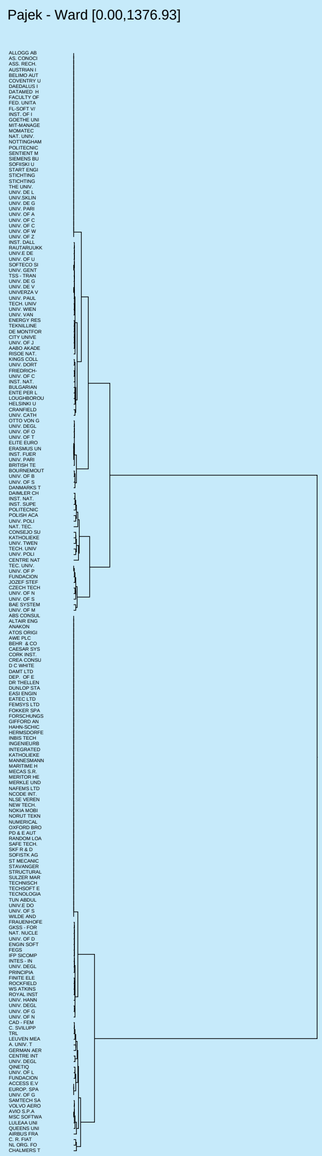


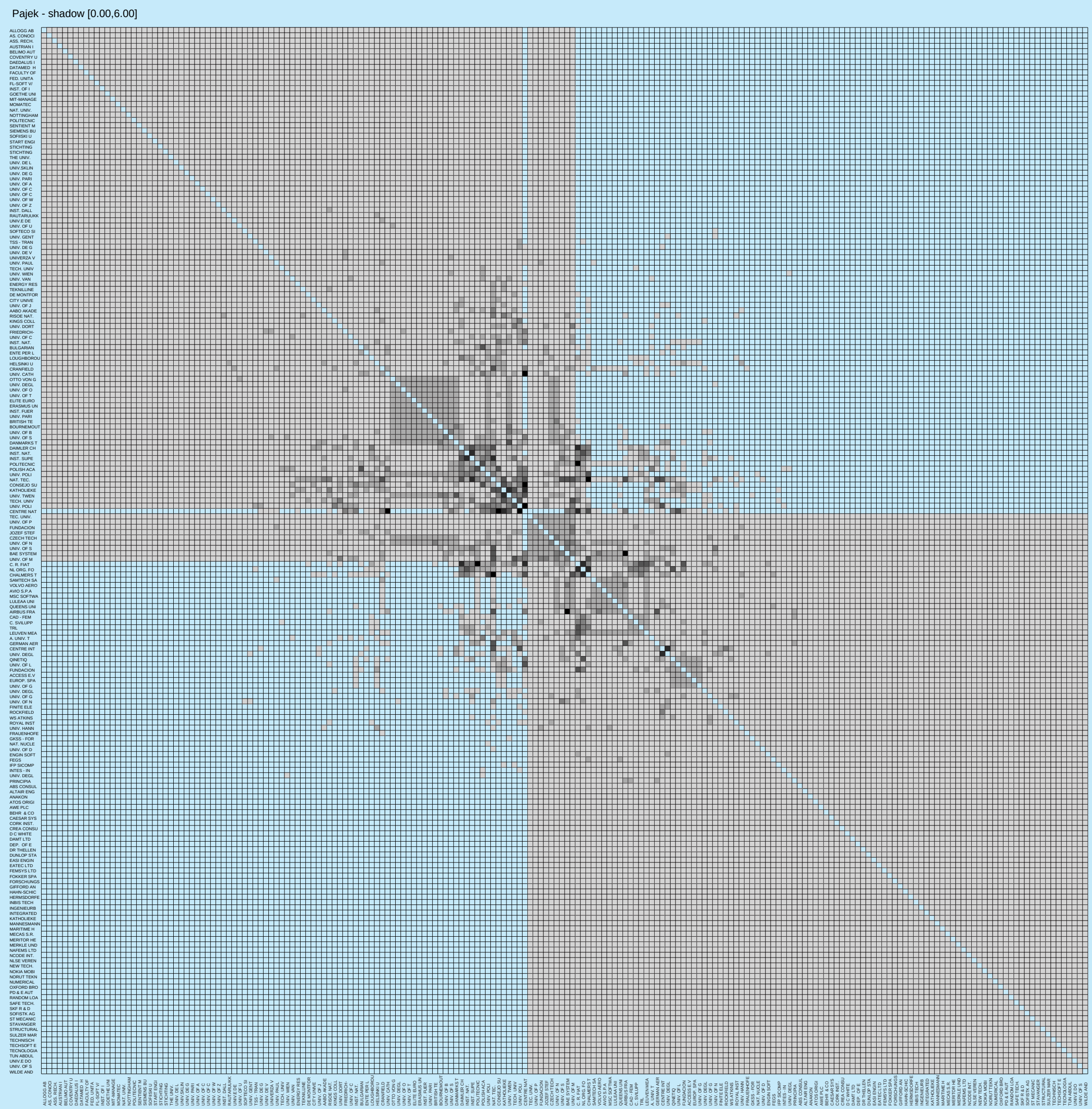
Analysis of `Institutions.net`



To identify the most important institutions we first computed p_S-cores vector and use it to determine the corresponding vertex islands. We got essentially one large island. Again the corresponding subnetwork is very dense. We prepared also a matrix display.

Analysis of `Institutions.net`





Clustering with relational constraints

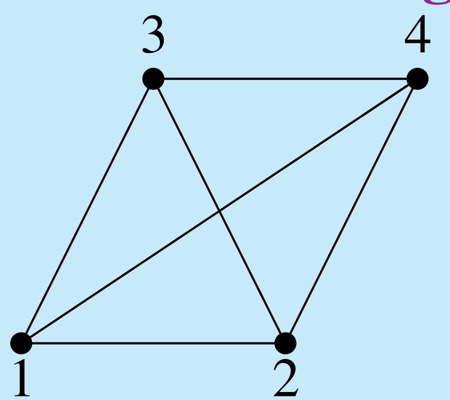
We can define different types of sets of feasible clusterings for the same relation R. Some examples of *types of relational constraint* $\Phi^i(R)$ are

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

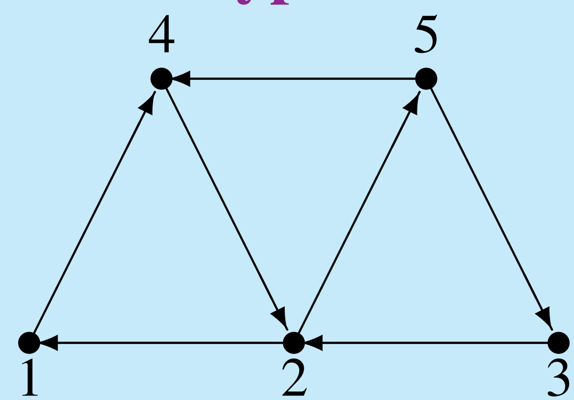
Trail – all arcs are distinct.

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 $R(L) \cap (C \setminus L) = \emptyset$.

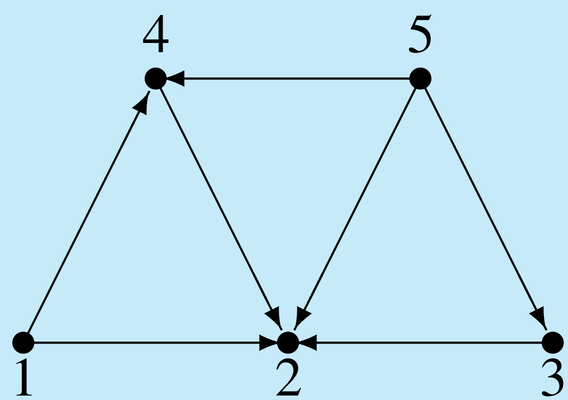
Some graphs of different types



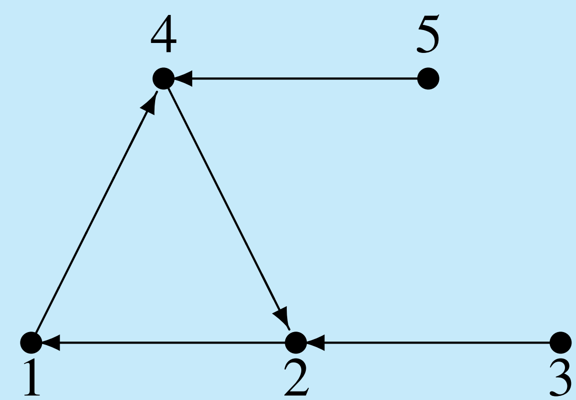
a clique



strongly connected units



weakly connected units



weakly connected units
with a center $\{1, 2, 4\}$

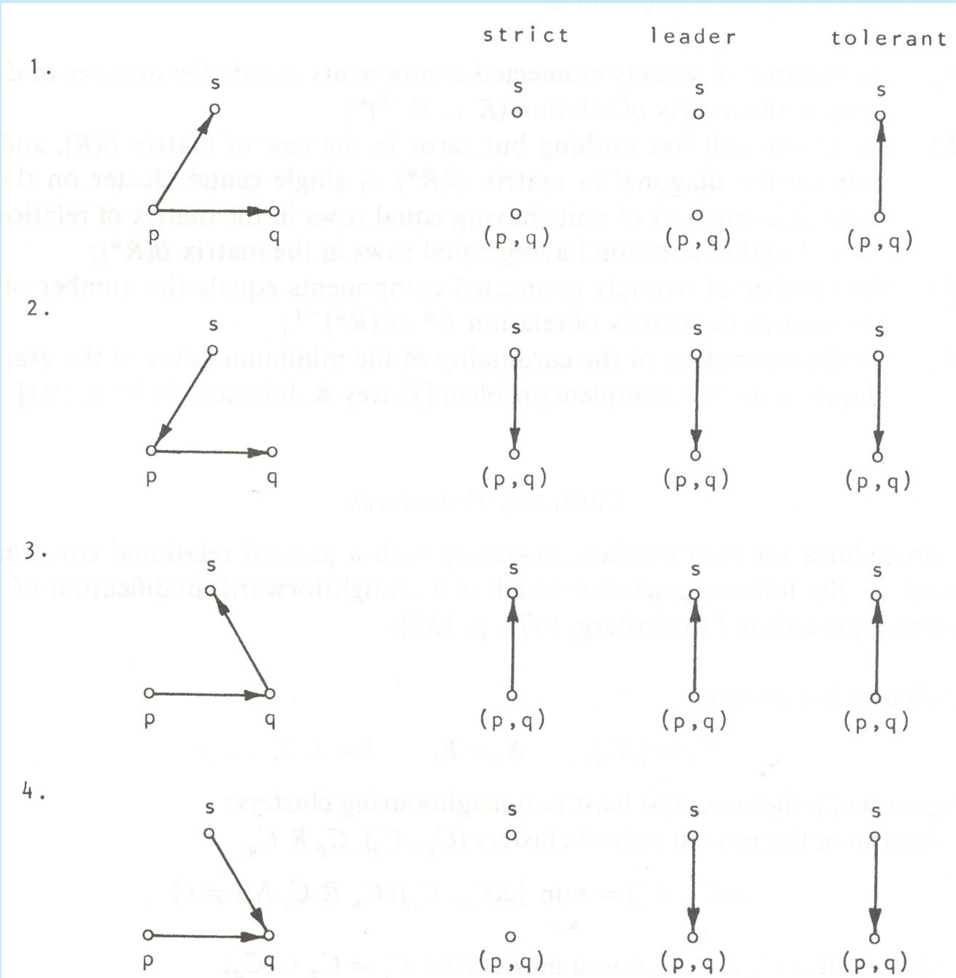
Agglomerative method for relational constraints

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** $\exists C_i, C_j \in \mathbf{C}(k)\colon (i \neq j \wedge \psi(C_i, C_j))$ **repeat**
2.1. $(C_p, C_q) := \mathrm{argmin}\{D(C_i, C_j)\colon i \neq j \wedge \psi(C_i, C_j)\};$
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 fusibility 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

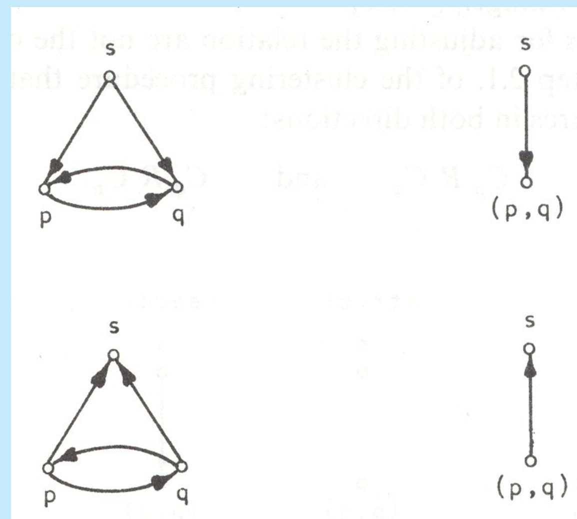


Φ^1 – tolerant

Φ^2 – leader

Φ^4 – two-way

Φ^5 – strict



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

$$D_{\min}(S,T) = \min_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$\begin{aligned} D_{\min}(S,T_1\cup T_2) &= \min_{(s,t)\in\hat{R}(S,T_1\cup T_2)} d(s,t) = \\ &= \min\left(\min_{(s,t)\in\hat{R}(S,T_1)} d(s,t), \min_{(s,t)\in\hat{R}(S,T_2)} d(s,t)\right) = \\ &= \min(D_{\min}(S,T_1), D_{\min}(S,T_2)) \end{aligned}$$

Maximum

$$D_{\max}(S,T) = \max_{(s,t)\in \hat{R}(S,T)} d(s,t)$$

$$\begin{aligned}
D_{\max}(S,T_1\cup T_2) &= \max_{(s,t)\in \hat{R}(S,T_1\cup T_2)} d(s,t) = \\
&= \max(\max_{(s,t)\in \hat{R}(S,T_1)} d(s,t), \max_{(s,t)\in \hat{R}(S,T_2)} d(s,t)) = \\
&= \max(D_{\max}(S,T_1), D_{\max}(S,T_2))
\end{aligned}$$

Average

$w : V \to \mathbb{R}$ – is a weight on units; for example $w(v) = 1$, for all $v \in \mathcal{U}$.

$$D_{\mathrm{a}}(S,T) = \frac{1}{w(\hat{R}(S,T))} \sum_{(s,t)\in\hat{R}(S,T)} d(s,t)$$

$$w(\hat{R}(S,T_1 \cup T_2)) = w(\hat{R}(S,T_1)) + w(\hat{R}(S,T_2))$$

$$w(\hat{R}(S,T_1 \cup T_2))D_{\mathrm{a}}(S,T_1 \cup T_2) = \sum_{(s,t)\in\hat{R}(S,T_1\cup T_2)} d(s,t) =$$
$$= \sum_{(s,t)\in\hat{R}(S,T_1)} d(s,t) + \sum_{(s,t)\in\hat{R}(S,T_2)} d(s,t) =$$
$$= w(\hat{R}(S,T_1)) \cdot D_{\mathrm{a}}(S,T_1) + w(\hat{R}(S,T_2)) \cdot D_{\mathrm{a}}(S,T_2))$$

$$D_{\mathrm{a}}(S,T_1\cup T_2) = \frac{w(\hat{R}(S,T_1))}{w(\hat{R}(S,T_1 \cup T_2))} D_{\mathrm{a}}(S,T_1) + \frac{w(\hat{R}(S,T_2))}{w(\hat{R}(S,T_1 \cup T_2))} D_{\mathrm{a}}(S,T_2)$$

Reducibility

The dissimilarity D has the *reducibility* property (Bruynooghe, 1977) iff

$$D(C_p, C_q) \leq \min(D(C_p, C_s), D(C_q, C_s)) \;\Rightarrow$$

$$\min(D(C_p, C_s), d(C_q, C_s)) \leq D(C_p \cup C_q, C_s)$$

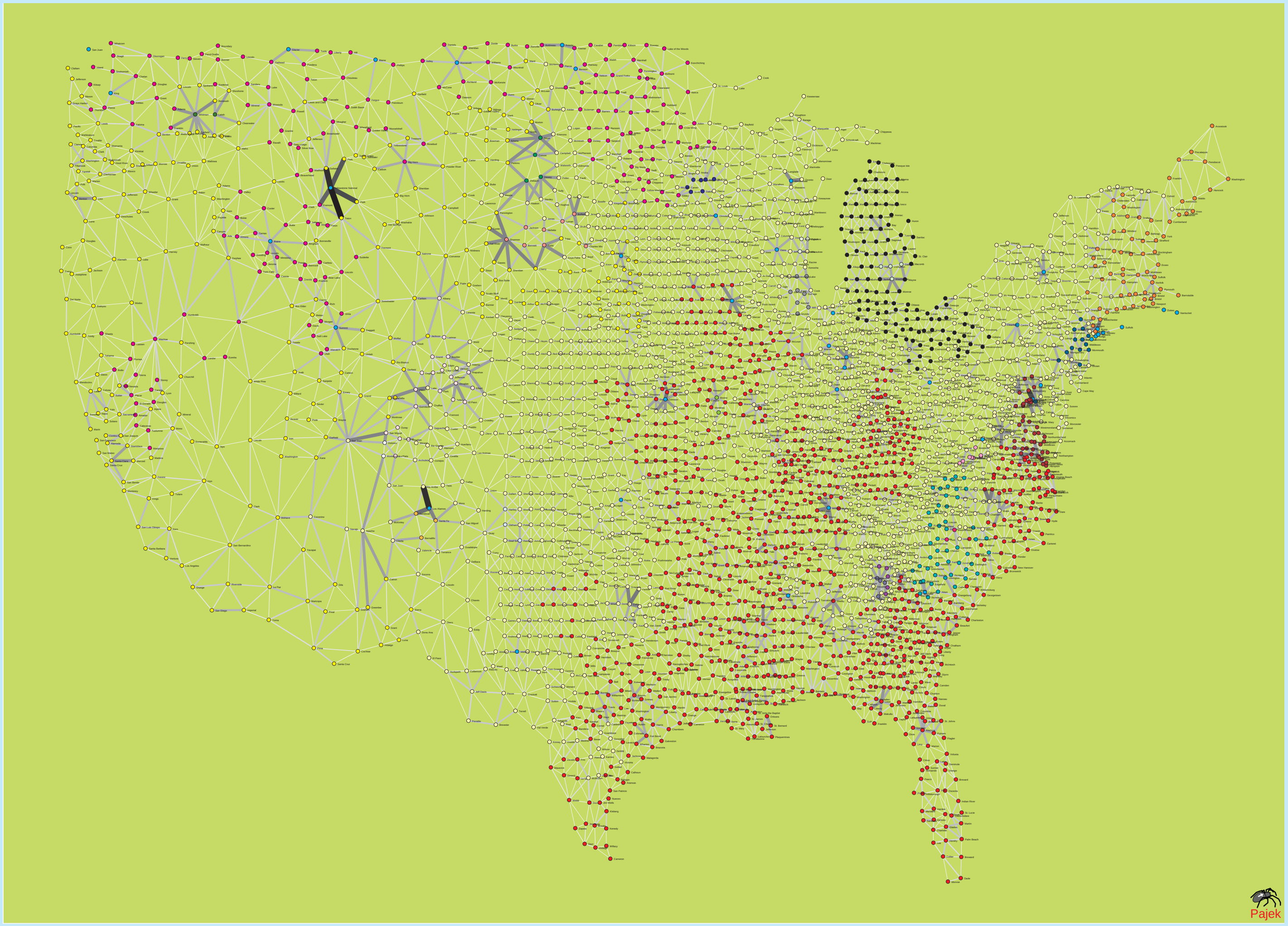
or equivalently

$$D(C_p, C_q) \leq t,\; D(C_p, C_s) \geq t,\; D(C_q, C_s) \geq t \;\Rightarrow\; D(C_p \cup C_q, C_s) \geq t$$

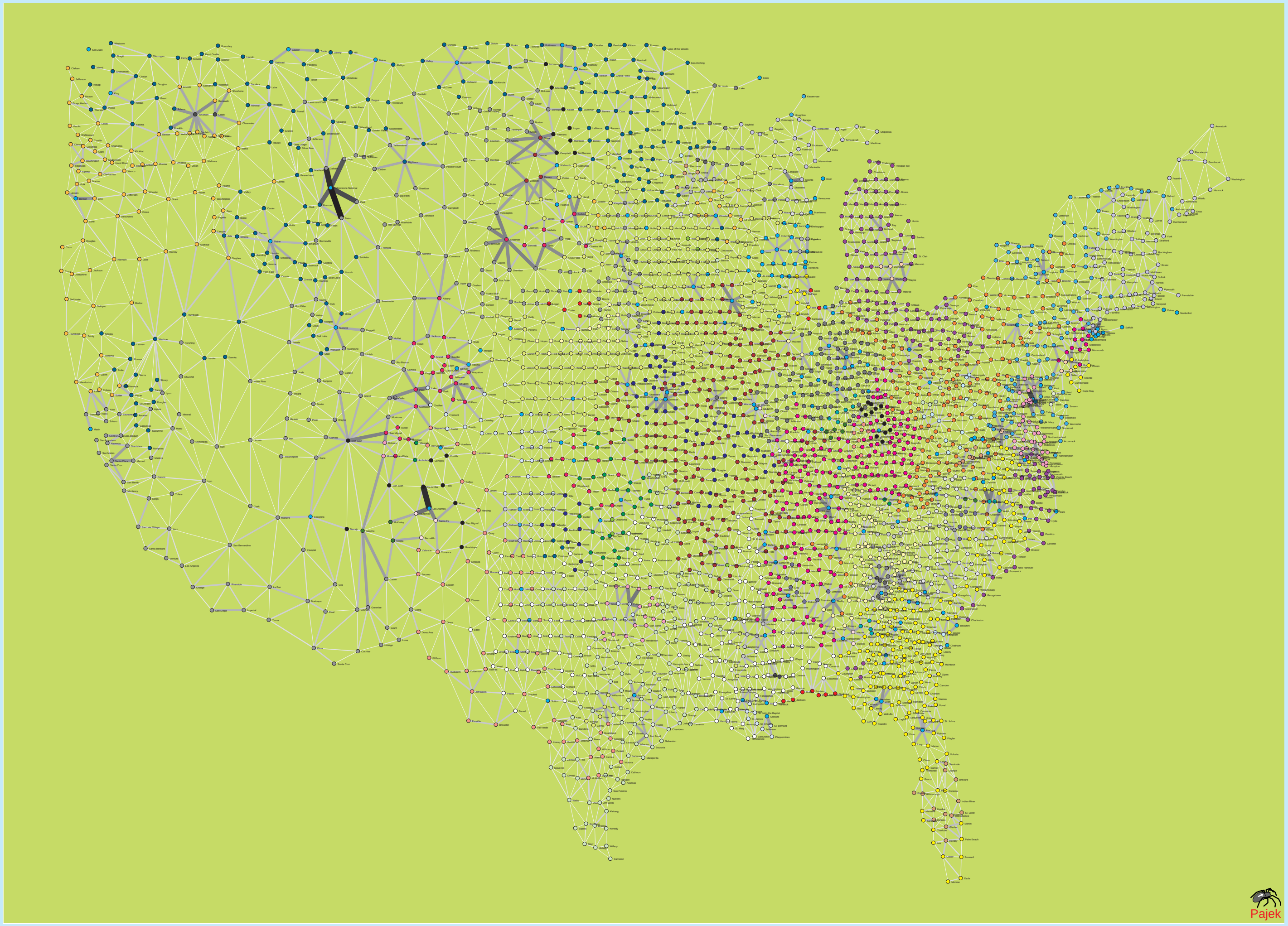
Theorem 1 *If a dissimilarity D has the reducibility property then h_D is a level function.*

All three disimilarities have the *reducibility* property. In this case also the nearest neighbor network for a given network is preserved after joining the nearest clusters. This allows us to develop a very fast agglomerative hierarchical clustering procedure.

Example: US counties $t = 1400$



Example: US counties $t = 200$



What else?

In 2005 we introduced in **Pajek** also support for *multi-relational* networks that combined with *temporal* networks enable analysis of new kinds of networks – such as KEDS networks (*Kansas Event Data System* or *Tabari*).

You can use URLs in description of vertices (Nov 2005).

References

1. Batagelj, V. and Mrvar, A.(1996-): **Pajek**– *program for analysis and visualization of large network*, home page, data sets.
2. Batagelj, V. and Zaveršnik, M.(2002): *Generalized Cores*, arxiv cs.DS/0202039
3. Batagelj, V. and Zaveršnik, M.(2003): *Short cycle connectivity*. arxiv cs.DS/0308011
4. Batagelj V., Ferligoj A. (2000): Clustering relational data. Data Analysis (Eds.: W. Gaul, O. Opitz, M. Schader), Springer, Berlin, 3–15.
5. Bruynooghe, M. (1977), Méthodes nouvelles en classification automatique des données taxinomiques nombreuses. Statistique et Analyse des Données, **3**, 24–42.
6. Doreian, P., Batagelj, V., Ferligoj, A. (2000), *Symmetric-acyclic decompositions of networks*. *J. classif.*, **17**(1), 3–28.
7. Ferligoj A., Batagelj V. (1982): *Clustering with relational constraint*. *Psychometrika*, **47**(4), 413–426.
8. Ferligoj A., Batagelj V. (1983): *Some types of clustering with relational constraints*. *Psychometrika*, **48**(4), 541–552.
9. Mrvar, A. and Batagelj, V. (2004): *Relinking Marriages in Genealogies*. *Metodološki zvezki - Advances in Methodology and Statistics*, **1**, Ljubljana: FDV, 407-418.
10. Murtagh, F. (1985), Multidimensional Clustering Algorithms, *Compstat lectures*, **4**, Vienna: Physica-Verlag.
11. de Nooy, W., Mrvar, A. and Batagelj V. (2005): *Exploratory Social Network Analysis with Pajek*, CUP. Amazon. ESNA page.

12. J. P Scott: *Social Network Analysis: A Handbook*. SAGE Publications, 2000. Amazon.

13. S. Wasserman, K. Faust: *Social Network Analysis: Methods and Applications*. CUP, 1994. Amazon.

14. White, D.R., Batagelj, V., and Mrvar, A. (1999): *Analyzing Large Kinship and Marriage Networks with Pgraph and Pajek*. *Social Science Computer Review – SSCORE*, **17**, 245-274.

15. Zaveršnik, M. and Batagelj, V. (2004): *Islands*. Slides from *Sunbelt XXIV, Portorož, Slovenia, 12.-16. May 2004*, PDF