# Social Network Analysis, Large-scale[1]

VLADIMIR BATAGELJ
University of Ljubljana, Ljubljana, Slovenia

## Article Outline



## Glossary

For the basic notions on graphs and networks see the article Wouter de Nooy: Social network analysis.

Network – consists of vertices linked by lines and additional data about vertices and/or lines.

Network decomposition – identification of parts of network and their interconnections. Usually it is described by a partition of set of vertices or set of lines.

[1]To be published as a chapter in the Encyclopedia of Complexity and System Science (editor-in-chief Bob Meyers), in the Social Networks section (section editor John Scott), Springer Verlag, 2009.
version: April 7, 2008 / 11 : 38

Time complexity of algorithm – describes how the time needed to run the algorithm depends on the size of the input data.

Reduction of network – a network obtained by shrinking each cluster from a given partition into a vertex.

Condensation – a reduction for strong connectivity partition.

Cut – a subnetwork of vertices/lines with values of selected property above given threshold.

Island – a connected subnetwork of selected size of (locally) important, with respect to selected property, vertices/lines.

Pattern searching – identification of all appearences of selected small subnetwork (pattern or fragment) in a given network.

Topological sort – procedure to determine a compatible ordering in acyclic network.

## I. Definition of the Subject

A *network* is based on two sets – set of *vertices* (nodes), that represent the selected *units*, and set of *lines* (links), that represent *ties* between units. Each line has two vertices as its *end-points*; if they are equal it is called a *loop*. Vertices and lines form a *graph*. A line can be *directed* – an *arc*, or *undirected* – an *edge*.

Additional data about vertices or lines are usually known – their *properties* (attributes). For example: name/label, type, value, position, ... In general

$$\textbf{Network} = \textbf{Graph} + \textbf{Data}$$

The data can be measured or computed.

Formally, 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 and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$, $\mathcal{E} \cap \mathcal{A} = \emptyset$ is the set of lines; $\mathcal{A}$ is the set of *arcs* and $\mathcal{E}$ is the set of *edges*.
- $\mathcal{P}$ – set of *vertex value functions* or properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ – set of *line value functions* or weights: $w : \mathcal{L} \to B$

The size of a network/graph is expressed by two numbers: number of vertices $n = |\mathcal{V}|$ and number of lines $m = |\mathcal{L}|$. In a *simple undirected* graph (no parallel edges, no loops) $m \leq \frac{1}{2}n(n-1)$; and in a *simple directed* graph (no parallel arcs) $m \leq n^2$.

For a family of graphs $\mathbb{G}$ we define a *density* of graph $\mathcal{G}$ as $\gamma(\mathcal{G}) = \frac{m(\mathcal{G})}{m_{max}(\mathbb{G})}$.

## II. Introduction

*Small* networks (some tens of vertices) – can be represented by a picture and analyzed by many algorithms (***UCINET*** [64], ***NetMiner*** [61]). Also *middle size* networks (some hundreds of vertices), if they are not dense, can still be represented by a picture, but some analytical procedures can't be used.

Till 1990 most networks were small – they were collected by researchers using surveys, observations, archival records, ... The advances in IT allowed to create networks from the data already available in the computer(s) or by browsing on the Internet. *Large* networks became reality. Large networks are too big to be displayed in details; special algorithms are needed for their analysis (**Pajek** [62]). The availability of large data sets also provided incentives to the boost of theoretical research in (large) network analysis (not only in social science).

A recent overview of social network analysis software is given in Huisman and Van Duijn [22].

## III. Large Networks and Complexity of Algorithms

*Large* networks have several thousands or millions of vertices. The upper limit to their size is tehnologically dependent – they can be stored in computer's memory; otherwise we deal with a *huge* network (see Abello et al. [38]).

Large networks are usually sparse $m \ll n^2$; typically $m = O(n)$ or $m = O(n \log n)$ .

| network | $n = \lvert\mathcal{V}\rvert$ | $m = \lvert\mathcal{L}\rvert$ | source |
|---|---|---|---|
| ODLIS dictionary | 2909 | 18419 | ODLIS online |
| Citations SOM | 4470 | 12731 | Garfield's collection |
| Molecula 1ATN | 5020 | 5128 | Brookhaven PDB |
| Comput. geometry | 7343 | 11898 | BiBTEX bibliographies |
| English words 2-8 | 52652 | 89038 | Knuth's English words |
| Internet traceroutes | 124651 | 207214 | Internet Mapping Project |
| Franklin genealogy | 203909 | 195650 | Roperld.com gedcoms |
| World-Wide-Web | 325729 | 1497135 | Notre Dame Networks |
| Internet Movie DB | 1324748 | 3792390 | IMDB |
| Wikipedia | 659388 | 16582425 | Wikimedia |
| US patents | 3774768 | 16522438 | Nber |
| SI internet | 5547916 | 62259968 | Najdi Si |

A collection of large networks is available from **Pajek**'s datasets [62].

The *time complexity* of an algorithm describes how the time needed to run the algorithm depends on the size of the input data. In computer science the problems for which only algorithms of exponential (or higher) complexity are

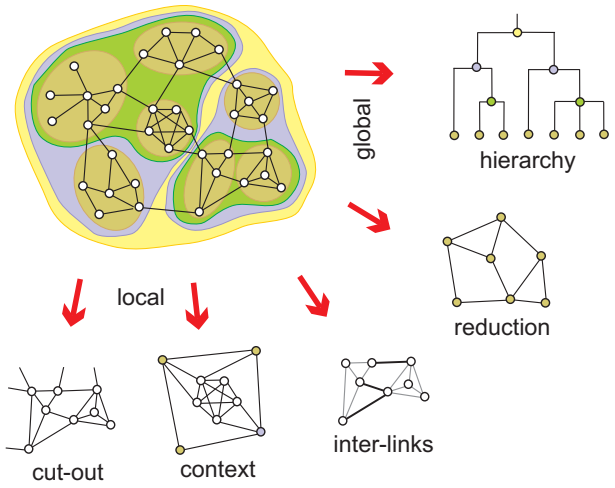


Figure 1: Decompositions

known are considered hard or intractable since the speed-up of computer only additively increases the size of problems that can be solved in a given period of time; but the problems for which an algorithm of polynomial complexity exists are considered 'nice'. When dealing with large instances of problems this isn't always true anymore. Let us look to time complexities of some typical algorithms:

| algorithm | T($n$) | 1.000 | 10.000 | 100.000 | 1.000.000 | 10.000.000 |
|---|---|---|---|---|---|---|
| Alg-A | O($n$) | 0.00 s | 0.015 s | 0.17 s | 2.22 s | 22.2 s |
| Alg-B | O($n \log n$) | 0.00 s | 0.06 s | 0.98 s | 14.4 s | **2.8 m** |
| Alg-C | O($n\sqrt{n}$) | 0.01 s | 0.32 s | 10.0 s | **5.27 m** | **2.78 h** |
| Alg-D | O($n^2$) | 0.07 s | 7.50 s | **12.5 m** | **20.8 h** | **86.8 d** |
| Alg-E | O($n^3$) | 0.10 s | **1.67 m** | **1.16 d** | **3.17 y** | **3.17 ky** |

For the interactive use on large networks already quadratic algorithms, O($n^2$), are too slow – we have to restrict our 'toolbox' to a selection of efficient, *subquadratic* algorithms.

How can we deal with large structures? Already Romans knew – ***divide et impera*** (divide and conquer). In case of networks ***divide*** means the use of (recursive) *decomposition* of a large network into several smaller networks (see Figure 1) that can be visualized and treated further using more sophisticated methods; and ***impera*** means that we have to take care about the interlinks among so obtained parts.

Another approach is the use of different *statistical* quantities to describe the properties of a network and using probabilistic models to derive the answers to some questions.

## IV. Decompositions

Decompositions of a network are usually described by clusterings of vertices or lines. In the following we shall use mainly the clusterings of vertices.

A nonempty subset $C \subseteq \mathcal{V}$ is called a *cluster* (group). A nonempty set of clusters $\mathbf{C} = \{C_i\}$ forms a *clustering*.

Clustering $\mathbf{C} = \{C_i\}$ is a *partition* iff

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

Clustering $\mathbf{C} = \{C_i\}$ is a *hierarchy* iff $C_i \cap C_j \in \{\emptyset, C_i, C_j\}$. In other words, in a hierarchy two clusters are either disjoint or is one contained in the other.

Hierarchy $\mathbf{C} = \{C_i\}$ is *complete*, iff $\cup\mathbf{C} = \mathcal{V}$; and is *basic* if for all $v \in \cup\mathbf{C}$ also $\{v\} \in \mathbf{C}$.

*Contraction* of cluster $C$ in a graph $\mathcal{G}$ is called a graph $\mathcal{G}/C$, in which all vertices of the cluster $C$ are replaced by a single new vertex, say $c$. More precisely: $\mathcal{G}/C = (\mathcal{V}', \mathcal{L}')$, where $\mathcal{V}' = (\mathcal{V} \backslash C) \cup \{c\}$ and $\mathcal{L}'$ consists of lines from $\mathcal{L}$ that have both end-points in $\mathcal{V} \setminus C$. Beside these it contains also a 'star' with the center $c$ and: arc $(v, c)$, if $\exists p \in \mathcal{L}, u \in C : p(v, u)$; or arc $(c, v)$, if $\exists p \in \mathcal{L}, u \in C : p(u, v)$. There is a loop $(c, c)$ in $c$ if $\exists p \in \mathcal{L}, u, v \in C : p(u, v)$.

In a network over graph $\mathcal{G}$ we have also to specify how the new values/weights are determined in the shrunk part of the network. Usually as the sum or maksimum/minimum of the original values.

For a given partition if we contract all clusters except few selected we obtain their *context*; and if we contract all clusters we obtain the *reduction* of a given network.

On the left side of the Figure 2 the *matrix display* of Snyder and Kick's [33] international trade network is presented. Vertices in the display are reordered according to the partition by (sub)continents. On the right side the corresponding reduction of the network is presented. The lines in the reduction have the thickness proportional to the weights

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

where $n(C_i, C_j)$ is the number of lines from cluster $C_i$ to cluster $C_j$; and $n(C_i)$ is the number of lines inside the cluster $C_i$.

A *subgraph* $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ of a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ is a graph which set of lines is a subset of set of lines of $\mathcal{G}$, $\mathcal{L}' \subseteq \mathcal{L}$, its vertex set is a subset of set

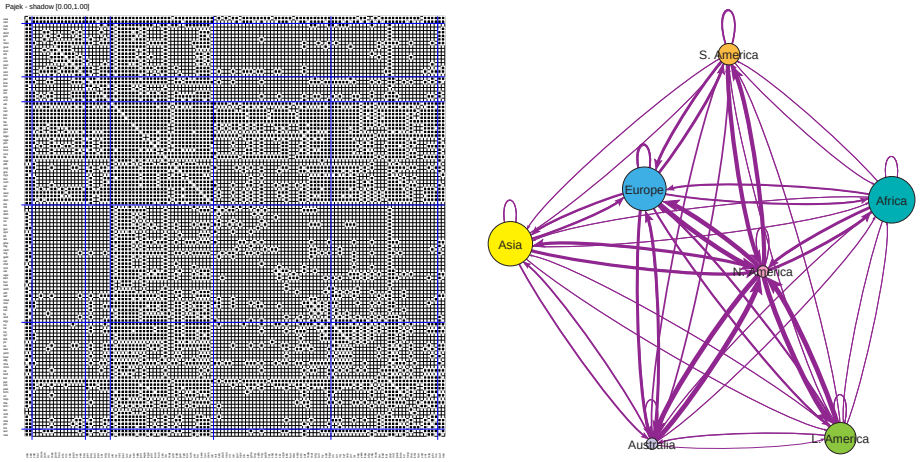


Figure 2: Snyder and Kick's international trade; matrix display and reduction

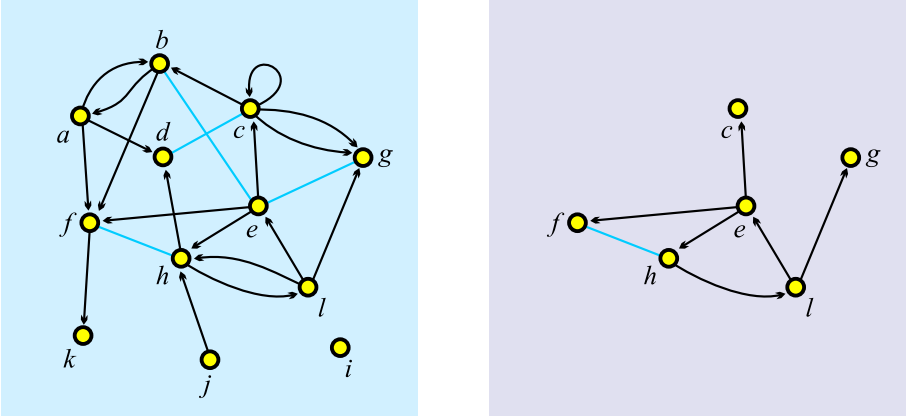


Figure 3: Graph and its subgraph

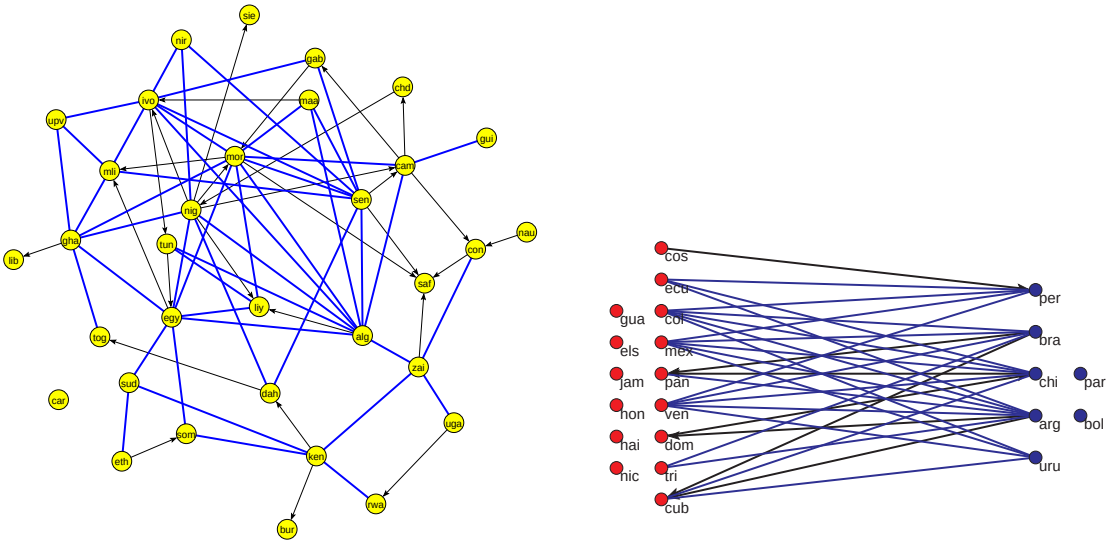


Figure 4: Africa cut-out and inter-links between South and Latin America

of vertices of $\mathcal{G}$, $\mathcal{V}' \subseteq \mathcal{V}$, and it contains all end-vertices of $\mathcal{L}'$. The graph on the right side of Figure 3 is a subgraph of the graph on the left side.

A subgraph can be *induced* by a given subset of vertices $\mathcal{V}'$, then $\mathcal{L}' = \mathcal{L}|\mathcal{V}'$ consists of all lines from $\mathcal{L}$ which have both end-points in $\mathcal{V}'$; or lines $\mathcal{L}'$, then $\mathcal{V}' = \mathcal{V}|\mathcal{L}'$ consists of all end-points of lines from $\mathcal{L}'$. It is a *spanning* subgraph iff $\mathcal{V}' = \mathcal{V}$.

On the left side of Figure 4 the *cut-out* of African countries from the Snyder and Kick's network is presented – the induced subgraph by Africa cluster; and on the right side the *inter-links* between Latin America and South America – the induced subgraph by Latin America and South America clusters with inside cluster lines removed.

## V. Connectivity

A *walk* from vertex $u$ to vertex $v$ is a sequence of lines $l(v_{i-1}, v_i)$, $i = 1, \ldots, k$ such that $v_0 = u$ and $v_k = v$. $k$ is called the *length* of the walk. If in the definition of a walk we don't care about the direction of its lines we get a *semiwalk*. A walk is *closed* iff $u = v$. A graph is *acyclic* iff it doesn't contain any closed walk. A walk in which all vertices are different is a *path*.

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

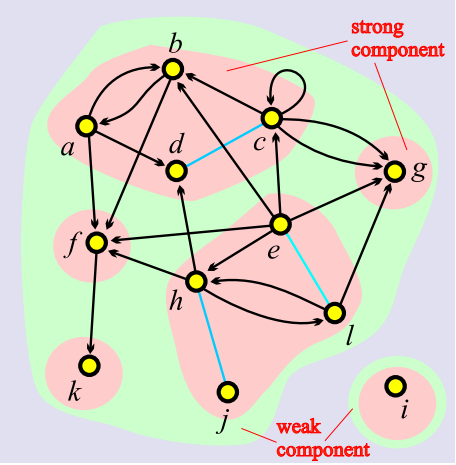


Figure 5: Weak and strong components

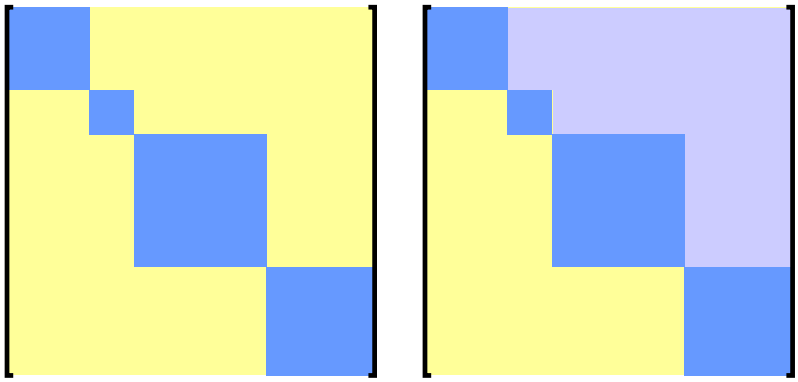

Figure 6: Weak and strong components in matrix display

vertex $u$ iff they are mutually reachable.

Weak and strong connectivity are equivalence relations. Equivalence classes induce weak/strong *components* (See Figure 5).

Reordering the vertices of network such that the vertices from the same class of weak partition are put together we get a matrix representation (left side of Figure 6) consisting of diagonal blocks – weak components. The out-diagonal blocks are zero-blocks. Most problems can be solved separately on each component and afterward these solutions combined into final solution.

If we shrink every strong component of a given graph into a vertex, delete all loops and identify parallel arcs the obtained reduced graph, called also the *condensation* of a given graph, is acyclic (Harary et al. [49]). For every acyclic graph an *ordering* / *level* function $i : \mathcal{V} \to \mathbb{N}$ exists s.t. $(u, v) \in \mathcal{A} \Rightarrow i(u) < i(v)$. The procedure to determine such ordering is called *topological sort* (Cormen et

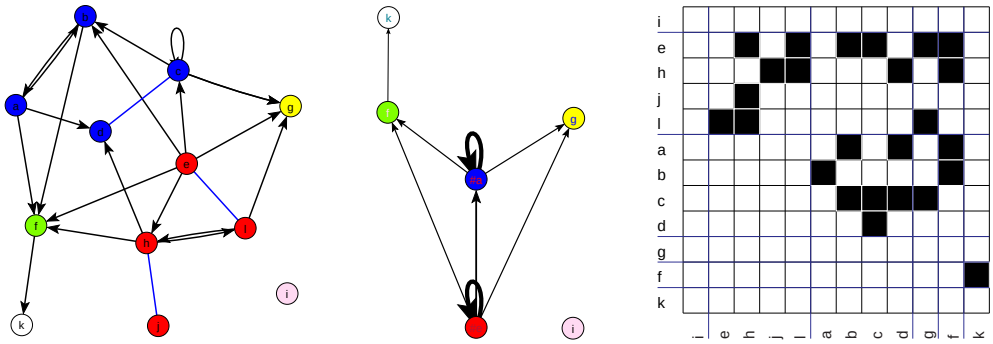


Figure 7: Condensation

al. [44]). Reordering in matrix display the vertices of a network by this ordering we obtain a representation as at the right side of Figure 6 – the blocks below the diagonal are zero-blocks.

A directed graph, its condensation and its topologically ordered matrix display are presented in Figure 7.

For several network analysis problems more efficient algorithms exist for acyclic networks.

## VI. Cuts

The basic approach to find interesting groups inside a network is to express our intentions (question) with an appropriate property/weight (measured or computed from network structure) and then identify the substructures of elements with the highest (lowest) values of the selected property. This approach is known as a method of *cuts*.

There exist several measures of importance of vertices in a network such as: degree, betweeness, closeness (Freeman [19]; Brandes [14]), hubs and authorities (Kleinberg [24]), clustering coefficient, ...

The *degree* $\deg(v)$ of vertex $v$ equals to the number of lines having vertex $v$ as their end-point. The *maximum degree* of a graph is denoted by $\Delta$. Similary the in-degree $\mathrm{indeg}(v)$ of vertex $v$ equals to the number of lines having vertex $v$ as their terminal point, and the out-degree $\mathrm{outdeg}(v)$ ...

The *vertex-cut* of a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, p)$, for a property $p : \mathcal{V} \to \mathbb{R}$, at selected level $t$ is a subnetwork $\mathcal{N}(t) = (\mathcal{V}', \mathcal{L}(\mathcal{V}'), p)$, 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 end-points in $\mathcal{V}'$.

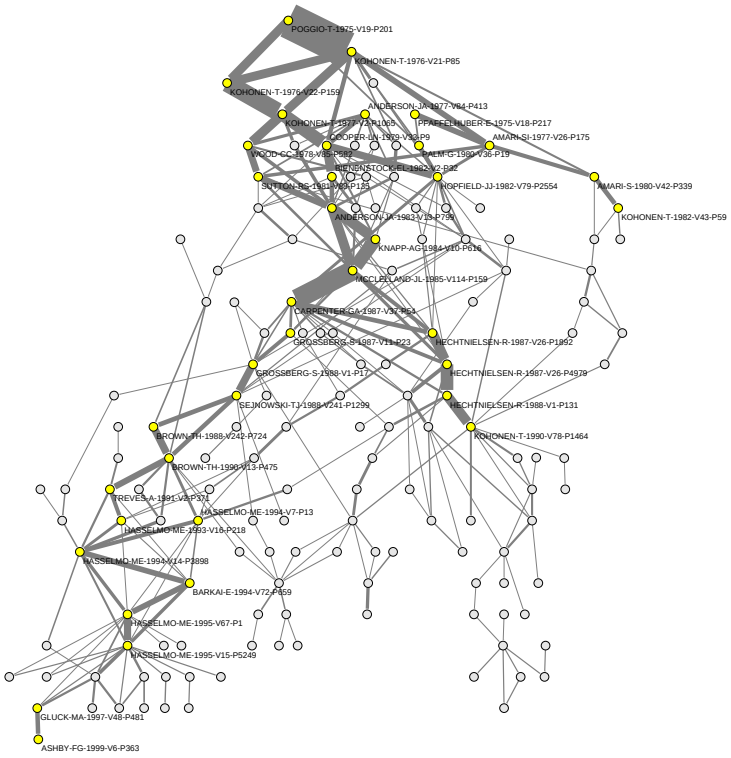


Figure 8: Main component of arc cut at level 0.007 of the SOM citation network

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

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

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

In the analysis of a cut $\mathcal{N}(t)$ we look at its components. 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 'less interesting'; and the components of size larger than $K$ are cut again at some higher level.

The values of threshold $t$ and size bounds $k$ and $K$ are determined by inspecting the distribution of vertex/line-values and the distribution of component sizes and considering additional knowledge on the nature of network or goals of analysis.

The $p_S$-core at level 46 (see Figure 11) of the collaboration network in the field of computational geometry is an example of vertex cut.

The citation network analysis started in 1964 with the paper of Garfield et al. [20]. In 1989 Hummon and Doreian [23] proposed three indices – weights

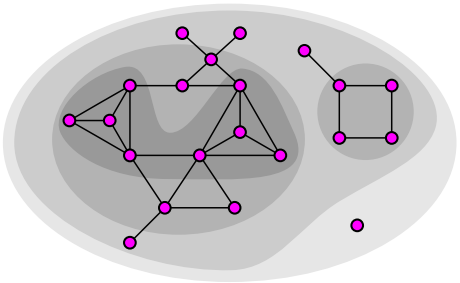

Figure 9: Cores

of arcs that are proportional to the number of different source-sink paths passing through the arc. In Figure 8 the main component of the arc cut at level 0.007 for SPC (search path count) weights of the SOM (selforganizing maps) citation network (4470 vertices, 12731 arcs) is presented.

## VII. Dense groups – cores and short rings

Several notions were proposed in attempts to formally describe dense groups in graphs.

*Clique* of order $k$, $k \geq 3$, is a maximal complete subgraph (isomorphic to complete graph $K_k$ – graph with $k$ vertices and all possible edges among them).

Other notions are: $s$-plexes, $s$-clans, LS sets, lambda sets, cores, ... (Wasserman and Faust [53]). For all of them, except for cores, it turned out that they are difficult (no fast algorithm exists) to detemine.

The notion of core was introduced by Seidman in 1983 [31]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. A subgraph $\mathcal{H}_k = (\mathcal{W}, \mathcal{E}|\mathcal{W})$ induced by the set $\mathcal{W}$ is a *$k$-core* or a *core of order* $k$ iff for all $v \in \mathcal{W} : \deg_{\mathcal{H}_k}(v) \geq k$, and $\mathcal{H}_k$ is a maximal subgraph with this property. The core of maximum order is also called the *main* core. The *core number* of vertex $v$ is the highest order of a core that contains this vertex. In general graphs instead of the degree $\deg(v)$ we can also use: in-degree, out-degree, in-degree + out-degree, etc., determining different types of cores.

From the Figure 9, representing 0, 1, 2 and 3 core, we can see the following properties of cores:

- The cores are nested: $i < j \implies \mathcal{H}_j \subseteq \mathcal{H}_i$. They form a hierarchy.
- Cores are not necessarily connected subgraphs.

An efficient algorithm for determining the cores hierarchy is based on the following property: If from a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we recursively delete all vertices,

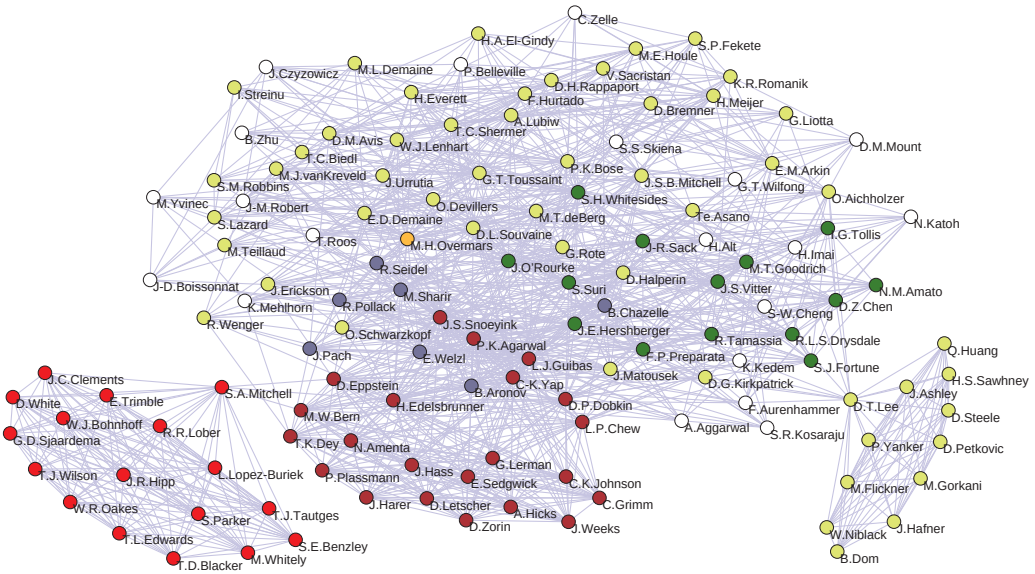


Figure 10: Cores of orders 10–21 in Computational Geometry collaboration network

and edges incident with them, of degree less than $k$, the remaining graph is the $k$-core.

The Figure 10 presents the cores of orders 10 to 21 in the collaboration network ($n = 7343$, $m = 11898$) for the field of Computational geometry – two authors are linked iff they wrote a paper together. The weight of the edge equals to the number of joint papers.

The notion of core can be generalized to networks. Let $\mathcal{N} = (\mathcal{V}, \mathcal{E}, w)$ be a network, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph and weight $w : \mathcal{E} \to \mathbb{R}$ is a function assigning values to edges. A *vertex property function* on $\mathcal{N}$, or a *p-function* for short, is a function $p(v, U)$, $v \in \mathcal{V}$, $U \subseteq \mathcal{V}$ with real values. Let $N_U(v) = N(v) \cap U$, where $N(v)$ is the set of neighbors of $v$. Besides degrees, here are some other examples of $p$-functions [12]:

$$
\begin{aligned}
p_S(v,U) &= \sum_{u \in N_U(v)} w(v,u), \text{ where } w : \mathcal{E} \to \mathbb{R}_0^+ \\
p_M(v,U) &= \max_{u \in N_U(v)} w(v,u), \text{ where } w : \mathcal{E} \to \mathbb{R} \\
p_k(v,U) &= \text{number of cycles of length } k \text{ through vertex } v \text{ in } (U, \mathcal{E}|U) \\
p_\gamma(v,U) &= \frac{\deg(v,U)}{\max_{u \in N(v)} \deg(u)}, \text{ if } \deg(v) > 0;\ 0, \text{ otherwise} \\
p_\delta(v,U) &= \max_{u \in N_U^+(v)} \deg(u) - \min_{u \in N_U^+(v)} \deg(u)
\end{aligned}
$$

$$p_a(v,U) \;=\; \frac{1}{|N_U(v)|}\sum_{u\in N_U(v)} w(v,u), \text{ if } N_U(v)\neq\emptyset;\ 0, \text{ otherwise}$$

The subgraph $\mathcal{H}=(C,\mathcal{E}|C)$ induced by the set $C\subseteq\mathcal{V}$ is a *$p$-core at level* $t\in\mathbb{R}$ iff for all $v\in C: t\leq p(v,C)$ and $C$ is a maximal such set.

The function $p$ is *monotone* iff it has the property

$$C_1\subset C_2 \Rightarrow \forall v\in\mathcal{V}: (p(v,C_1)\leq p(v,C_2))$$

The degrees and the functions $p_S$, $p_M$, $p_k$, $p_\gamma$ and $p_\delta$ are monotone; and $p_a$ is not. For a monotone function the $p$-core at level $t$ can be determined, as in the ordinary case, by successively deleting vertices with value of $p$ lower than $t$; and the cores on different levels are nested

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

The $p$-function is *local* iff $p(v,U)=p(v,N_U(v))$ . The degrees, $p_S$, $p_M$, $p_\gamma$, $p_\delta$ and $p_a$ are local; but $p_k$ is **not** local for $k\geq 4$. For a local monotone $p$-function an $O(m\max(\Delta,\log n))$ algorithm for determining the $p$-core levels exists, assuming that $p(v,N_C(v))$ can be computed in $O(\deg_C(v))$.

Figure 11 presents the $p_S$-core at level 46 of the collaboration network in the field of computational geometry. Note, for example, that R. Klein (lower left) has in-core degree only 2, but its in-core sum of weights is at least 46 – he wrote most of his papers with C. Icking.

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

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

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 $w_3$.

Related to triangular (3-rings) network is the notion of *triangular connectivity* that can be used to operationalize the notion of Granovetter's strong and weak ties [21]. This notion can be generalized to short cycle connectivity. For details see Batagelj and Zaveršnik [13]. For efficient algorithms for computing triangles in networks see Batagelj and Mrvar [9], Schank and Wagner [32], and Latapy [58].

In Figure 12 the edge-cut at level 16 of triangular network of Erdős collaboration graph (without Erdős, $n=6926$, $m=11343$) is presented (Batagelj and Mrvar [8]).

In directed networks there are two types of triangles or 3-rings (cyclic and transitive, see Figure 14).

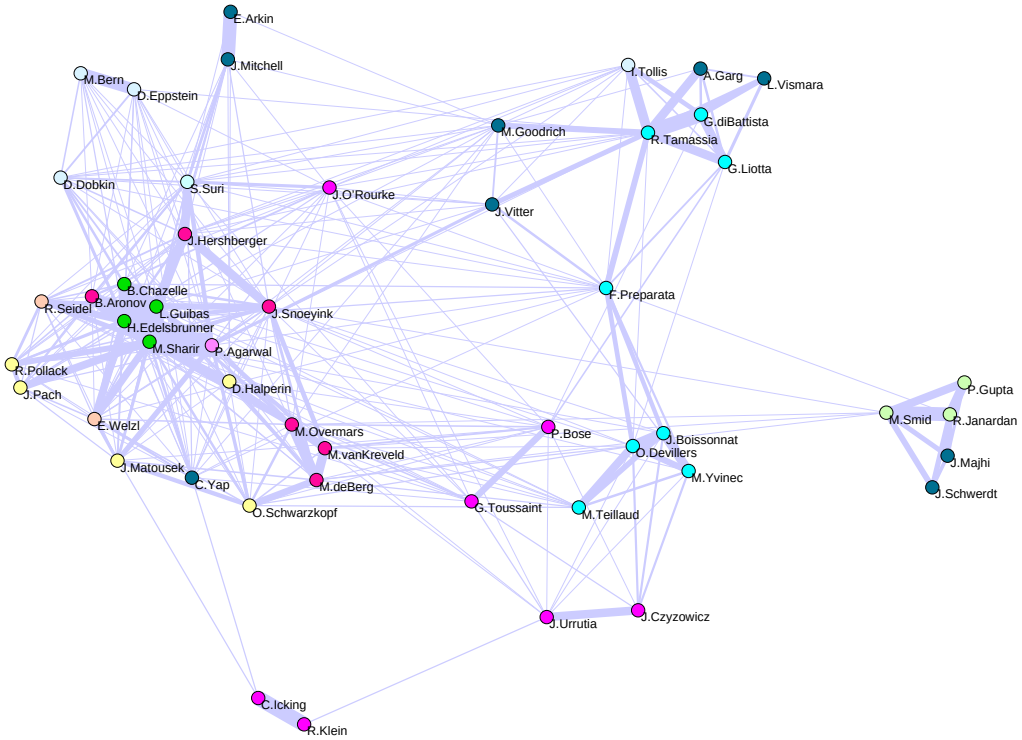


Figure 11: $p_S$-core at level 46 in Computational Geometry collaboration network

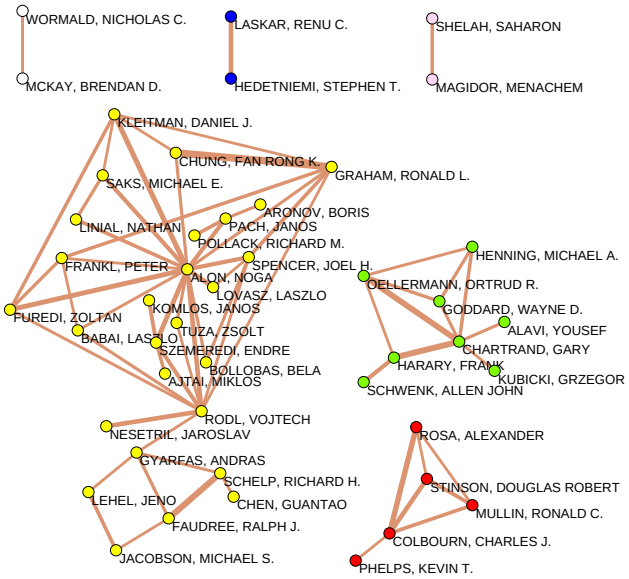


Figure 12: Edge-cut at level 16 of triangular network of Erdős collaboration graph

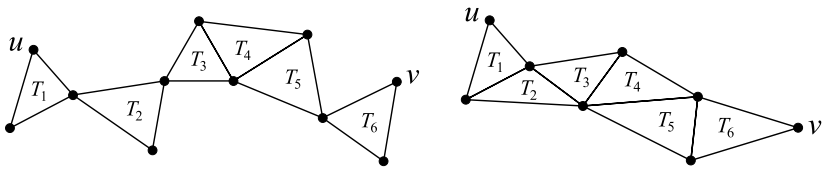


Figure 13: Vertex and edge triangular connectivity

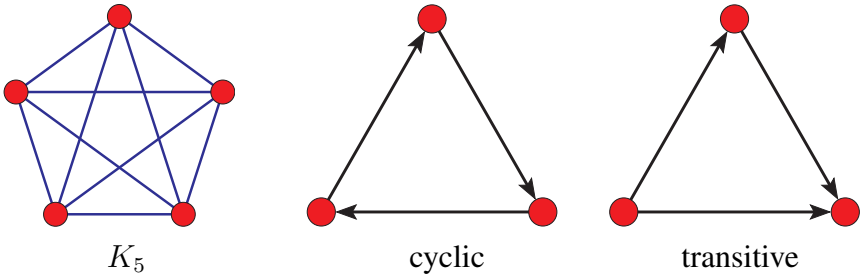


Figure 14: $K_5$ and cyclic and transitive 3-ring

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a simple undirected graph. *Clustering* in vertex $v$ is usually measured as a quotient between the number of lines in subgraph $\mathcal{G}^1(v) = \mathcal{G}(N(v))$ induced by the neighbors of vertex $v$ and the number of lines in the complete graph on these vertices:

$$C(v) = \begin{cases} \dfrac{2|\mathcal{L}(\mathcal{G}^1(v))|}{\deg(v)(\deg(v)-1)} & \deg(v) > 1 \\ 0 & \text{otherwise} \end{cases}$$

For simple directed graphs we have to omit the number 2.

So defined clustering coefficient attains largest values mostly on vertices of low degree – it is not useful for data analysis task. A better coefficient is obtained by the following correction

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

where $\Delta$ is the maximum degree in graph $\mathcal{G}$. This measure attains its largest value in vertices that belong to an isolated clique of size $\Delta$.

## VIII. Islands

Islands are very general and efficient approach to determine the 'important' sub-networks in a given network with respect to a given property of vertices or lines.

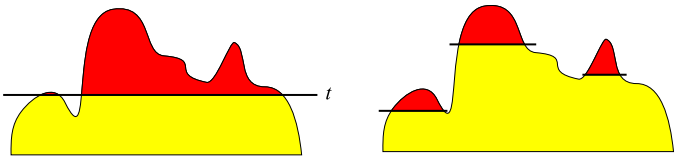


Figure 15: Cuts and islands

It is an improvement of the cuts approach. 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 [37].

In the islands approach we select only maximal islands of sizes inside the given size bounds $k$ to $K$, but on different levels. In this way we bypass the problems of the cuts approach: determining the 'right' threshold value and too small/large sizes of obtained components. Besides this we can also identify locally important islands with small heights – emerging groups. Very efficient algorithms exist to determine the islands hierarchy and to list all the islands of selected sizes. An island is *simple* iff it has only one peak.

As an example, let us take the Nber network of US Patents [59]. It has 3774768 vertices and 16522438 arcs. 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. The main island turns out to be the island on the theme *LCD – Liquid crystal display*.

In Figure 16 four islands for transitivity triangular weight from *The Edinburgh Associative Thesaurus* network [55] ($n = 23219$, $m = 325624$) are presented. From the left bottom island of words around the leader 'WORK' we see that the data were collected asking students.

## IX. 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 large organic molecula (carbon rings) and searching for relinking marriages in genealogies (Batagelj and Mrvar [11]; Batagelj [5]).

The Figure 17 presents three connected relinking marriages in the genealogy (represented as a p-graph) of ragusan noble families. In a p-graph the vertices represent married couples or nonmarried individuals. A solid arc indicates the _ *is a son of* _ relation, and a dotted arc indicates the _ *is a daughter of* _ relation.

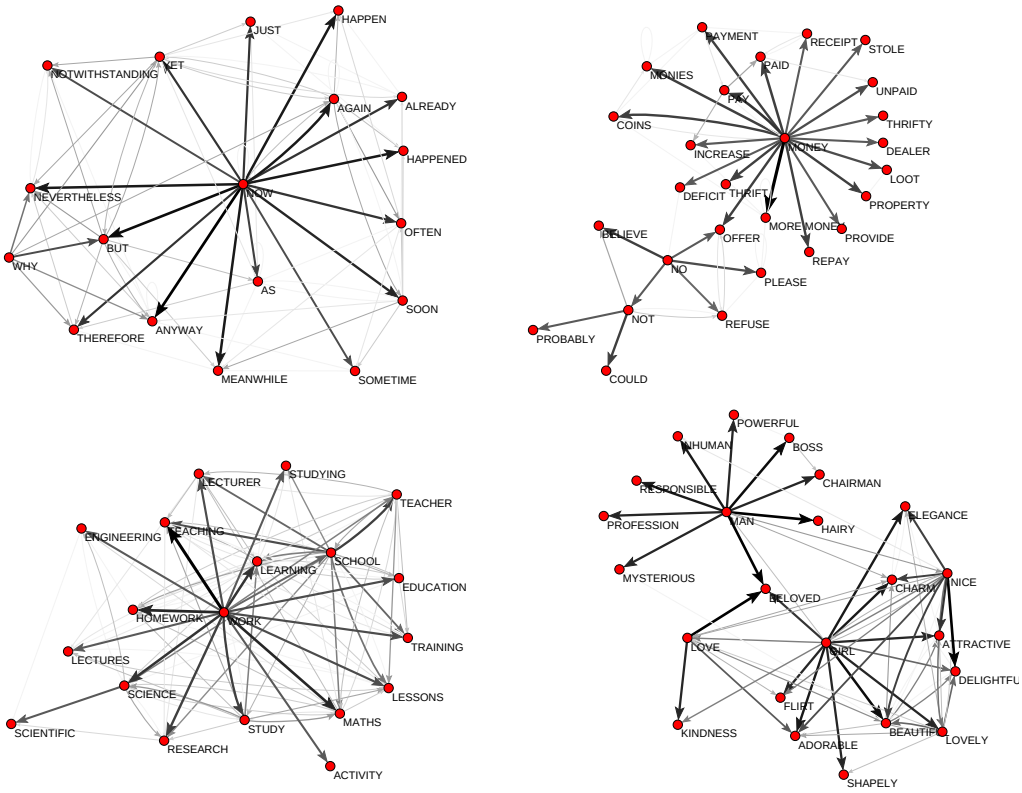


Figure 16: Selected Islands from The Edinburgh Associative Thesaurus

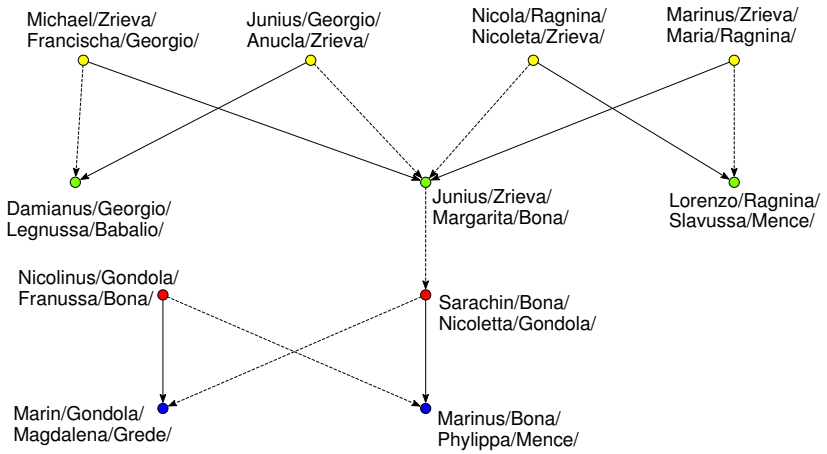


Figure 17: Marriages among relatives in Ragusa

In all three patterns a brother and a sister from one family found their partners in the same other family.

## X. Two Mode Networks

A network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$ in which the set of vertices $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ is composed of two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$, and $\mathcal{L}$ is a set of *lines* linking $\mathcal{V}_1$ and $\mathcal{V}_2$ is called a *two-mode* or bipartite network.

The two-mode networks often appear in applications, but till recently no directed methods for analysis of larger two-mode networks were available. To identify dense parts of two-mode network we can use the adapted cores and short rings approaches (Ahmed et al. [2]).

The subset of vertices $C \subseteq \mathcal{V}$ is a *$(p, q)$-core* in a two-mode network $\mathcal{N} = (\mathcal{V}_1, \mathcal{V}_2; \mathcal{L})$, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ iff

**a**. in the induced subnetwork $\mathcal{H} = (C_1, C_2; \mathcal{L}(C))$, $C_1 = C \cap \mathcal{V}_1$, $C_2 = C \cap \mathcal{V}_2$ it holds for all $v \in C_1 : \deg_{\mathcal{H}}(v) \geq p$ and for all $v \in C_2 : \deg_{\mathcal{H}}(v) \geq q$ ;

**b**. $C$ is the maximal subset of $\mathcal{V}$ satisfying condition **a**.

The two-mode cores have the following properties:

- $C(0, 0) = \mathcal{V}$
- $C(p, q)$ is not always connected
- $(p_1 \leq p_2) \wedge (q_1 \leq q_2) \Rightarrow C(p_1, q_1) \subseteq C(p_2, q_2)$

To determine a $(p, q)$-core an algorithm similar to the ordinary core algorithm can be used: recursively remove from the first set all vertices of degree less than $p$, and from the second set all vertices of degree less than $q$. It can be implemented to run in $O(m)$ time.

The main question when applying the bipartite cores is what are the right values of $p$ and $q$? The most interesting are the values on the 'border' that don't produce too large cores.

In Figure 18 the (247,2)-core and (27,22)-core from the Internet Movie Database [56] (two-mode network actors $\times$ movies, $n = 1324748 = 428440 + 896308$ vertices and $m = 3792390$ arcs) are presented. Both deal with wrestling.

In 2-mode network there are no 3-rings. The densest substructures are complete bipartite subgraphs $K_{p,q}$ – see $K_{4,5}$ on the left side of Figure 19. They contain many 4-rings

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

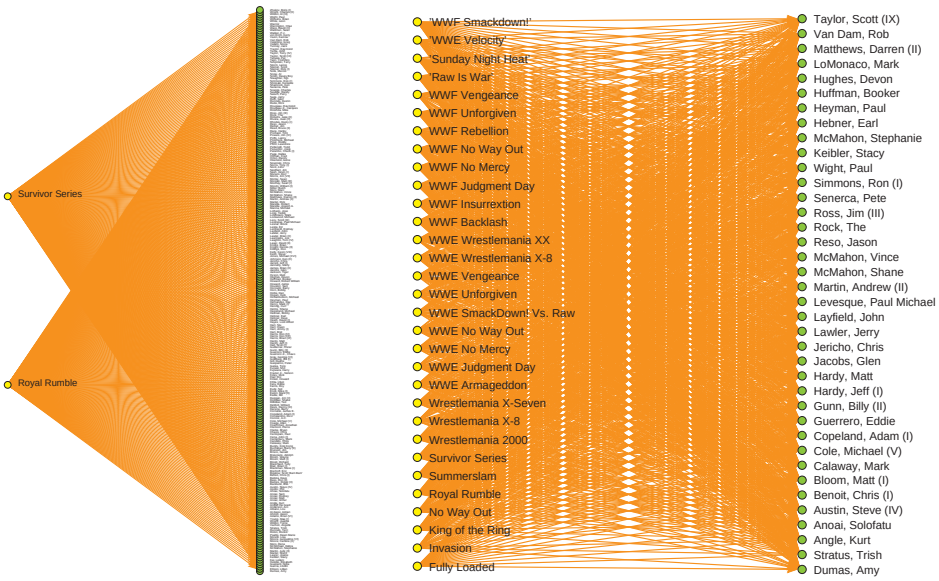


Figure 18: (247,2)-core and (27,22)-core of IMDB – wrestling

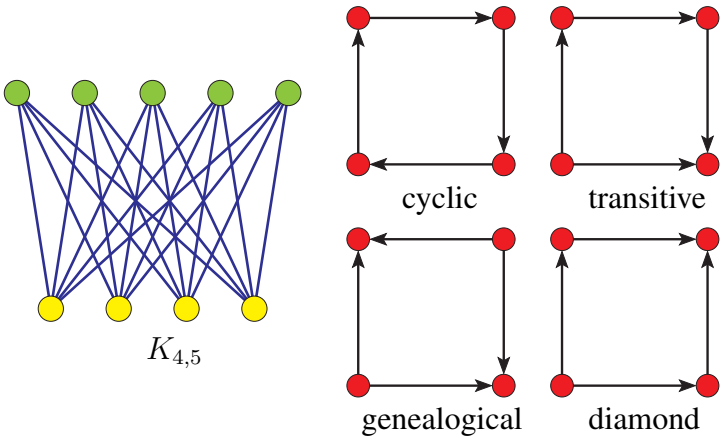


Figure 19: $K_{4,5}$ and directed 4-rings

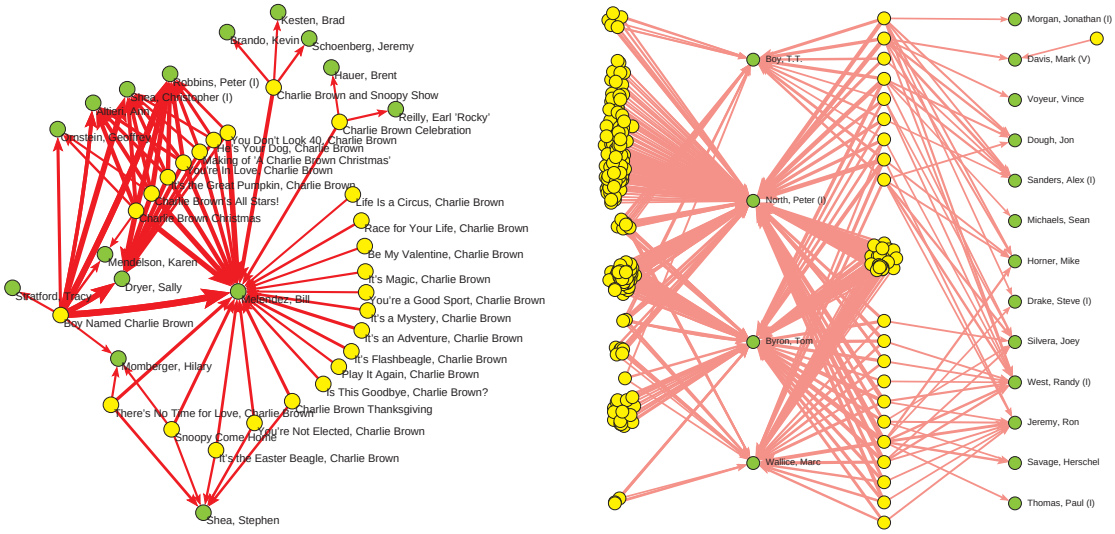


Figure 20: Islands for $w_4$ / Charlie Brown and Adult

There are 4 types of directed 4-rings – see the right side of Figure 19. In the case of transitive rings we can count also on how many transitive rings the arc is a *shortcut*.

In the Internet Movie Database we obtained for $w_4$ 12465 simple line islands on 56086 vertices; 30 among them have size at least 50. Two of them are presented on Figure 20.

## XI. Multiplication of networks

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

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

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

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

The standard matrix multiplication is too slow to be used for large networks. For sparse large networks we can multiply much faster considering only nonzero elements. In general the multiplication of large sparse networks is a 'dangerous' operation since the result can 'explode' – it is not sparse. But in many interesting cases we can assure that also the product is sparse. For example, 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.

A more detailed analysis gives: Let $d_{min}(k) = \min(\deg_A(k), \deg_B(k))$, $\Delta_{min} = \max_{k\in\mathcal{K}} d_{min}(k)$, $d_{max}(k) = \max(\deg_A(k), \deg_B(k))$, $\mathcal{K}(d) = \{k \in \mathcal{K} : d_{max}(k) \geq d\}$, and $d^* = \text{argmin}_d(|\mathcal{K}(d)| \leq d)$. If for the sparse networks $\mathcal{N}_A$ and $\mathcal{N}_B$ the quantities $\Delta_{min}$ and $d^*$ are small then also the resulting product network $\mathcal{N}_C$ is sparse.

For example, using network multiplication we can in a given genealogy from the basic relations (P – parent-of, L – is a man, J – is a woman) compute all other kinship relations. For details see Batagelj and Mrvar [11].

An important application of network multiplication is conversion of two-mode network to the corresponding one-mode networks. Often we transform a two-mode network $\mathcal{N}$ into an ordinary (one-mode) network $\mathcal{N}_1 = (\mathcal{I}, \mathcal{E}_1, w_1)$ or/and $\mathcal{N}_2 = (\mathcal{J}, \mathcal{E}_2, w_2)$, where $\mathcal{E}_1$ and $w_1$ are determined by the matrix $\mathbf{W}^{(1)} = \mathbf{W}\mathbf{W}^T$, $w^{(1)}_{ij} = \sum_{k\in\mathcal{J}} w_{ik} \cdot w^T_{kj}$ and $\mathbf{W}^T$ is the transpose of matrix $\mathbf{W}$. Evidently the matrix $\mathbf{W}^{(1)}$ is symmetric $w^{(1)}_{ij} = w^{(1)}_{ji}$. There is an edge $\{i, j\} \in \mathcal{E}_1$ in $\mathcal{N}_1$ iff $N(i) \cap N(j) \neq \emptyset$. Its weight is $w_1(i, j) = w^{(1)}_{ij}$. The network $\mathcal{N}_2$ is determined in a similar way by the matrix $\mathbf{W}^{(2)} = \mathbf{W}^T\mathbf{W}$.

The networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are analyzed using standard methods for one-mode networks.

Another very important application of network multiplication is producing different networks from data tables. A *data table* $\mathcal{T}$ is a set of *records* $\mathcal{T} = \{T_k : k \in \mathcal{K}\}$, where $\mathcal{K}$ 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 $k$.

Suppose that the property $\mathbf{q}$ has the range $2^{\mathcal{Q}}$. For example: Authors[WasFau] = { S. Wasserman, K. Faust }, PubYear[WasFau] = { 1994 }, ... If $\mathcal{Q}$ is finite (it can always be transformed in such set by partitioning the set $\mathcal{Q}$ and recoding the values) we can assign to the property $\mathbf{q}$ a two-mode network $\mathcal{K} \times \mathbf{q} = (\mathcal{K}, \mathcal{Q}, \mathcal{E}, w)$ where $(k, v) \in \mathcal{E}$ iff $v \in q(k)$, and $w(k, v) = 1$.

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

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

We can join a pair of properties $\mathbf{q}_i$ and $\mathbf{q}_j$ also with respect to the third property

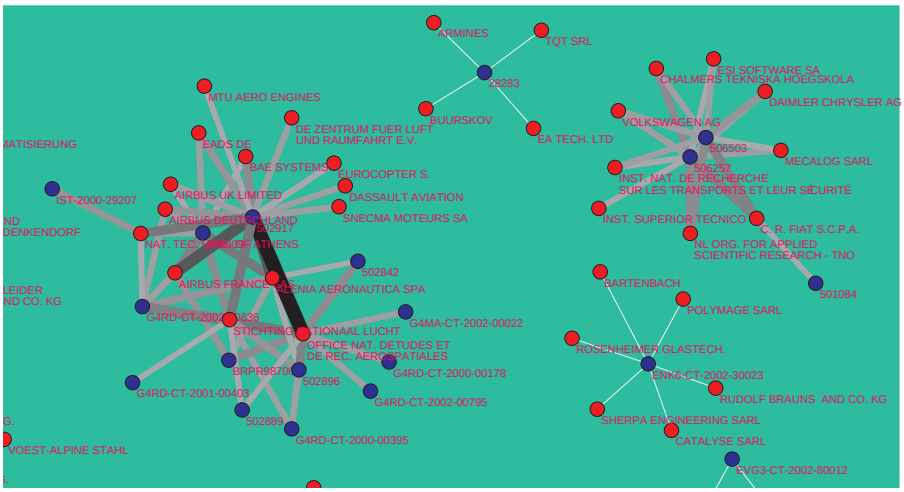


Figure 21: The main two islands in ProjInst

$\mathbf{q}_s$: we get a two-mode network $[\mathbf{q}_i \times \mathbf{q}_j]/\mathbf{q}_s = [\mathbf{q}_i \times \mathbf{q}_s] \star [\mathbf{q}_s \times \mathbf{q}_j]$.

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. The rows are the descriptions of projects participants (idents) and columns correspond to different their properties. From this table three two-mode networks were produced: Project – $\mathbf{P} = [$ idents $\times$ projects $]$; Country – $\mathbf{C} = [$ idents $\times$ countries $]$; and Institution – $\mathbf{U} = [$ idents $\times$ institutions $]$; where $|\text{idents}| = 8869$, $|\text{projects}| = 933$, $|\text{institutions}| = 3438$, and $|\text{countries}| = 60$.

Since all three networks have the common set (idents) we can derive from them using network multiplication several interesting networks, such as: ProjInst – $\mathbf{W} = [$ projects $\times$ institutions $] = \mathbf{P}^T \star \mathbf{U}$; Countries – $\mathbf{S} = [$ countries $\times$ countries $] = \mathbf{C}^T \star \mathbf{C}$; and Institutions – $\mathbf{Q} = [$ institutions $\times$ institutions $]/$ projects $= \mathbf{W}^T \star \mathbf{W}$.

For identifying important parts of ProjInst network the 4-rings weights were computed and in the obtained network the line islands were determined. 101 islands were obtained, 18 of the size at least 5. In Figure 21 the two most important islands are presented: aviation companies and car companies.

In Figure 22 the collaboration among countries is presented. For dense (sub)-networks we get better visualization by using matrix display. To determine the ordering of vertices we used Ward's clustering procedure with corrected Euclidean distance as dissimilarity measure (Doreian et al. [46]). The permutation determined by hierarchy can often be improved by changing the positions of clusters in the clustering tree. We get a typical center-periphery structure (See Figure 23).

Note that in matrix display some details become appearent, such as the col-

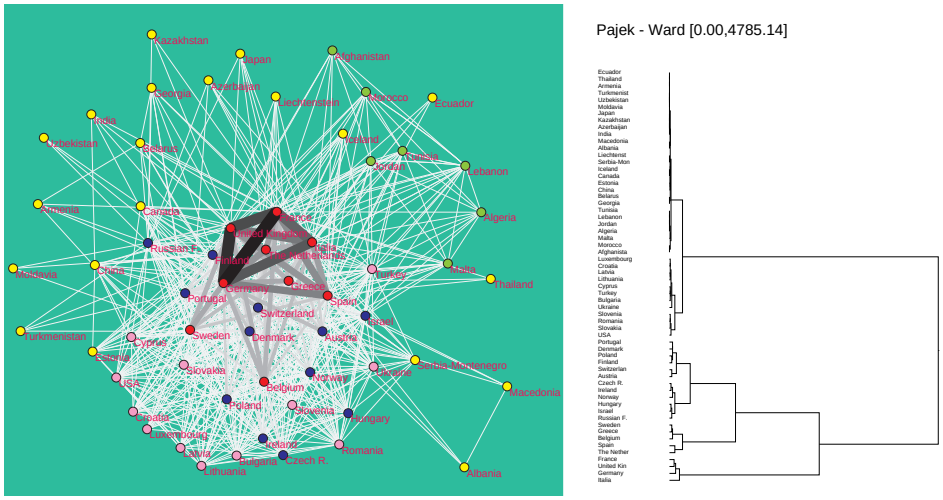


Figure 22: Collaboration among countries

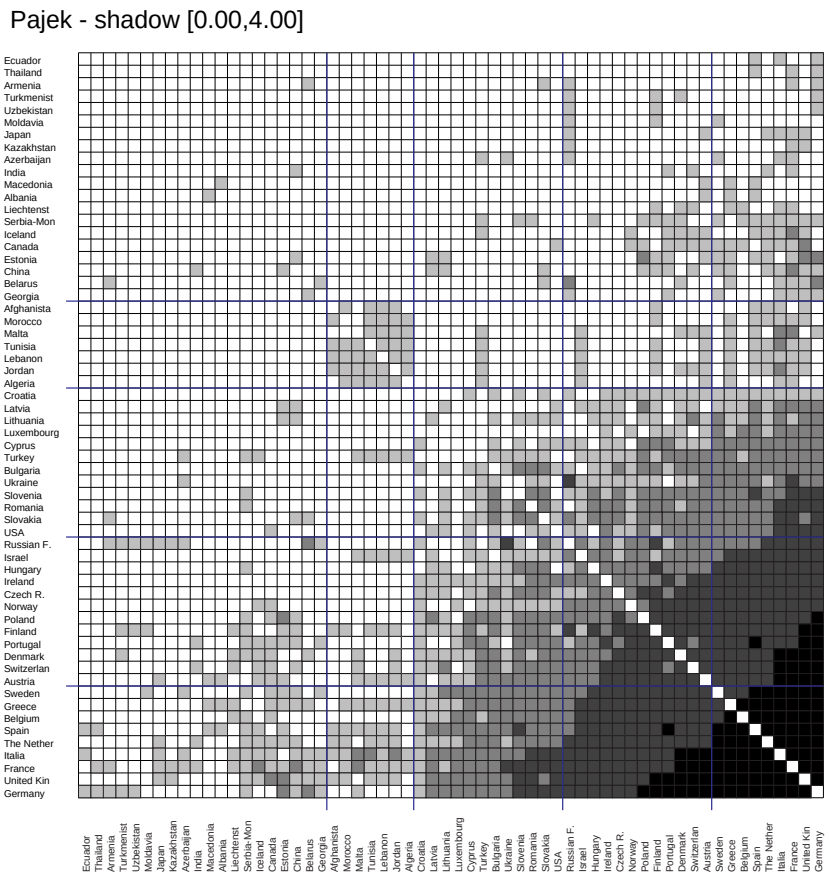


Figure 23: Matrix display of collaboration among countries

laboration inside the peripherical group Afghanistan, Moroco, Malta, Tunisia, Lebanon, Jordan and Algeria; or collaboration of Russian Federation with ex-Soviet republics Turkmenistan, Uzbekistan, Moldavia, Kazakhstan, Azerbaijan and Japan.

## XII. Statistical approach

There are many properties *computed* from the network data that give us different information about it. For example:
*global properties*: number of vertices, lines (edges/arcs), components; diameter; centralization; maximum core number, ...
*local properties*: degrees, core numbers, indices (betweeness, hubs, authorities, ...). Usually we look at their *distributions* or *inspect* the values of interesting elements.

Another interesting task is searching for associations between computed (structural) data and input (measured) data.

Paul Erdős and Alfréd Rényi introduced in 1959 the notion of random graph in which each pair of vertices is linked with a given probability $p$. The theory of ER random graphs is well developed (see Bollobás [42]). Some characteristic results:

- the degree distribution is binomial (in the limit Poisson's) and most of the vertices have degree (very) close to the average degree;
- for $p \geq \frac{1}{n}$ cycles appear in the graph, and soon also the *giant component*;
- for $p \geq \frac{\log_2 n}{n}$ almost all graphs are connected;

Real-life networks are usually not random in the Erdős–Rényi sense. The analysis of their distributions gave new views about their structure.

On the left side of Figure 24 a degree distribution in ER graph on $n = 100000$ vertices with average degree $\overline{\deg} = 30$ is presented. On the right side a degree distribution for US Patents citation network is presented (in log-log scale). Evidently this distribution is very far away from Poisson distribution.

In 1967 a psychologist Stanley Milgram made his experiment with letters. The letter should reach a target person. The persons involved in experiment were asked to send the letter with these instructions to his or her acquaintance that is supposed to be closer (in the acquaintances network) to the target person. The letter was sent from Boston to Omaha. The average length of the successful paths was 6 – *six degrees of separation*. The average path length on the internet is 19 clicks.

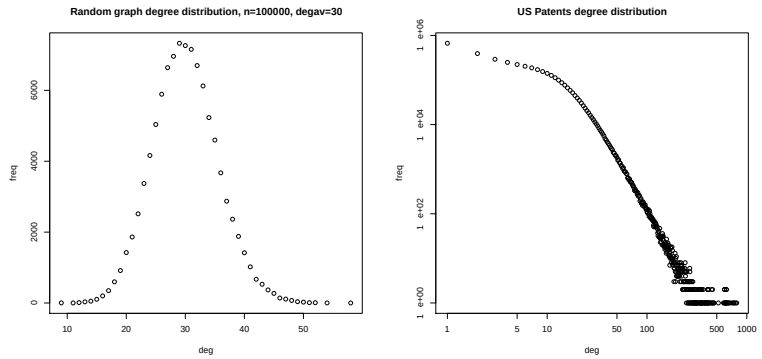


Figure 24: Distributions: ER-random and US patents

The networks in which the average shortest path length is small are called *small worlds*. Duncan Watts and Steven Strogatz developed in late 90-ties a procedure for construction of (random) small worlds by *rewiring* – an edge is randomly selected and one of its endpoints is attached to same other vertex. After each rewiring step the average length of geodesics is usually descreased because the rewiring creates shortcuts.

Albert-László Barabási from University of Notre Dame in 1998 analyzed several networks and noticed:

- the degree distribution follows the *power law* – the probability $p_d$ that a vertex has a degree $d$ equals to $p_d = cd^{-\gamma}$ . In a log-log scale diagram it is represented by a line;
- in a network there exist some vertices with *large degree* (very unprobable in ER graphs). These vertices link the network into a single component.

It turned out that most of real life networks (persons – e-mail, phone calls, sexual contacts (drug users, AIDS), collaboration; movie actors – playing in the same movie; proteins – interactions; words – semantic relations; ...) have such characteristics. Because for these networks their degree distribution has no natural scale they were named *scale-free* networks. For a discussion about the notion of scale-free network see Li et al. [27].

The first explanation (Barabási) of scale-free nature of many real-life networks was:

- these networks are growing;
- in this process new vertices are added and linked with new edges to already existing vertices. The random selection of vertex to which a new vertex is

attached is not uniform but follows the *preferential attachment* rule – the selection probability is proportional to the degree of a vertex.

Based on this model it can be shown that:

- the degree distribution is the power law;
- the average length of geodesics is $O(\log n)$ ;
- these networks are resilient against random vertex or edge removals (random attacks), but quickly become disconnected when large degree nodes (Achilles' heel) are removed (targeted attacks).

Mark Granovetter noticed in 1973 that in social networks groups appear linked with *strong ties* [21]. They link in larger networks with *weak ties*. Also in other real-life networks vertices often form groups – the clustering coefficient is larger than in ER networks.

Several improvements and alternative models were proposed that also produce scale-free networks with some additional properties characteristic for real-life networks: copying (Kleinberg [25]), combining random and preferential attachment (Pennock et al. [30]), R-mat (Chakrabarti et al. [16]), forest fire (Leskovec et al. [26]), aging, fitness, nonlinear preferences, ...

There are several applications of the scale-free networks theory. For example searching (Adamic et al. [1]) and spreading of epidemies (Barthélemy, Barrat, Pastor-Sattoras, Vespignani, Complex Networks Collaboratory [54]).

For general overviews see Albert, Barabási [3], Newman [29], Dorogovtsev and Mendes [47], and Newman at al. [51].

## XIII. Future Directions

In 2005 the support for *multi-relational* networks was introduced in **`Pajek`**. Combined with *temporal* networks it enables analysis of new kinds of networks – such as KEDS networks [57] (*Kansas Event Data System* or *Tabari*). These networks are usually small in terms of vertices but can be (very) large in terms of lines – different interaction events among actors.

The last developed approach for analysis of large networks is adaptation of hierarchical clustering with relational constraints based on Ferligoj and Batagelj [18] to large networks. The basic idea to get a fast algorithm is to compute the dissimilarities between units (vertices) only for the linked pairs of units (Batagelj and Mrvar [10]). This approach is one of the possible approaches to analysis of spatial networks [63].

There are still several fields of social network analysis for which efficient approaches to deal with large networks have to be developed such as blockmodeling, probabilistic models, ...

In the near future new versions of network analysis software will appear using very large computer memories enabled by the new 64-bit computer architecture. A special challenge is development of methods and software for analysis of huge networks.

# Bibliography

## Primary Literature


[1] Adamic L.A., Lukose R.M., Huberman B.A.(2002) *Local Search in Unstructured Networks*. in Handbook of Graphs and Networks: From the Genome to the Internet, S. Bornholdt, and H.G. Schuster (eds.), Wiley-VCH, Berlin.

[2] Ahmed A., Batagelj V., Fu X., Hong S-H., Merrick D., Mrvar A. (2007) *Visualisation and Analysis of the Internet Movie Database*. Asia-Pacific Symposium on Visualisation (APVIS2007): Sydney, NSW, Australia, February 5-7, 2007 : proceedings. New York: IEEE, p. 17-24.

[3] Albert R., Barabási A-L. (2002) *Statistical mechanics of complex networks*. Reviews of Modern Physics, Vol. 74, 47-97.

[4] Alvarez-Hamelin J.I., Dall'Asta L., Barrat A., Vespignani A. (2005) *k-core decomposition: a tool for the visualization of large scale networks*. cs.NI/0504107, 28 Apr 2005.

[5] Batagelj V. (1989) *Similarity measures between structured objects*. Proceedings of International Course and Conference on the Interfaces between Mathematics, Chemistry and Computer Science, Dubrovnik, 20-25 June 1988, A. Graovac (Ed.). Studies in Physical and Theoretical Chemistry, Vol. 63; Elsevier/Noth-Holland, Amsterdam, p. 25-40.

[6] Batagelj V., Brandes U. (2005) *Efficient Generation of Large Random Networks*. Physical Review E 71, 036113, 2005.

[7] Batagelj V., Ferligoj A. (2000) *Clustering relational data*. in Data Analysis (ed.: W. Gaul, O. Opitz, M. Schader), Springer, Berlin, 3-15.

[8] Batagelj V., Mrvar A. (2000) *Some Analyses of Erdős Collaboration Graph*. Social Networks **22**, 173–186.

[9] Batagelj V., Mrvar A. (2001) *A Subquadratic Triad Census Algorithm for Large Sparse Networks with Small Maximum Degree*. Social Networks **23**, 23743.

[10] Batagelj V., Mrvar A. (2007) *Hierarchical clustering with relational constraints of large data sets*. 6th Slovenian International Conference on Graph Theory, Bled, 24 – 30 June 2007.

[11] Batagelj V., Mrvar A. (2008) *Analysis of kinship relations with Pajek*. in press *Social Science Computer Review – SSCORE.*

[12] Batagelj V., Zaveršnik, M. (2002) *Generalized Cores*, arxiv cs.DS/0202039

[13] Batagelj V., Zaveršnik M. (2007) *Short cycle connectivity*. Discrete Mathematics 307(3-5), 310-318.

[14] Brandes U. (2001) *A Faster Algorithm for Betweenness Centrality*. Journal of Mathematical Sociology 25(2):163-177.

[15] Breiger R.L. (2004) *The analysis of social networks*.

[16] Chakrabarti D., Zhan Y., Faloutsos C. (2004) *R-MAT: A Recursive Model for Graph Mining*. in SIAM Data Mining 2004, Orlando, Florida, USA.

[17] Doreian P., Batagelj V., Ferligoj A. (2000) *Symmetric-Acyclic Decompositions of Networks*. Journal of Classification, 17(1), 3-28.

[18] Ferligoj A., Batagelj V. (1983) *Some types of clustering with relational constraints*. *Psychometrika*, **48**(4), 541–552.

[19] Freeman L.C. (1979) *Centrality in Social Networks: A Conceptual Clarification.* Social Networks 1: 211-213.

[20] Garfield E, Sher IH, and Torpie RJ. (1964) *The Use of Citation Data in Writing the History of Science*. Philadelphia: The Institute for Scientific Information, December 1964.

[21] Granovetter M. (1973) *The Strength of Weak Ties*. American Journal of Sociology 78: 1360-80.

[22] Huisman M., Van Duijn M.A.J. (2005) *Software for social network analysis*. In: P.J. Carrington, J. Scott, S. Wasserman, Models and methods in social network analysis (pp. 270-316). Cambridge: Cambridge University Press.

[23] Hummon N.P., Doreian P. (1990) *Computational Methods for Social Network Analysis*. Social Networks, **12**, 273-288.

[24] Kleinberg J. (1998) *Authoritative sources in a hyperlinked environment*. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.

[25] Kleinberg J., Kumar R., Raghavan P., Rajagopalan S., Tomkins A.(1999) *The Web as a graph: measurements, models and methods*. Proceedings of the 5th International Computing and combinatorics Conference.

[26] Leskovec J., Kleinberg J., Faloutsos C. (2006) *Laws of Graph Evolution: Densification and Shrinking Diameters*.

[27] Li L., Alderson D., Tanaka R., Doyle J.C., Willinger W. (2005) *Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications*. cond-mat/0501169.

[28] Mane K.K., Börner K. (2004) *Mapping topics and topic bursts in PNAS*. PNAS 101: 5287-5290.

[29] Newman M.E.J. (2003) *The structure and function of complex networks*. SIAM Review 45, 167-256.

[30] Pennock D.M., Flake G.W., Lawrence S., Glover E.J., Giles C.L. (2002) *Winners dont take all: Characterizing the competition for links on the web*. PNAS 99(8), 52075211.

[31] Seidman S.B. (1983) *Network Structure And Minimum Degree*. *Social Networks* **5**:269–287.

[32] Schank T., Wagner D. (2005) *Finding, counting and listing all triangles in large graphs, an experimental study*. In Workshop on Experimental and Efficient Algorithms (WEA), 606-609.

[33] Snyder D., Kick E. (1979) *The World System and World Trade: An Empirical Exploration of Conceptual Conflicts*. Sociological Quaterly, 20,1, 23-36.

[34] Snijders T.A.B. (2005) *Models for Longitudinal Network Data*. Chapter 11 in P. Carrington, J. Scott, and S. Wasserman (Eds.), Models and methods in social network analysis. New York: Cambridge University Press.

[35] Stuckenschmidt H., Klein M. (2004) *Structure-Based Partitioning of Large Concept Hierarchies*. Proceedings of the 3rd International Semantic Web Conference ISWC 2004, Hiroshima, Japan.

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

[37] Zaveršnik M., Batagelj V. (2004) *Islands*. Slides from *Sunbelt XXIV, Portorož, Slovenia, 12.-16. May 2004*,

## Books and Reviews

[38] Abello J., Pardalos P.M., Resende M.G. (Eds.) (2002) *Handbook of Massive Data Sets*. Springer.

[39] Ahuja R.K., Magnanti T.L., Orlin J.B. (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

[40] Batagelj V., Mrvar A. (2003) *Pajek – Analysis and Visualization of Large Networks*. in Jünger, M., Mutzel, P., (Eds.) Graph Drawing Software. Springer, Berlin, p. 77-103.

[41] Brandes U., Erlebach T. (Eds.) (2005) *Network Analysis: Methodological Foundations*. LNCS, Springer, Berlin.

[42] Bollobás B. (2001) *Random Graphs*. Cambridge University Press.

[43] Carrington P.J., Scott J., Wasserman S. (Eds.) (2005) *Models and Methods in Social Network Analysis*. Cambridge University Press.

[44] Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. (2001) *Introduction to algorithms*. Cambridge (Mass.): MIT Press.

[45] Degenne A., Forsé M. (1999) *Introducing Social Networks*. SAGE Publications.

[46] Doreian P., Batagelj V., Ferligoj A. (2005) *Generalized Blockmodeling*, Cambridge University Press.

[47] Dorogovtsev S.N., Mendes J.F.F. (2003) *Evolution of Networks: From Biological Nets to the Internet and Www*. Oxford University Press.

[48] de Nooy W., Mrvar A., Batagelj V. (2005) *Exploratory Social Network Analysis with Pajek*, Cambridge University Press.

[49] Harary F., Norman R.Z., Cartwright D. (1965) *Structural Models: An Intoroduction to the Theory of Directed Graphs*. John Wiley.

[50] Knuth D.E. (1993) *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley.

[51] Newman M.E.J., Barabási A-L., Watts D. (2006) *The Structure and Dynamics of Networks*. Princeton Studies in Complexity.

[52] Scott J.P. (2000) *Social Network Analysis: A Handbook*. SAGE Publications.

[53] Wasserman S., Faust K. (1994) *Social Network Analysis: Methods and Applications*. Cambridge University Press.

## Web Resources

[54] Complex Networks Collaboratory: http://cxnets.googlepages.com/

[55] The Edinburgh Associative Thesaurus: http://www.eat.rl.ac.uk/

[56] Internet Movie Database http://www.imdb.com/

[57] The Kansas Event Data System: http://web.ku.edu/keds/

[58] Matthieu Latapy. Triangle computation web page. http://www-rp.lip6.fr/~latapy/Triangles/

[59] Nber: http://www.nber.org/patents/

[60] Center for Complex Network Research, Notre Dame: http://www.nd.edu/~networks/

[61] Netminer: http://www.netminer.com/

[62] **Pajek**: http://vlado.fmf.uni-lj.si/pub/networks/Pajek data sets: http://vlado.fmf.uni-lj.si/pub/networks/data/.

[63] Center for Spatially Integrated Social Science: http://www.csiss.org/

[64] UCINET: http://www.analytictech.com/