Photo: V. Batagelj, ***Net***

Islands

Vladimir Batagelj
Matjaž Zaveršnik

University of Ljubljana

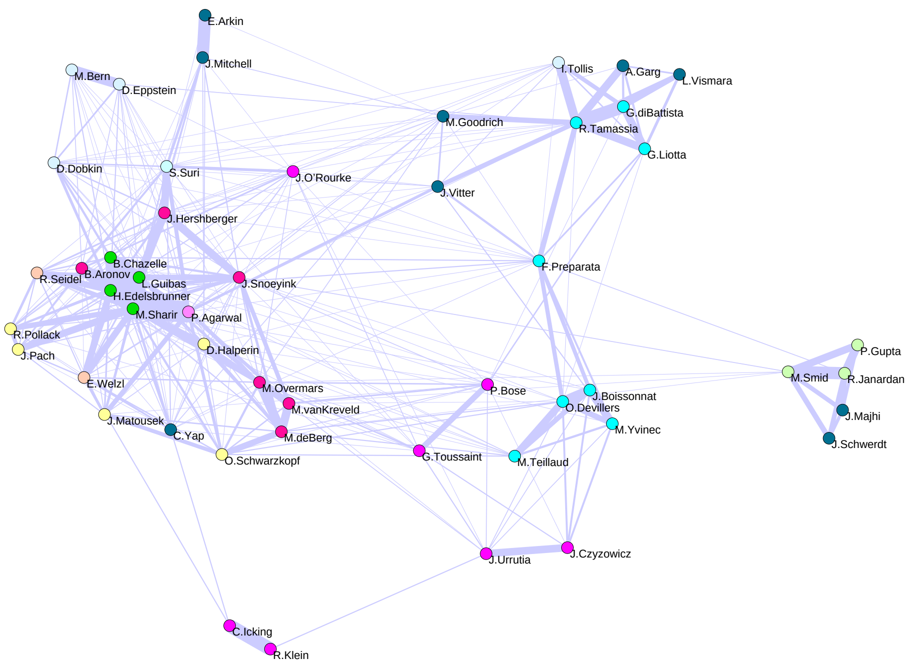
COSIN Meeting at the University of Karlsruhe

November 8th and 9th 2004

version: November 7, 2004 / 22 : 20

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 and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of lines (links, ties). Undirected lines $\mathcal{E}$ are called *edges*, and directed lines $\mathcal{A}$ are called *arcs*.
  $n = \text{card}(\mathcal{V})$, $m = \text{card}(\mathcal{L})$
- $\mathcal{P}$ *vertex value functions* / properties: $p : \mathcal{V} \rightarrow A$
- $\mathcal{W}$ *line value functions* / weights: $w : \mathcal{L} \rightarrow B$

Cuts

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

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

  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 $\mathcal{N} = (\mathcal{V}, \mathcal{L}, w)$, $w : \mathcal{L} \to \mathbb{R}$, at selected level t is a subnetwork $\mathcal{N}(t) = (\mathcal{V}(\mathcal{L}'), \mathcal{L}', w)$, determined by

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

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

- The line-cut at level t is a vertex-cut at the same level for

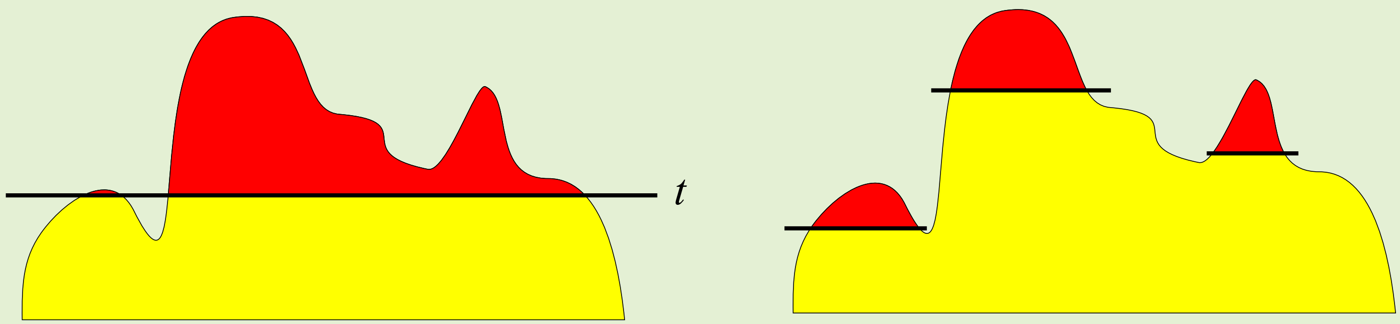
$$p(v) = \max_{u \in N(v)} w(v, u)$$

  where we preserve only lines with $w(e) \geq t$.

Simple analysis using cuts

- After making a cut at selected level t we look at the components of the $\mathcal{N}(t)$. Their number and sizes depend on t. Usually there are many small and some large 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/line values and the distribution of component sizes and considering additional knowledge about the nature of network or goals of analysis.

Cuts and islands



Vertex islands

- Nonempty subset of vertices $\mathcal{C} \subseteq \mathcal{V}$ is a *vertex island*, if
  - the corresponding induced subgraph $\mathcal{G}|\mathcal{C} = (\mathcal{C}, \mathcal{L}(\mathcal{C}))$ is connected, and
  - the values of the vertices in the neighborhood of $\mathcal{C}$ are less than or equal to the values of vertices from $\mathcal{C}$.

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

- Vertex island $\mathcal{C} \subseteq \mathcal{V}$ is a *regular vertex island*, if the stronger condition holds:

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

Some properties of vertex islands

- The sets of vertices of connected components of vertex-cut at selected level t are regular vertex islands.
- The set $\mathcal{H}_p(\mathcal{N})$ of all regular vertex islands of network $\mathcal{N}$ is a complete hierarchy:
  - two islands are disjoint or one of them is a subset of the other
  - each vertex belongs to at least one island
- Vertex islands are invariant for the strictly increasing transformations of the property p.
- Two linked vertices cannot belong to two disjoint regular vertex islands.

Algorithm for determining maximal regular vertex islands of limited size

- We sink the network into the water, then we lower the water level step by step.
- Each time a new vertex v appears from the water, we check with which of the already visible islands is connected.
- We join these islands and the vertex v obtaining a new (larger) island. These islands are *subislands* of the new island.
  Vertex v is a *port* of the new island (the vertex with the smallest value).
- This can be done in $\mathcal{O}(\max(n \log n, m))$ time.

algorithm: hierarchy of islands ...

$islands := \emptyset$
sort $\mathcal{V}$ in decreasing order according to p
for each $v \in \mathcal{V}$ (in the obtained order) **do begin**
    $island :=$ **new** $Island()$
    $island.port := v$
    $island.subislands := \{i \in islands : i \cap N(v) \neq \emptyset\}$
    $islands := islands \cup \{island\} \setminus island.subislands$
    for each $i \in island.subislands$ **do** $i.regular := p(i.port) > p(v)$
    determine the type of $island$
end
for each $i \in islands$ **do** $i.regular :=$ **true**

... algorithm: list of islands

$L := \emptyset$
while $islands \neq \emptyset$ **do begin**
    select $island \in islands$
    $islands := islands \setminus \{island\}$
    if $|island| < min$ **then delete** $island$
    else if $|island| > max \vee \neg island.regular$ **then begin**
        $islands := islands \cup island.subislands$
        delete $island$
    end
    else $L := L \cup \{island\}$
end

Simple vertex islands

- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is a *local vertex peak*, if it is a regular vertex island and all of its vertices have the same value.
- Vertex island with a single local vertex peak is called a *simple vertex island*.
- The types of vertex islands:
  - FLAT – all vertices have the same value
  - SINGLE – island has a single local vertex peak
  - MULTI – island has more than one local vertex peaks
- Only the islands of type FLAT or SINGLE are simple islands.

Determining the type of vertex island

if $|island.subislands| = 0$ **then** $island.type := \text{FLAT}$
else if $|island.subislands| = 1$ **then begin**
  select $i \in island.subislands$
  if $i.type \neq \text{FLAT}$ **then** $island.type := i.type$
  else if $p(i.port) = p(v)$ **then** $island.type := \text{FLAT}$
  else $island.type := \text{SINGLE}$
end
else begin
  for each $i \in island.subislands$ **do begin**
    $ok := i.type = \text{FLAT} \wedge p(i.port) = p(v)$
    if $\neg ok$ **then break**
  end
  if ok **then** $island.type := \text{FLAT}$
  else $island.type := \text{MULTI}$
end

Line islands

- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is a *line island*, if it is a singleton (degenerated island) or there exists a spanning tree $\mathcal{T}$ on $\mathcal{C}$ such that the values of lines with exactly one endpoint in $\mathcal{C}$ are less than or equal to the values of lines of the tree $\mathcal{T}$.

$$\max_{\substack{(u\,;\,v)\in\mathcal{L}:\\ u\in\mathcal{C}\wedge v\notin\mathcal{C}}} w((u\,;v)) \leq \min_{e\in\mathcal{L}(\mathcal{T})} w(e)$$

- Line island $\mathcal{C} \subseteq \mathcal{V}$ is a *regular line island*, if the stronger condition holds:

$$\max_{\substack{(u\,;\,v)\in\mathcal{L}:\\ u\in\mathcal{C}\wedge v\notin\mathcal{C}}} w((u\,;v)) < \min_{e\in\mathcal{L}(\mathcal{T})} w(e)$$

Some properties of line islands

- The sets of vertices of connected components of line-cut at selected level t are regular line islands.
- The set $\mathcal{H}_w(\mathcal{N})$ of all nondegenerated regular line islands of network $\mathcal{N}$ is hierarchy (not necessarily complete):
  - two islands are disjoint or one of them is a subset of the other
- Line islands are invariant for the strictly increasing transformations of the weight w.
- Two linked vertices may belong to two disjoint regular line islands.

Algorithm for determining maximal regular line islands of limited size

- We sink the network into the water, then we lower the water level step by step.
- Each time a new line e appears from the water, we check with which of the already visible islands is connected (there are exactly two such islands).
- We join these two islands obtaining a new (larger) island.
  These islands are *subislands* of the new island.
  Line e is a *port* of the new island (not necessarily the line with the smallest value).
- This can be done in $\mathcal{O}(m \log n)$ time.

algorithm: hierarchy of islands ...

$islands := \{\{v\} : v \in \mathcal{V}\}$
for each $i \in islands$ **do** $i.port :=$ **null**
sort $\mathcal{L}$ in decreasing order according to w
for each $e(u\,;v) \in \mathcal{L}$ (in the obtained order) **do begin**
    $i1 := island \in islands : u \in island$
    $i2 := island \in islands : v \in island$
    if $i1 \neq i2$ **then begin**
        $island :=$ **new** $Island()$
        $island.port := e$
        $island.subisland1 := i1$
        $island.subisland2 := i2$
        $islands := islands \cup \{island\} \setminus \{i1, i2\}$
        $i1.regular := i1.port =$ **null** $\vee\; w(i1.port) > w(e)$
        $i2.regular := i2.port =$ **null** $\vee\; w(i2.port) > w(e)$
    end
    determine the type of $island$
end
for each $i \in islands$ **do** $i.regular :=$ **true**

... algorithm: list of islands

$L := \emptyset$
while $islands \neq \emptyset$ **do begin**
  select $island \in subislands$
  $subislands := subislands \setminus \{island\}$
  if $|island| < min$ **then delete** $island$
  else if $|island| > max \vee \neg island.regular$ **then begin**
    $islands := islands \cup \{island.subisland1, island.subisland2\}$
    delete $island$
  end
  else $L := L \cup \{island\}$
end

Simple line islands

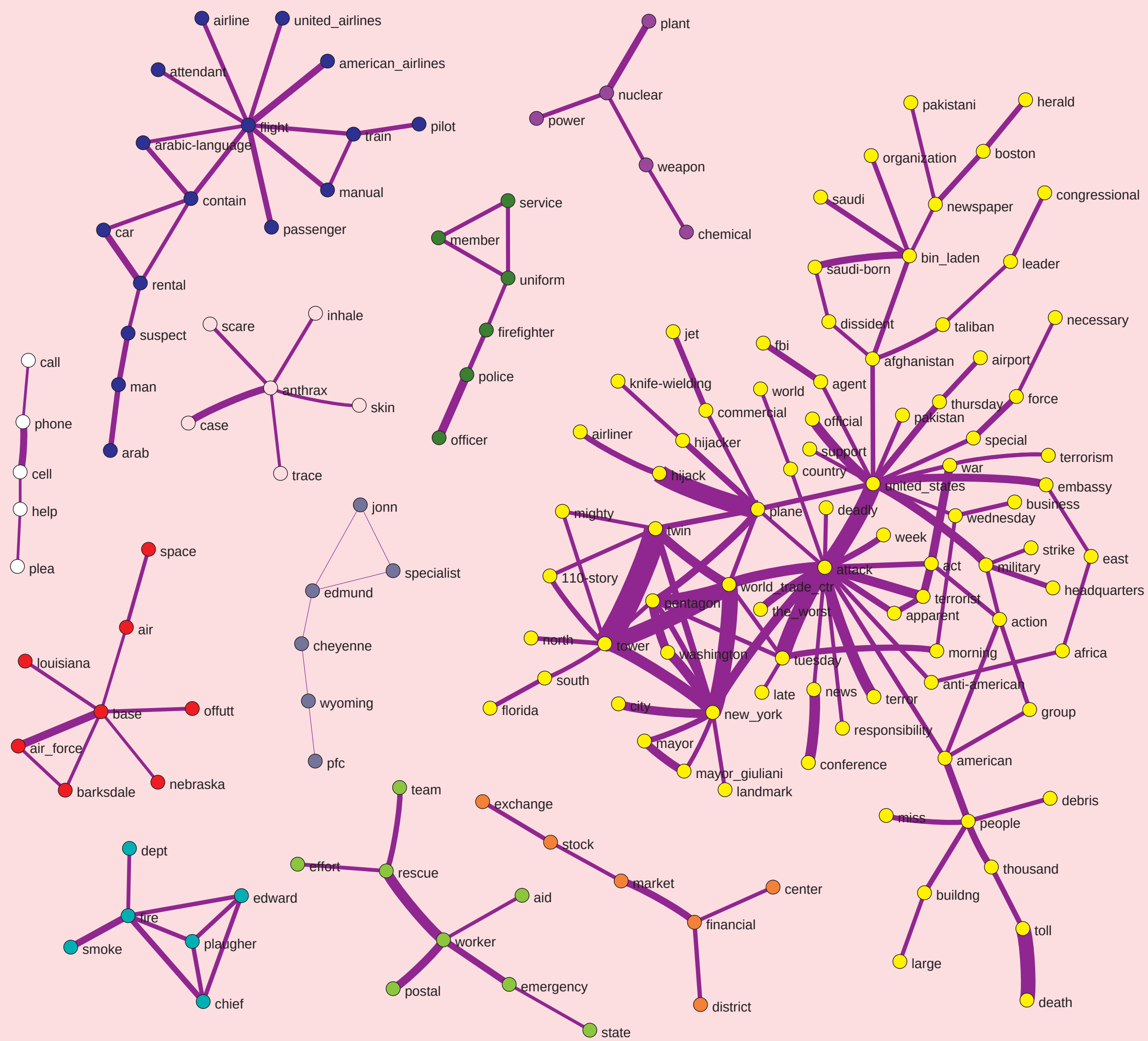
- The set of vertices $\mathcal{C} \subseteq \mathcal{V}$ is a *local line peak*, if it is a regular line island and there exists a spanning tree of the corresponding induced network, in which all lines have the same value as the line with the largest value.
- Line island with a single local line peak is called a *simple line island*.
- The types of line islands:
  - FLAT – there exists a spanning tree, in which all lines have the same value as the line with the largest value.
  - SINGLE – island has a single local line peak.
  - MULTI – island has more than one local line peaks.
- Only the islands of type FLAT or SINGLE are simple islands.

Determining the type of line islands

$p1 := i1.type = \text{FLAT} \wedge (i1.port = \mathbf{null} \vee w(i1.port) = w(e))$
$p2 := i2.type = \text{FLAT} \wedge (i2.port = \mathbf{null} \vee w(i2.port) = w(e))$

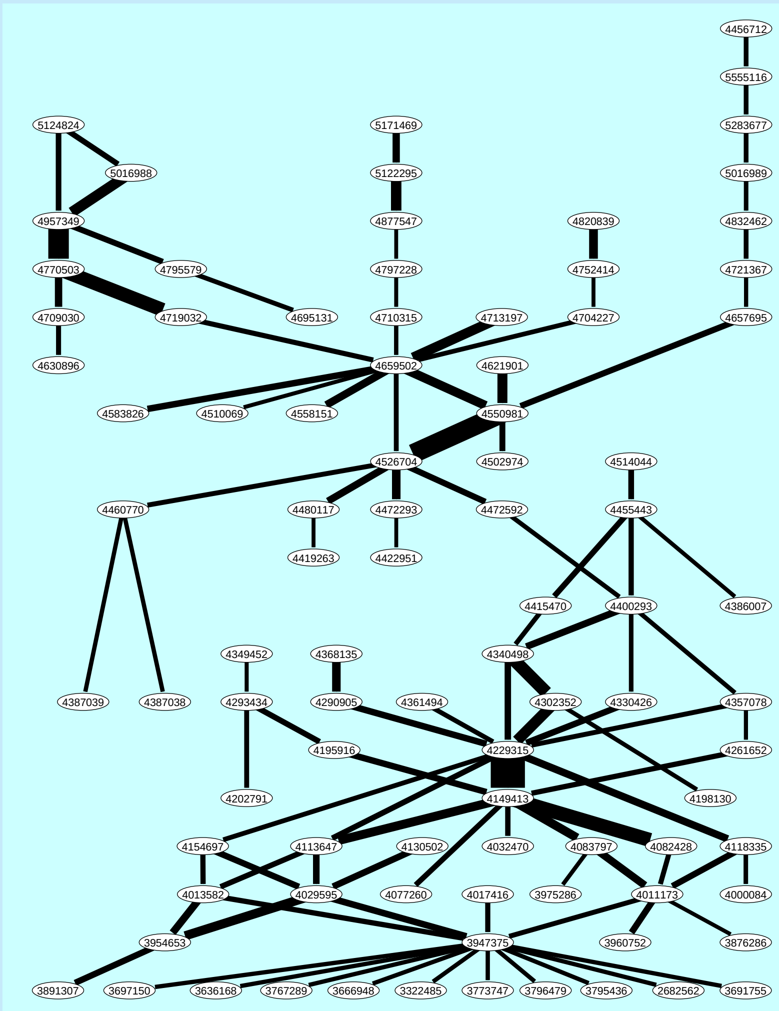
if $p1 \wedge p2$ **then** $island.type := \text{FLAT}$
else if $p1 \vee p2$ **then** $island.type := \text{SINGLE}$
else $island.type := \text{MULTI}$

Example: 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.

Example: US Patents



The citation network of US patents from 1963 to 1999 (`http://www.nber.org/patents/`) is an example of very large network (3774768 vertices and 16522438 arcs) that, using some special options in **`Pajek`**, can still be analyzed on PC with at least 1 G memory. The islands algorithm was applied on Hummon-Doreian SPC weights.

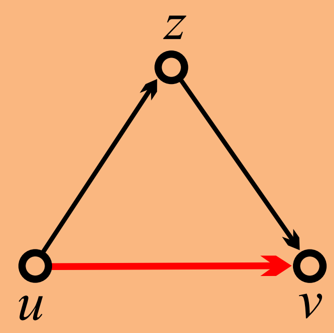
The obtained main island is presented in the figure. The vertices represent patents, the size of a line is proportional to its weight. Collecting from the ***United States Patent and Trademark Office*** (`http://patft.uspto.gov/netahtml/srchnum.htm`) the basic data about the patents we can see that they deal with the 'liquid crystal displays'.

Example: The Edinburgh Associative Thesaurus

- The Edinburgh Associative Thesaurus is a set of words and the counts of word associations as collected from subjects.
- The data were collected by asking several people to say a word which first comes to their mind upon receiving the stimulus word.
- The network contains 23219 vertices (words) and 325624 arcs (stimulus→response), including 564 loops. Almost 70% of arcs have value 1.
- The subjects were mostly undergraduates from a wide variety of British universities. The age range of the subjects was from 17 to 22 with a mode of 19. The sex distribution was 64 per cent male and 36 per cent female. The data were collected between June 1968 and May 1971.

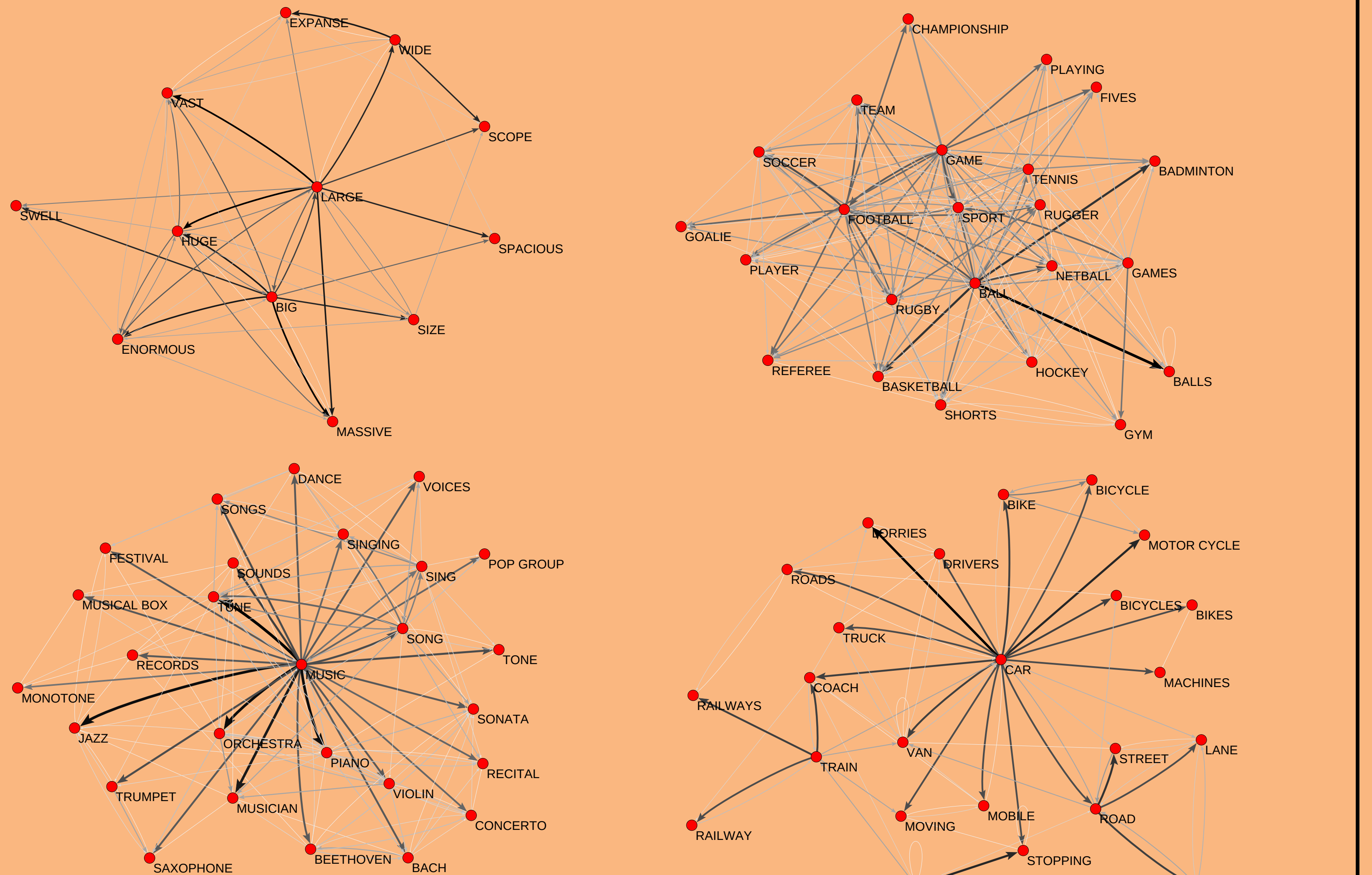
Transitivity weight

- We would like to identify the most important themes – groups of words with the strongest ties.
- For each arc we determined its weight by counting, to how many transitive triangles it belongs (we are also interested in indirect ties).

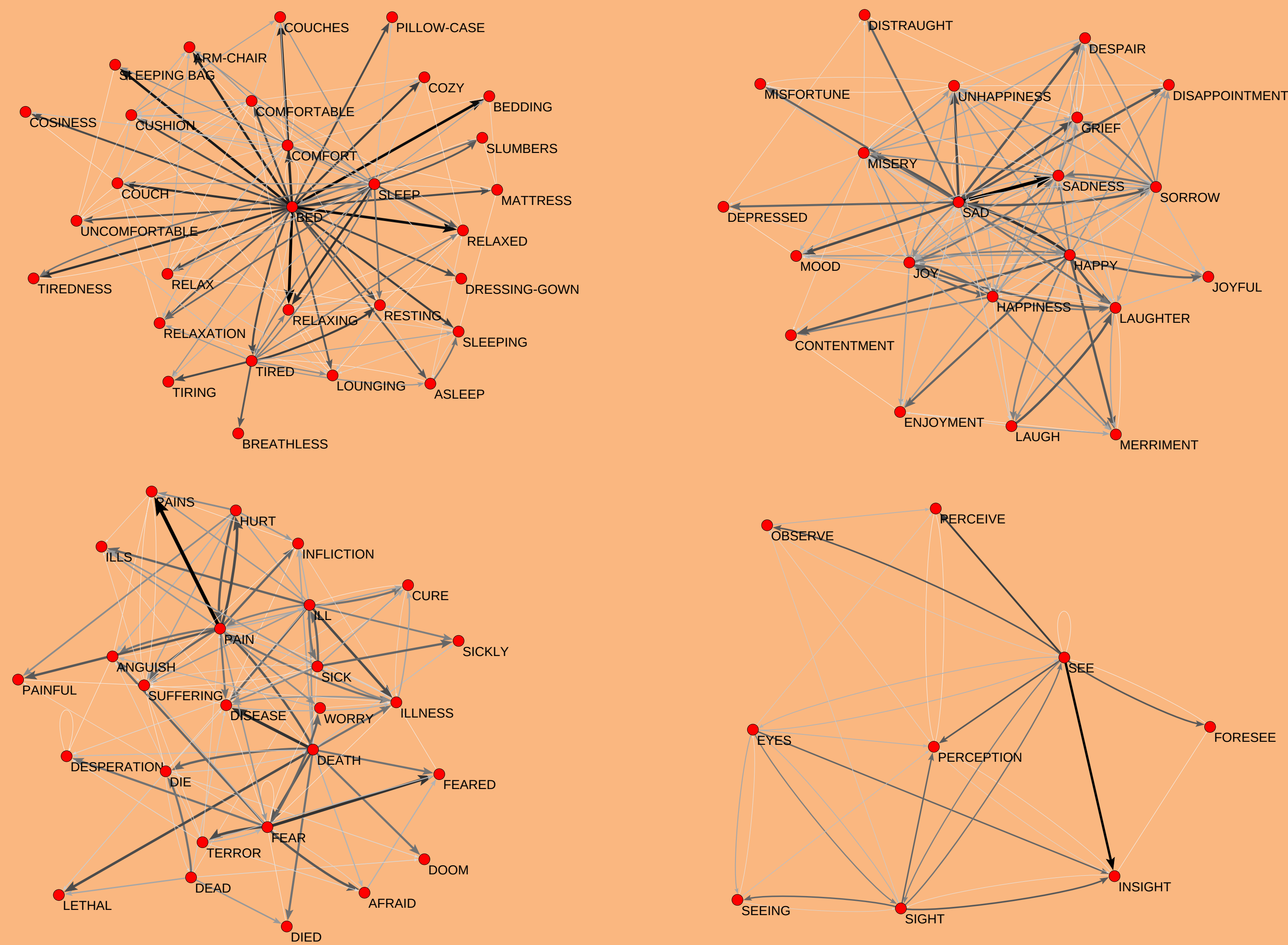


- There are 53 line islands of size at least 5 and at most 30. They contain 664 vertices (all together).

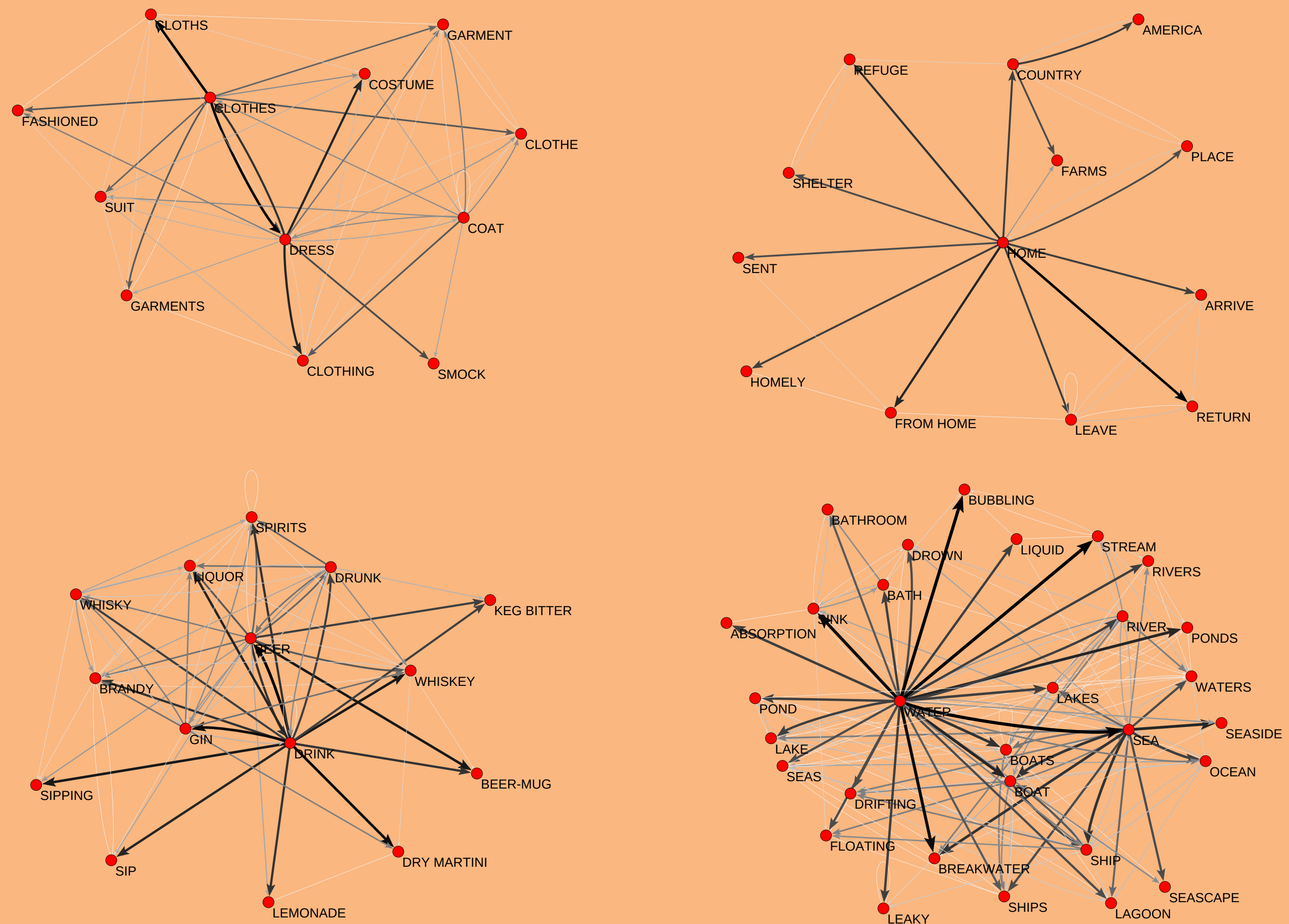
Selected themes in EAT



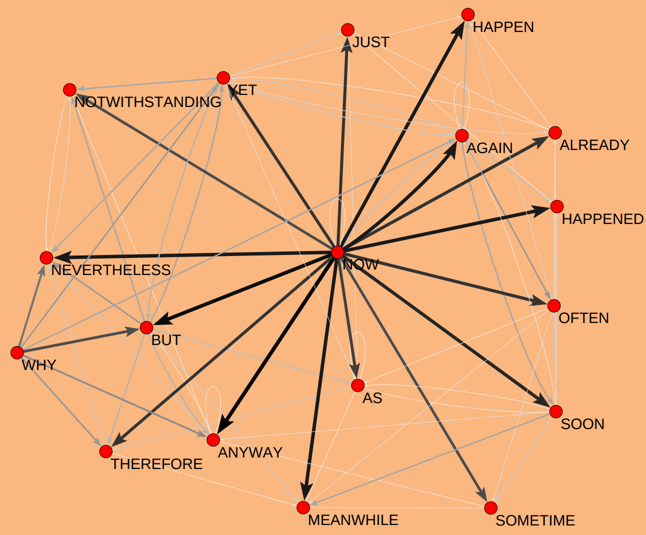
Selected themes in EAT

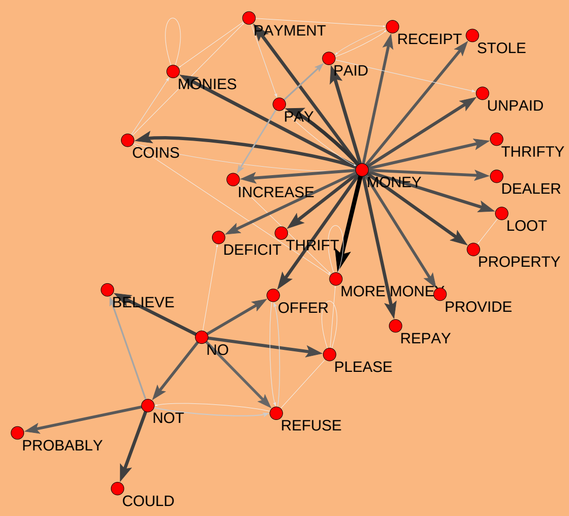


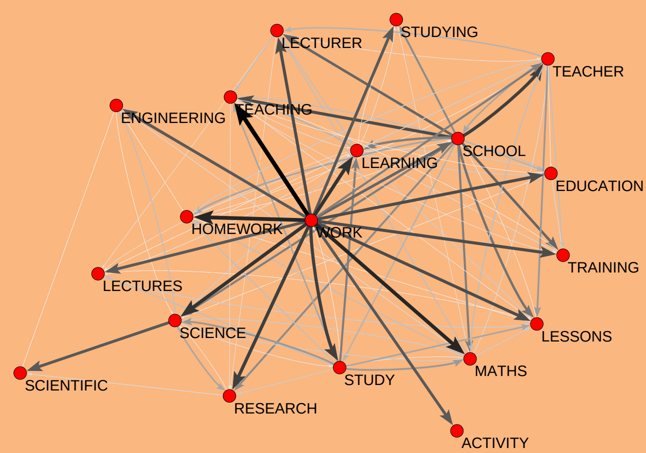
Selected themes in EAT

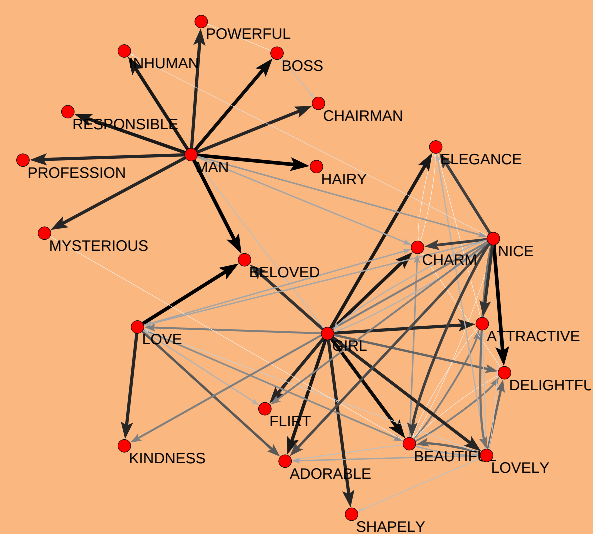


Selected themes in EAT



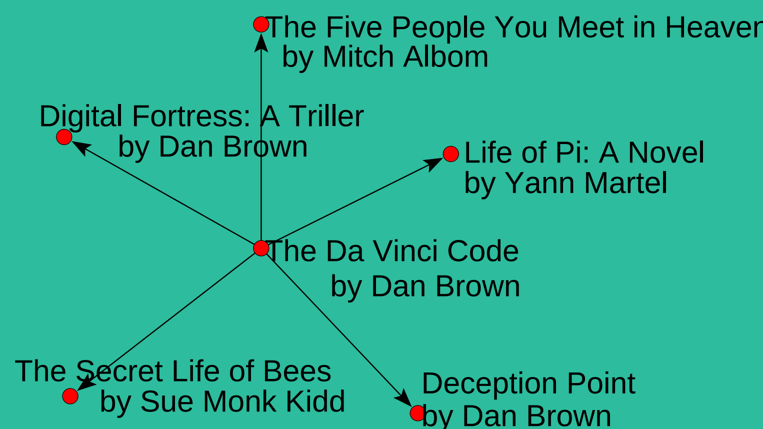






Example: Amazon CDs and books networks

The *vertices* in Amazon networks are books / CDs; while the *arcs* are determined based on the list of products (CDs/books) under the title: Customers who bought this CD/book also bought



... Amazon CDs and books networks

Using relatively simple program written in Python we 'harvested' the books network from June 16 till June 27, 2004; and the CDs network from July 7 till July 23, 2004.

We harvested only the portion of each network reachable from the selected starting book/CD.

The books network has 216737 vertices and 982296 arcs.

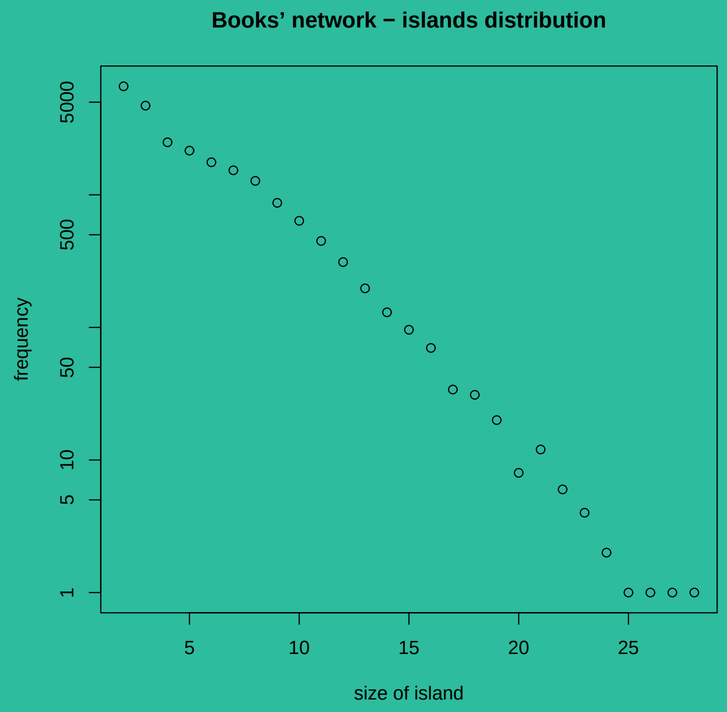
The CDs network has 79244 vertices and 526271 arcs.

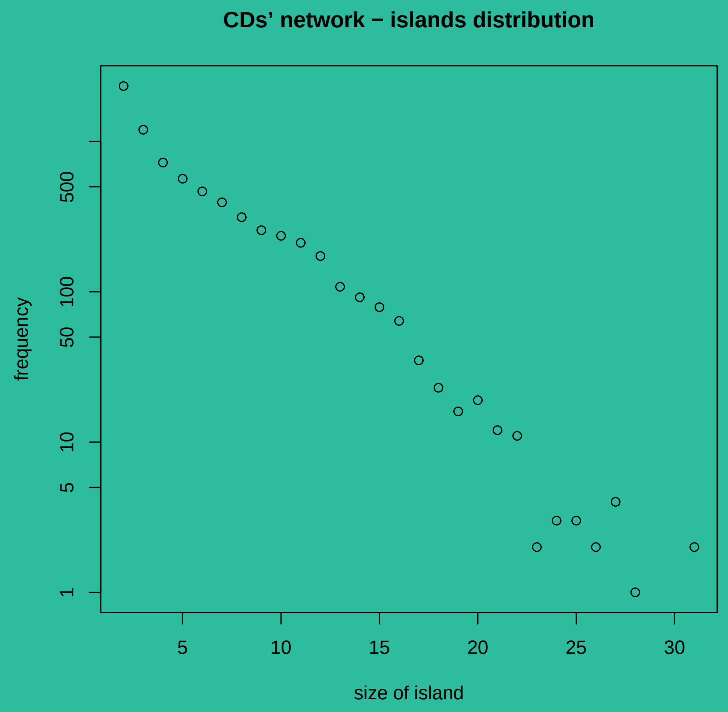
By the construction both networks have limited out-degree and are weakly connected. 178281 books have the out-degree 5; and 55373 CDs have out-degree 8.

The networks were analysed by *Nataša Kejžar* and *Simona Korenjak-Černe*.

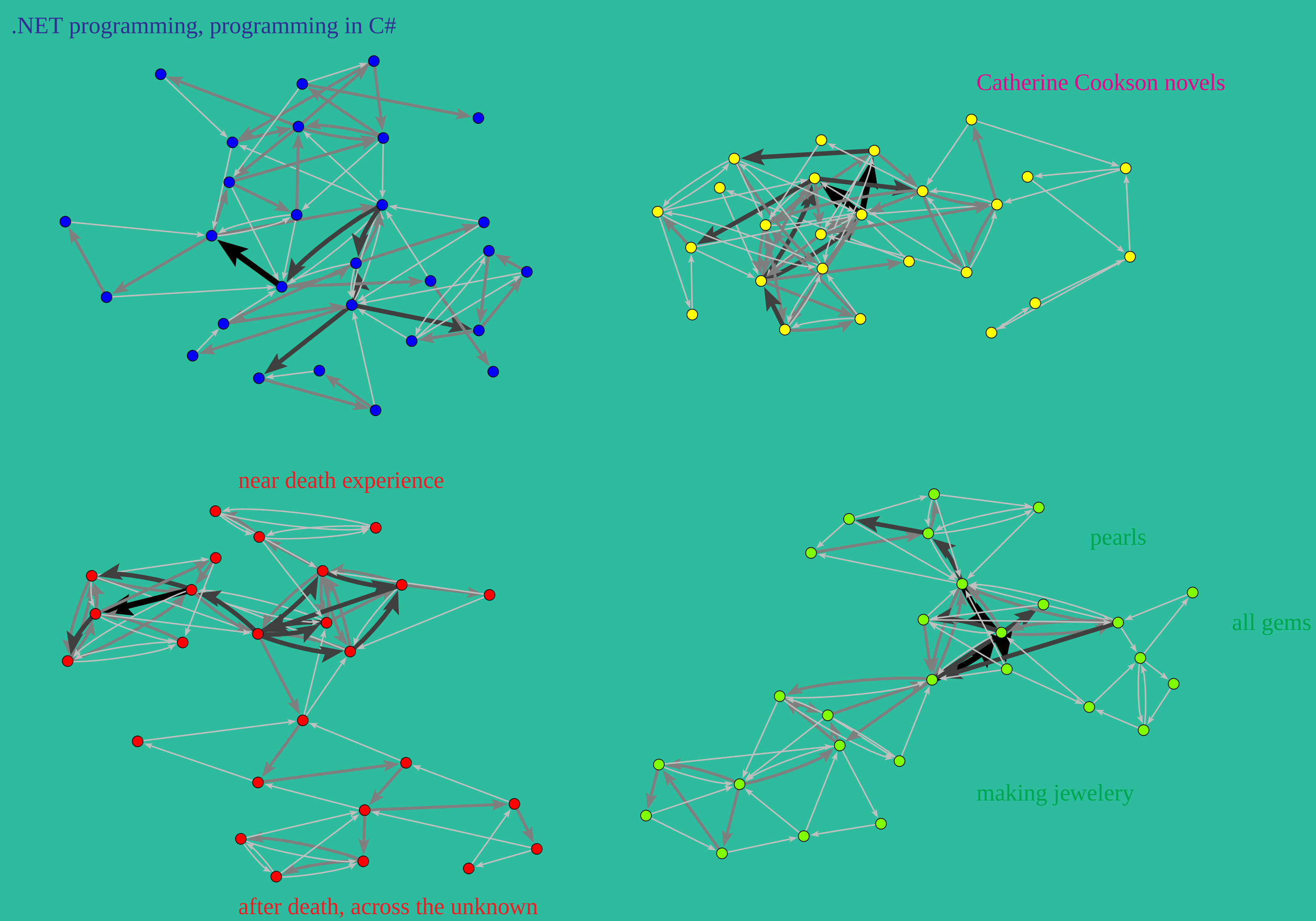
Simple arc islands size distribution

We took the number of cyclic triangles as weights on arcs.

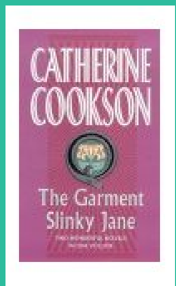


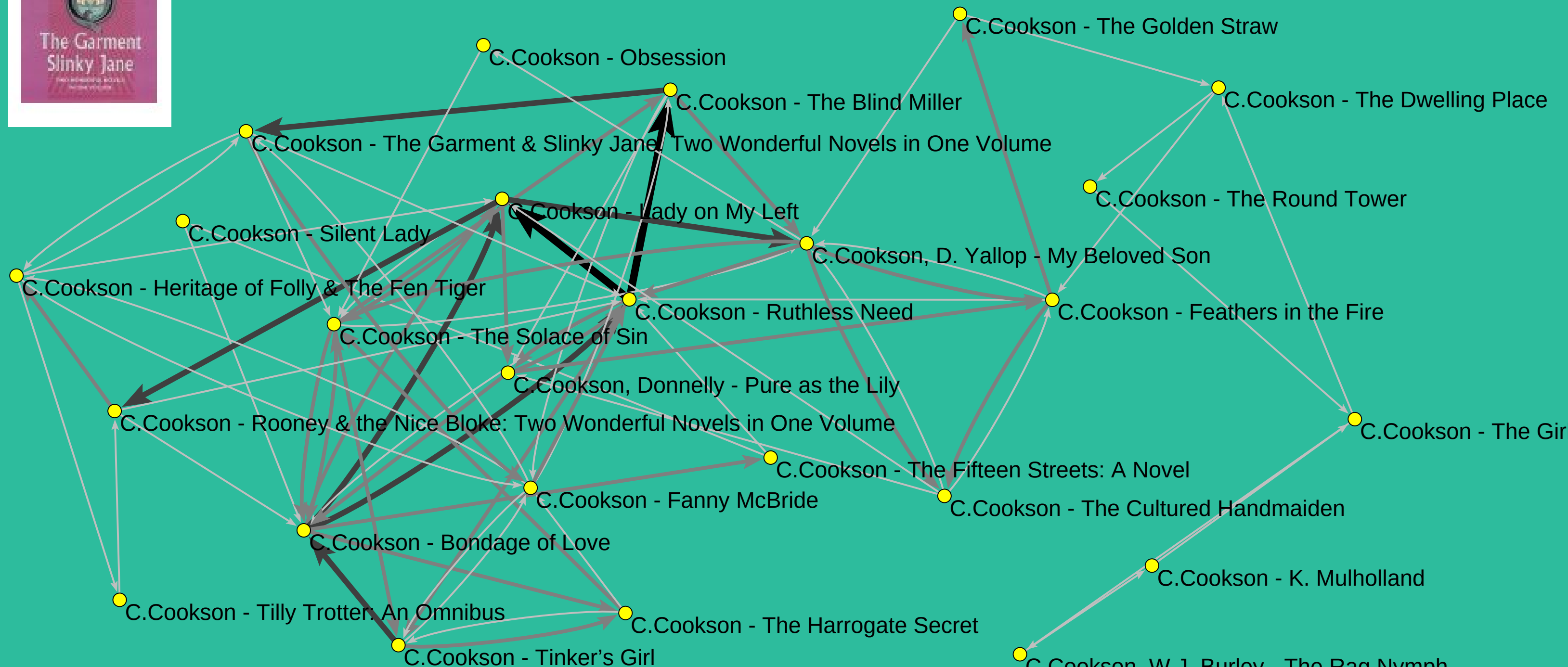


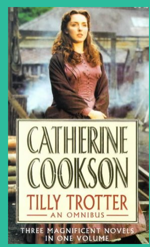
Islands with at least 25 vertices



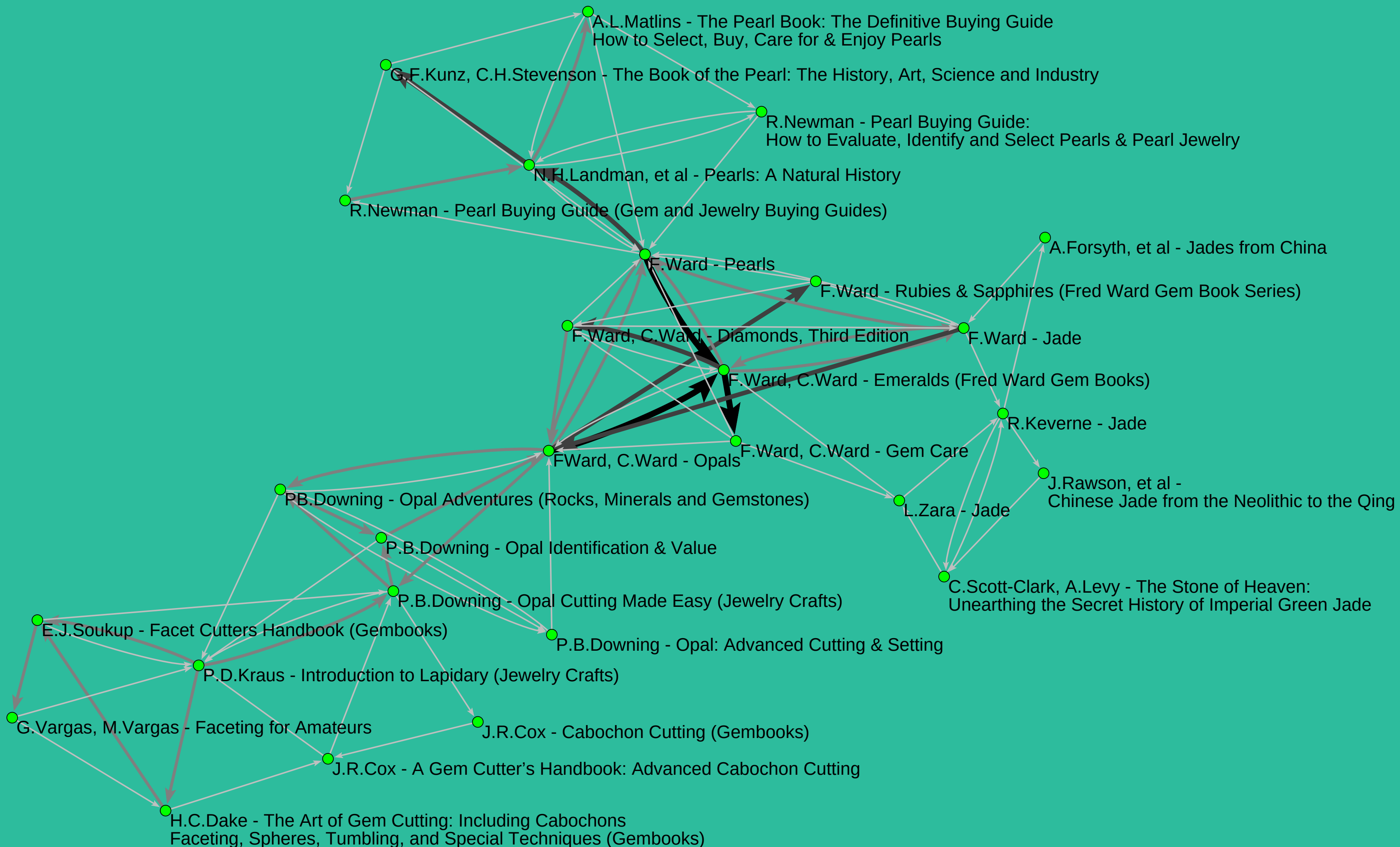
Island of Catherine Cookson novels







Island of precious stones



Conclusions

- We proposed an approach to the analysis of networks that can be used also for very large networks with millions of vertices and lines.
- The proposed approach is very general – it can be applied to any property of vertices (vertex islands) and to any weight on lines (line islands).
- The islands algorithms are implemented in **`Pajek`** – a program (for Windows) for large network analysis and visualization

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

  They are available also as a separate program at

  `http://vlado.fmf.uni-lj.si/pub/networks/`
- The last version of these slides is available at

  `http://vlado.fmf.uni-lj.si/pub/networks/doc/mix/islands.pdf`