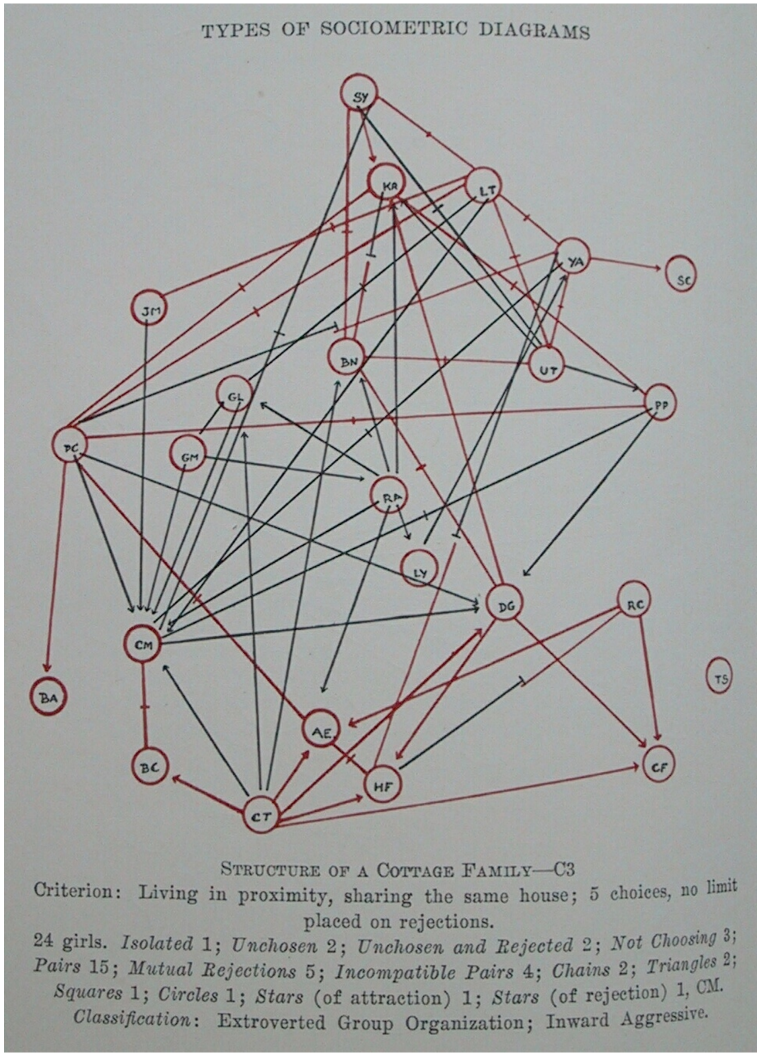


Course on Social Network Analysis

Graphs and Networks

Vladimir Batagelj

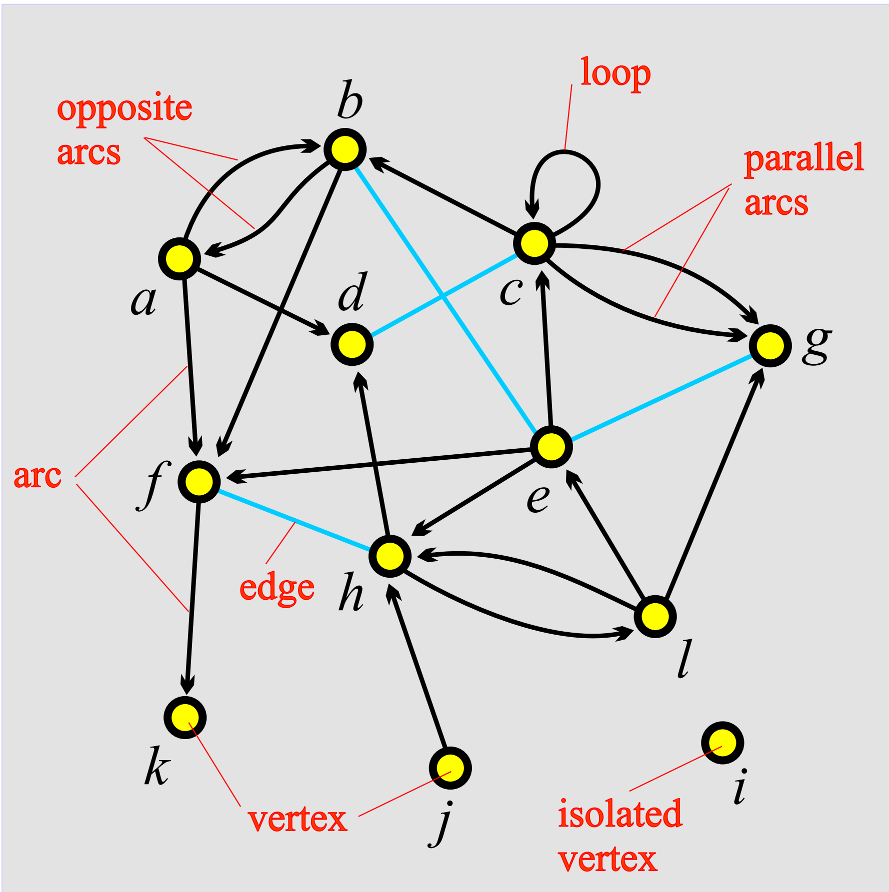
University of Ljubljana

Slovenia

Padova, April 10-11, 2003

Outline

Graph



actor – vertex, node

relation – line, edge, arc, link, tie

arc = directed line, (a, d)

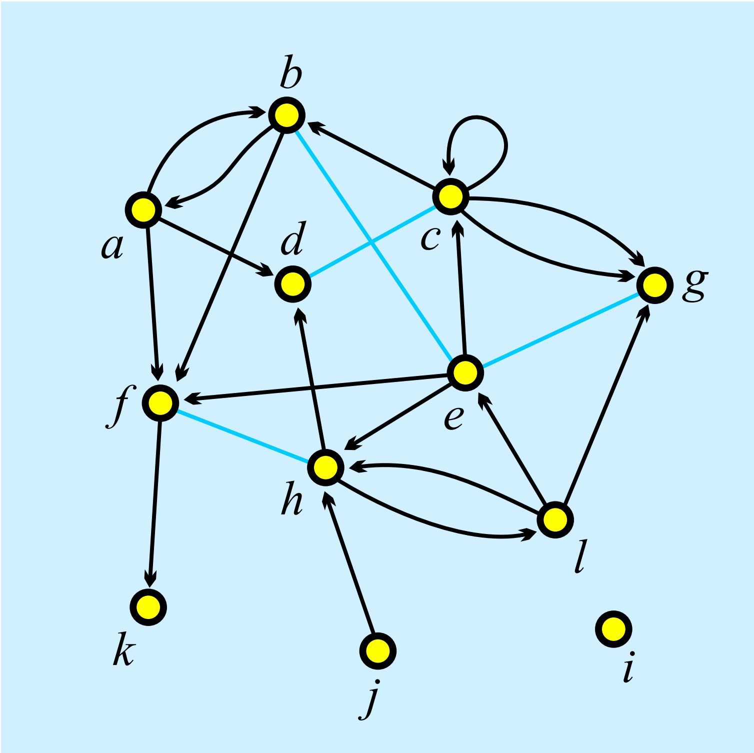
a is the *initial* vertex,

d is the *terminal* vertex.

edge = undirected line, $(c{:}\, d)$

c and d are *end* vertices.

Graph / Sets



$$V = \{a, b, c, d, e, f, g, h, i, j, k, l\}$$

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

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

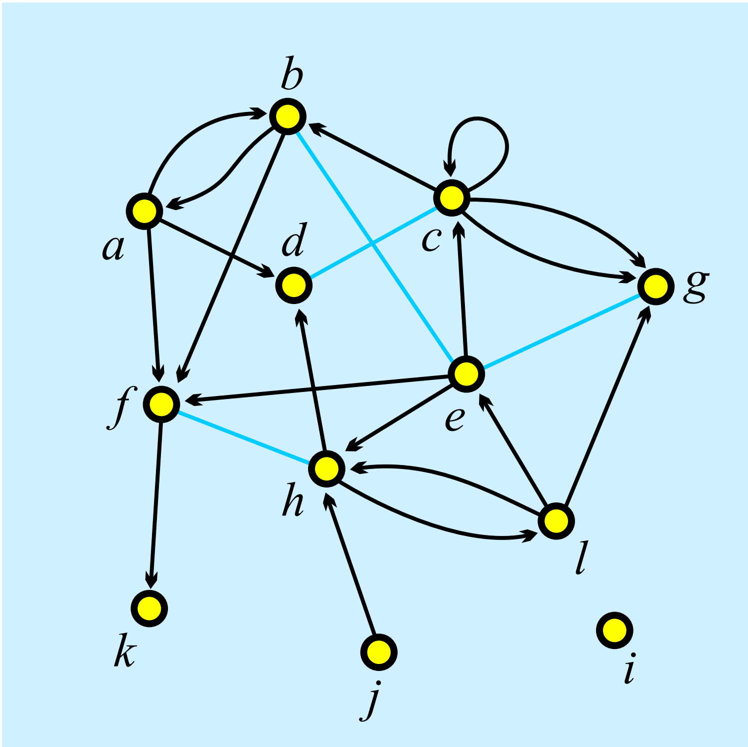
$$G = (V, A, E)$$

$$L = A \cup E$$

$A = \emptyset$ – *undirected* graph; $E = \emptyset$ – *directed* graph.

Pajek: `GraphSet`; `TinaSet`.

Graph / Neighbors



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

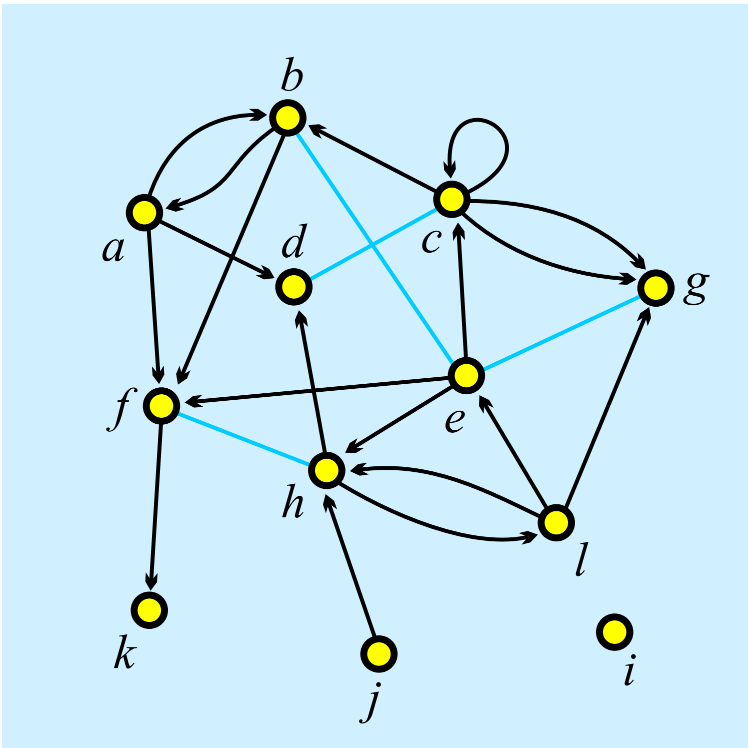
$$N_E(e) = \{b, g\}$$
$$N_E(c) = \{d\}$$
$$N_E(f) = \{h\}$$

Pajek: `GraphList`; `TinaList`.

$$N(v) = N_A(v) \cup N_E(v), \qquad = N_{out}(v),\ N_{in}(v)$$

Star in v, $S(v)$ is the set of all lines with v as their initial vertex.

Graph / Matrix

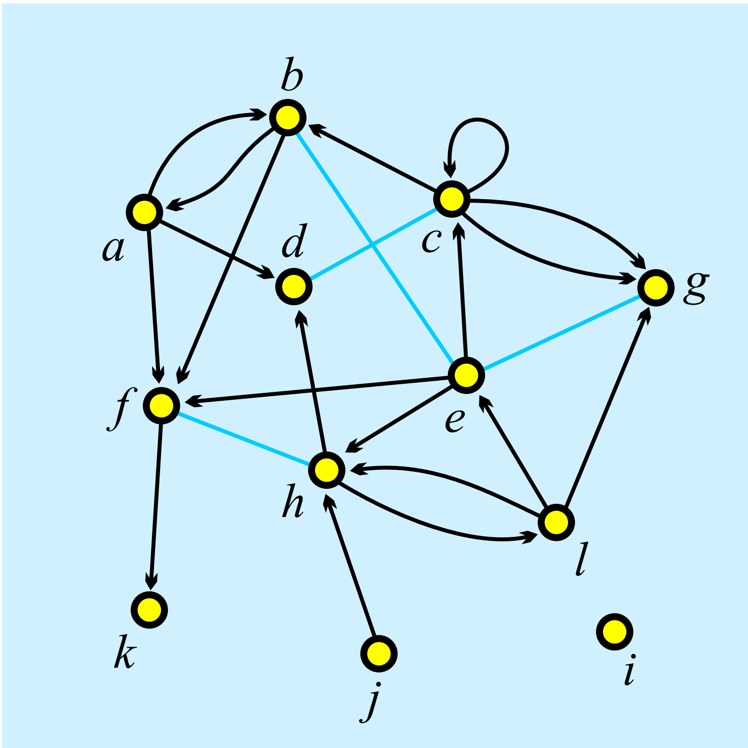


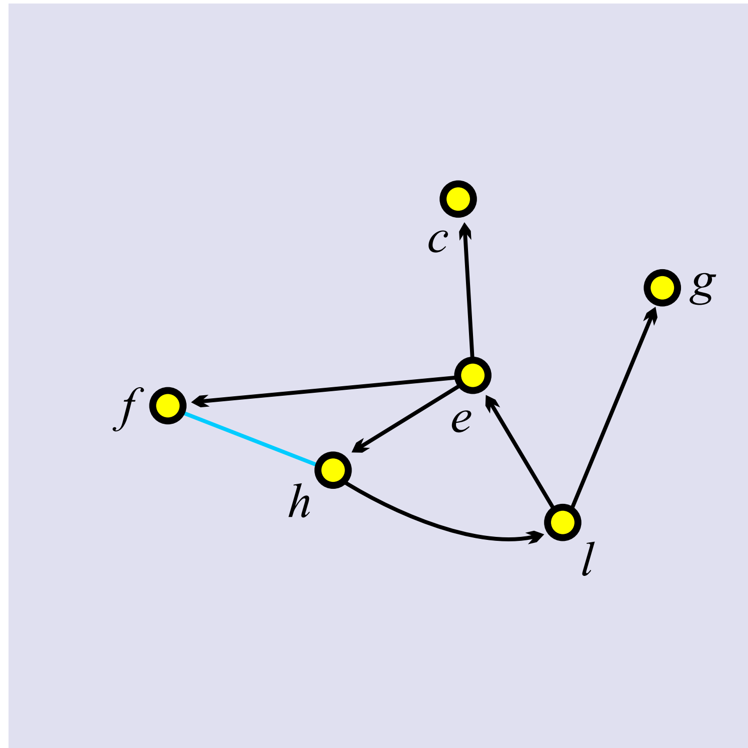
	a	b	c	d	e	f	g	h	i	j	k	l
a	0	1	0	1	0	1	0	0	0	0	0	0
b	1	0	0	0	1	1	0	0	0	0	0	0
c	0	1	1	1	0	0	2	0	0	0	0	0
d	0	0	1	0	0	0	0	0	0	0	0	0
e	0	1	1	0	0	1	1	1	0	0	0	0
f	0	0	0	0	0	0	0	1	0	0	1	0
g	0	0	0	0	1	0	0	0	0	0	0	0
h	0	0	0	1	0	1	0	0	0	0	0	1
i	0	0	0	0	0	0	0	0	0	0	0	0
j	0	0	0	0	0	0	0	1	0	0	0	0
k	0	0	0	0	0	0	0	0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

Pajek: `GraphMat`; `TinaMat`, picture `picture`.

Graph G is *simple* if in the corresponding matrix all entries are 0 or 1.

Subgraph

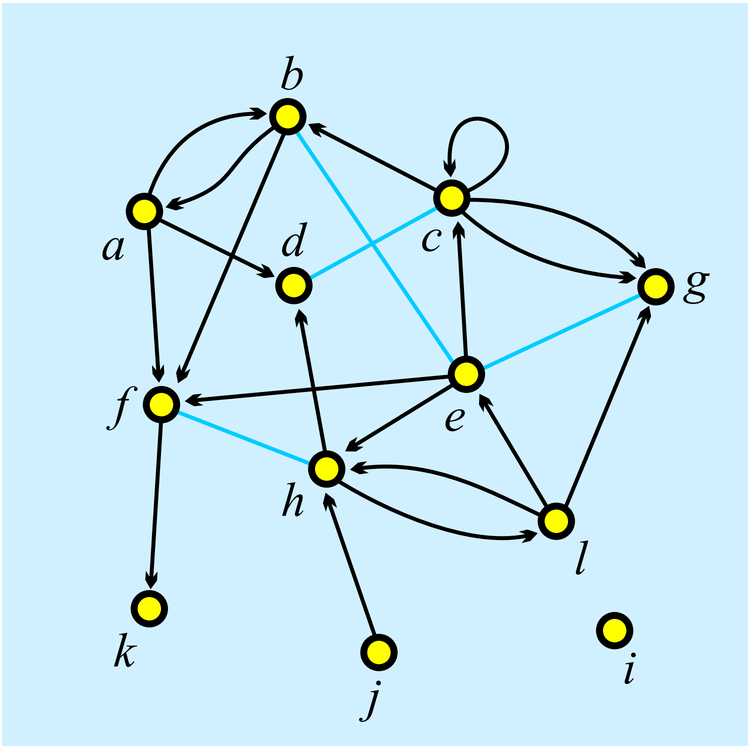




A *subgraph* $H = (V', L')$ of a given graph $G = (V, L)$ is a graph which set of lines is a subset of set of lines of G, $L' \subseteq L$, its vertex set is a subset of set of vertices of G, $V' \subseteq V$, and it contains all end-vertices of L'.

A subgraph can be *induced* by a given subset of vertices or lines.

Graph characteristics



number of vertices $n = |V|$

number of lines $m = |L|$

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

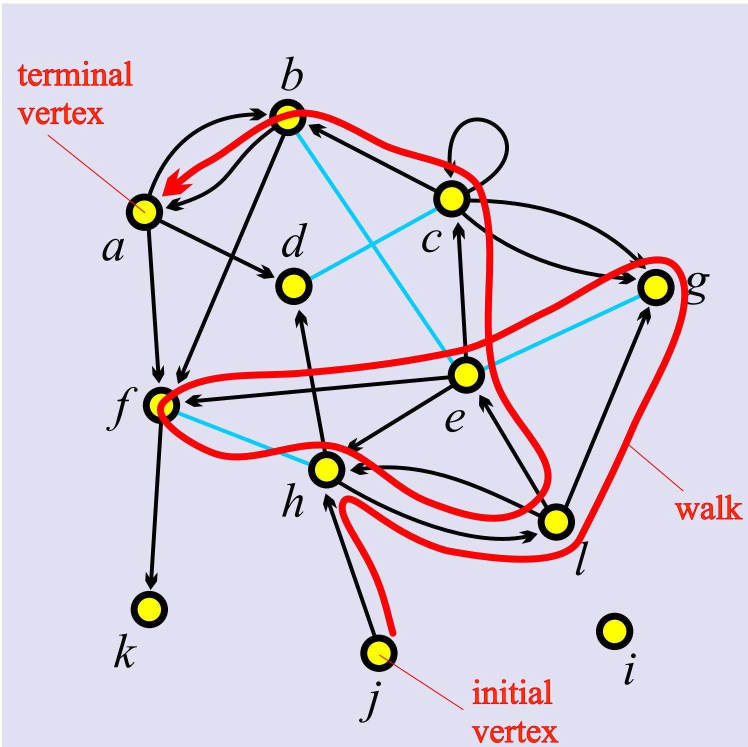
indegree of vertex v, $\text{indeg}(v)$ = number of lines with v as terminal vertex (end-vertex is both initial and terminal);

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

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

$$\sum_{v \in V} \text{indeg}(v) = \sum_{v \in V} \text{outdeg}(v) = |A| + 2|E|,\ \sum_{v \in V} \deg(v) = 2|L| - |E_0|$$

Walks



length $|s|$ of the walk s is the number of lines it contains.

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

$|s| = 11$

A walk is *closed* iff its initial and terminal vertex coincide.

If we don't consider the direction of the lines in the walk we get a *semiwalk* or *chain*.

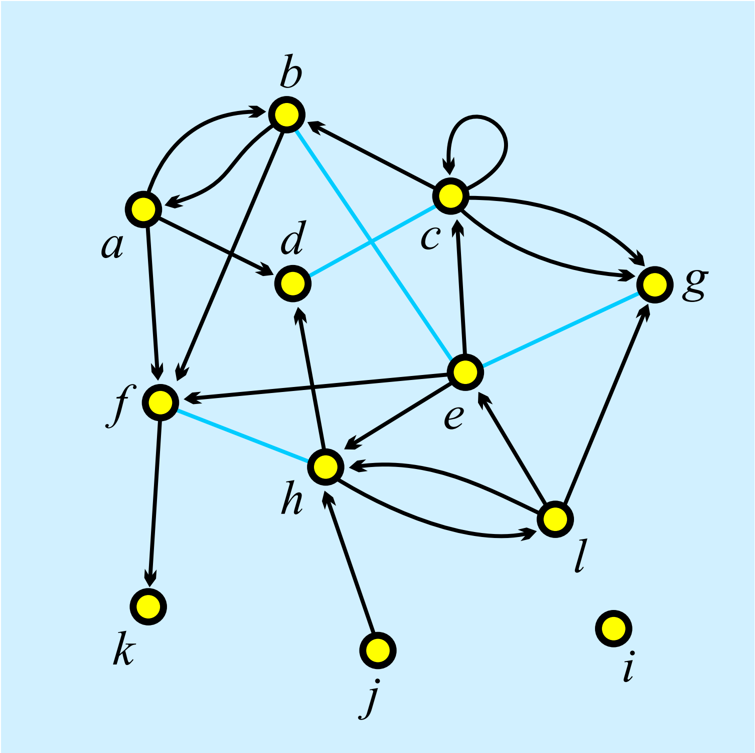
trail – walk with all lines different

path – walk with all vertices different

cycle – closed walk with all internal vertices different

A graph is *acyclic* if it doesn't contain any cycle.

Shortest paths



A shortest path from u to v is also called a *geodesic* from u to v. Its length is denoted by $d(u, v)$.

If there is no walk from u to v then $d(u, v) = \infty$.

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

$d(a, j) = \infty$

$\hat{d}(u, v) \;=\; \max(d(u, v), d(v, u))$ is a *distance*.

The *diameter* of a graph equals to the distance between the most distant pair of vertices: $D = \max_{u,v \in V} d(u, v)$.

Equivalence relations and Partitions

Let $\mathbf{C} = \{C_i\}$ be a set of subsets of V, $\emptyset \subset C_i \subseteq V$. $\mathbf{C}$ is a *partition* of V iff $\bigcup_i C_i = V$ and for $i \neq j$, $C_i \cap C_j = \emptyset$.

A relation R on V is an *equivalence* relation iff it is reflexive $\forall v \in V : vRv$, symmetric $\forall u, v \in V : uRv \Rightarrow vRu$, and transitive $\forall u, v, z \in V : uRz \wedge zRv \Rightarrow uRv$.

Each equivalence relation determines a partition into equivalence classes $[v] = \{u : vRu\}$.

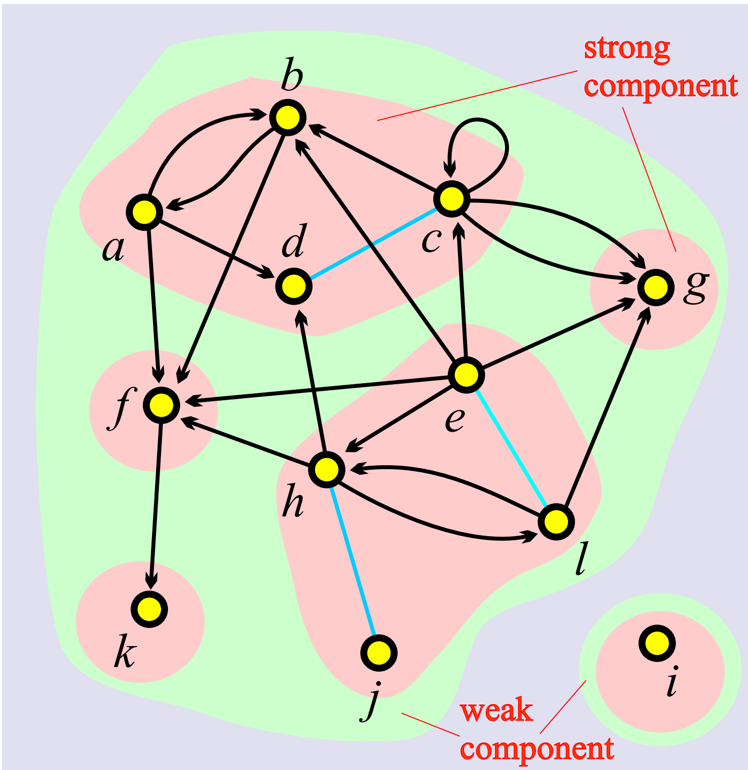
Each partition $\mathbf{C}$ determines an equivalence relation $uRv \Leftrightarrow \exists C \in \mathbf{C} : u \in C \wedge v \in C$.

k-neighbors of v is the set of vertices on 'distance' k from v, $N^k(v) = \{u \in v : d(v, u) = k\}$.

The set of all k-neighbors, $k = 0, 1, \ldots$ of v is a partition of V.

k-neighborhood of v, $N^{(k)}(v) = \{u \in v : d(v, u) \leq k\}$.

Connectivity



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

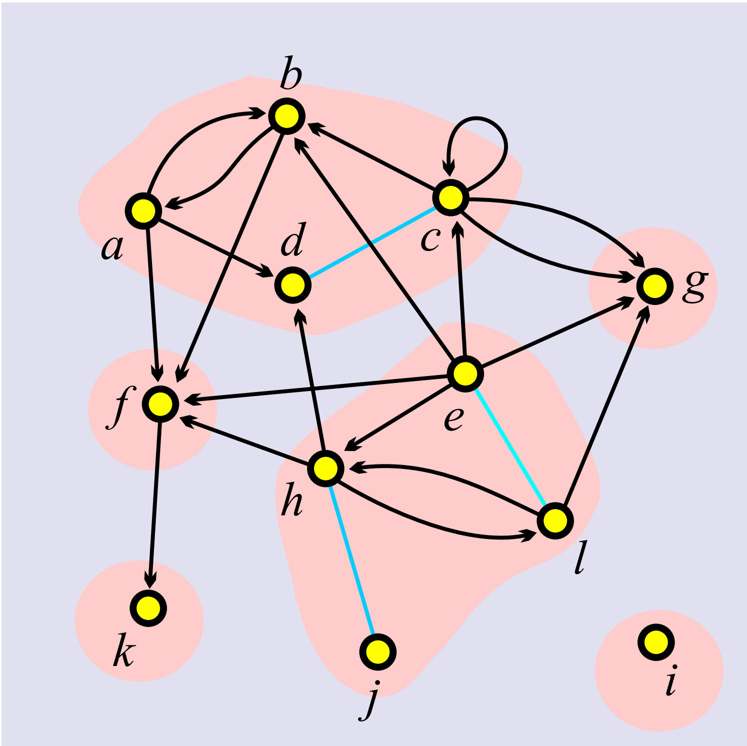
Vertex v is *weakly connected* with vertex u iff there exists a semiwalk with v and u as its end-vertices.

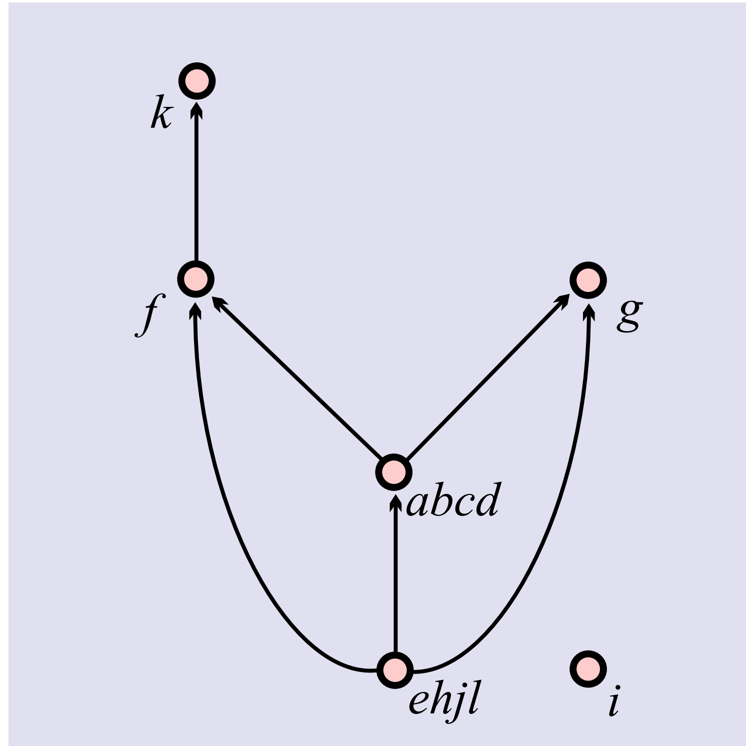
Vertex v is *strongly connected* with vertex u iff they are mutually reachable.

Weak and strong connectivity are equivalence relations.

Equivalence classes induce weak/strong *components*.

Reduction





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 $i : V \to \mathbb{N}$ exists s.t. $(u, v) \in A \Rightarrow i(u) < i(v)$.

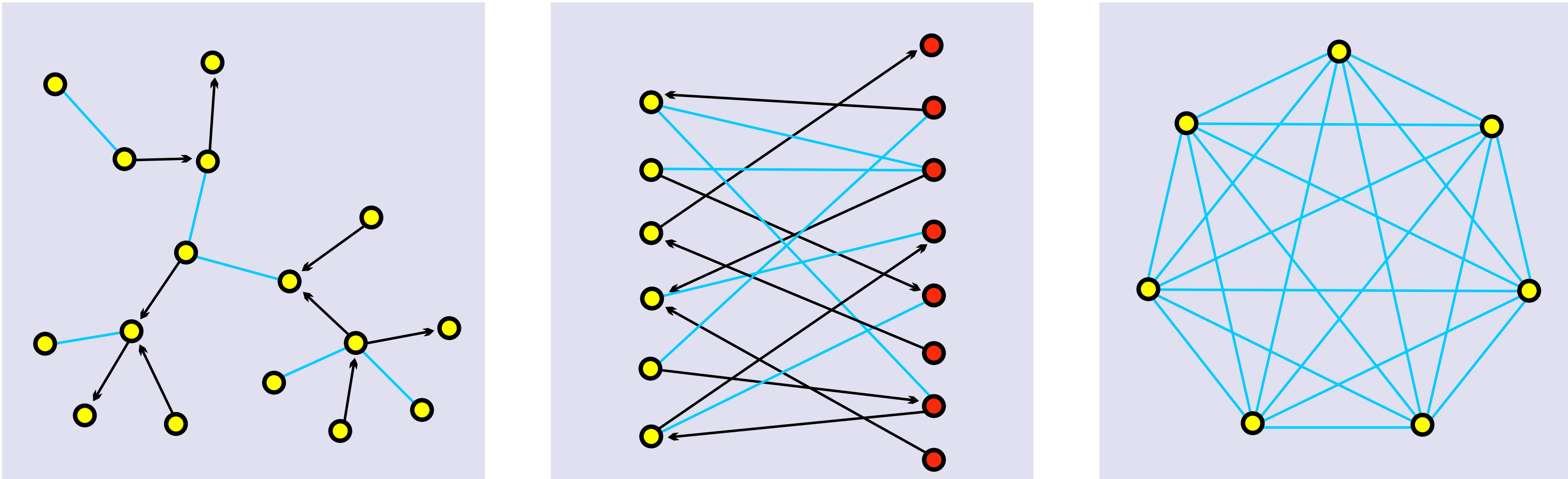
Biconnectivity

Vertices u and v are *biconnected* iff they are connected (in both directions) by two independent (no common internal vertex) paths.

Biconnectivity determines a partition of the set of lines.

A vertex is an *articulation* vertex iff its deletion increases the number of weak components in a graph.

Special graphs

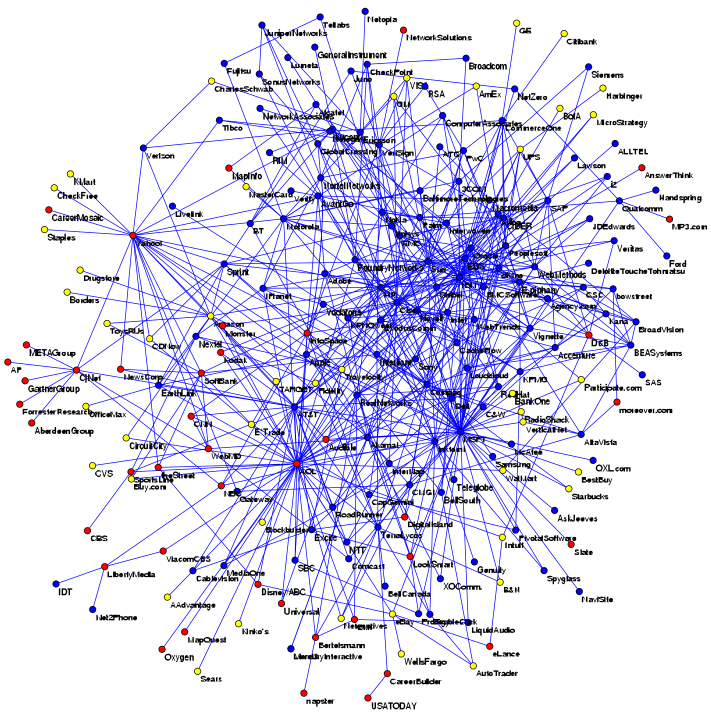


A weakly connected graph G is a *tree* iff it doesn't contain loops and semicycles of length at least 3.

A graph $G = (V, L)$ is *bipartite* iff its set of vertices V can be partitioned into two sets V_1 and V_2 such that every line from L has one end-vertex in V_1 and the other in V_2.

A simple undirected graph is *complete*, K_n, iff it contains all possible edges.

Krebs' Internet Industry Companies



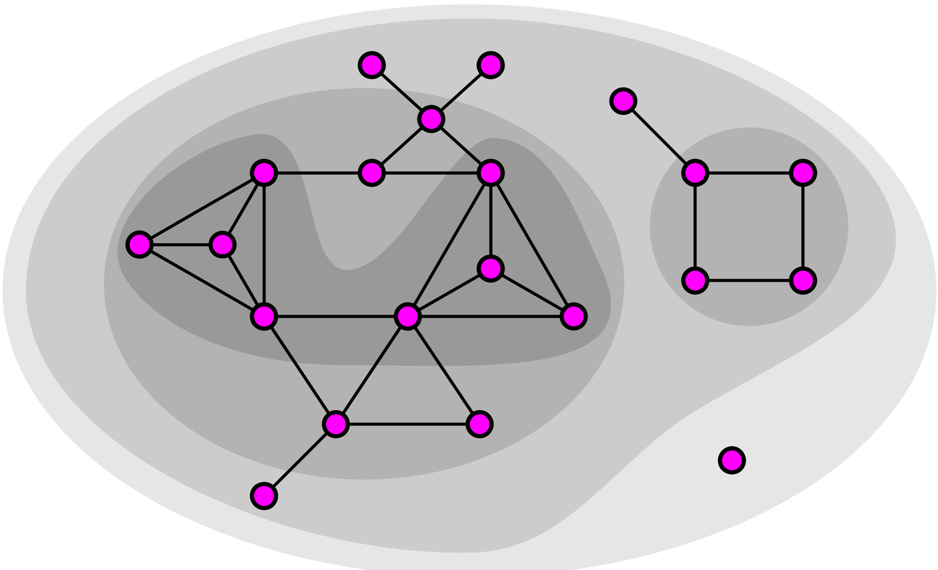
This network shows a subset of the total internet industry during the period from 1998 to 2001, $n = 219$, $m = 631$.

Two companies are connected with a line if they have announced a joint venture, strategic alliance or other partnership (red - content, blue - infrastructure, yellow - commerce).

Network source: **http://www.orgnet.com/netindustry.html**.

Cores

The notion of core was introduced by Seidman in 1983.



A subgraph $\mathbf{H} = (W, L|W)$ induced by the set W in a graph $\mathbf{G} = (V, L)$ is a k-*core* or a *core of order* k iff $\forall v \in W : \deg_H(v) \geq k$, and $\mathbf{H}$ is a maximum subgraph with this property.

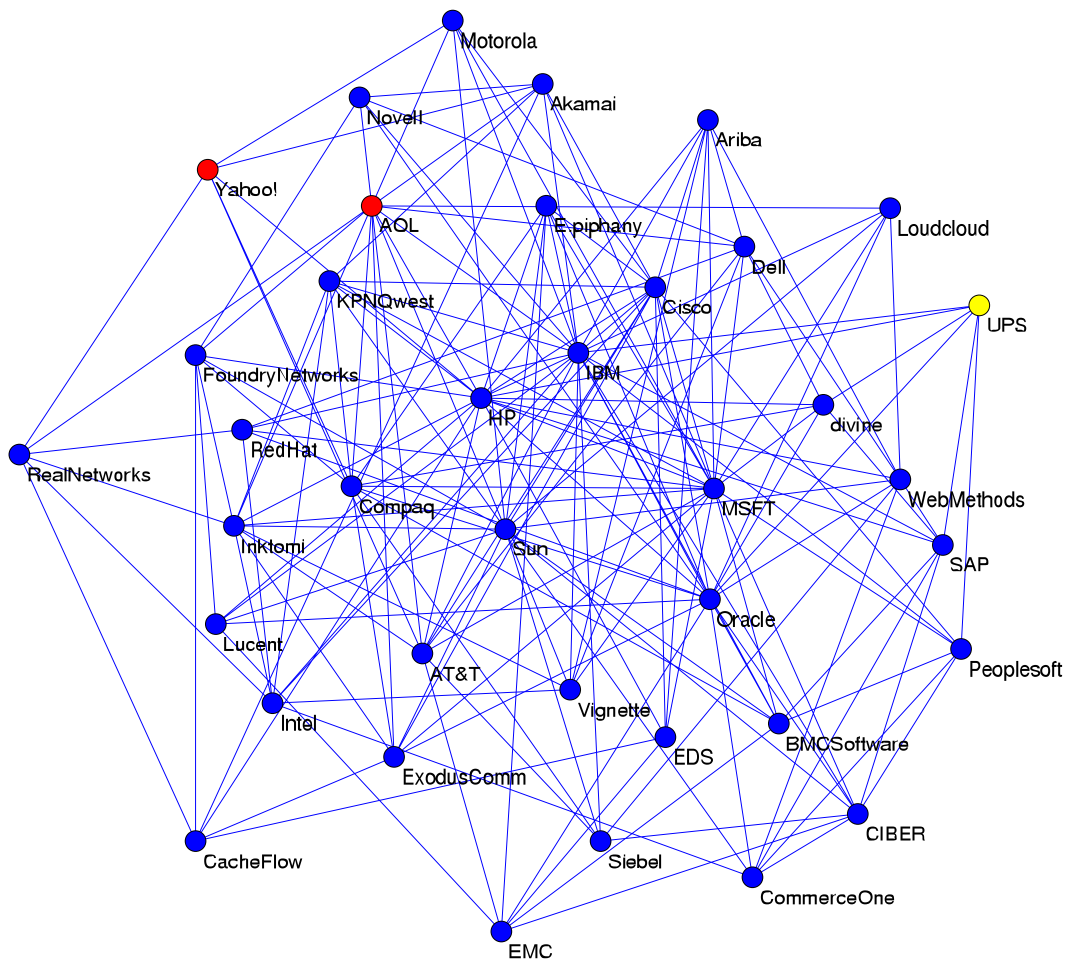
... Cores

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. The degree $\deg(v)$ can be: in-degree, out-degree, in-degree + out-degree, ..., determining different types of cores.

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

The notion of cores can be generalized to *valued cores*.

6-core of Krebs' Internet Industry Companies



Networks

A graph with additional information on vertices and/or lines is called a *network*.

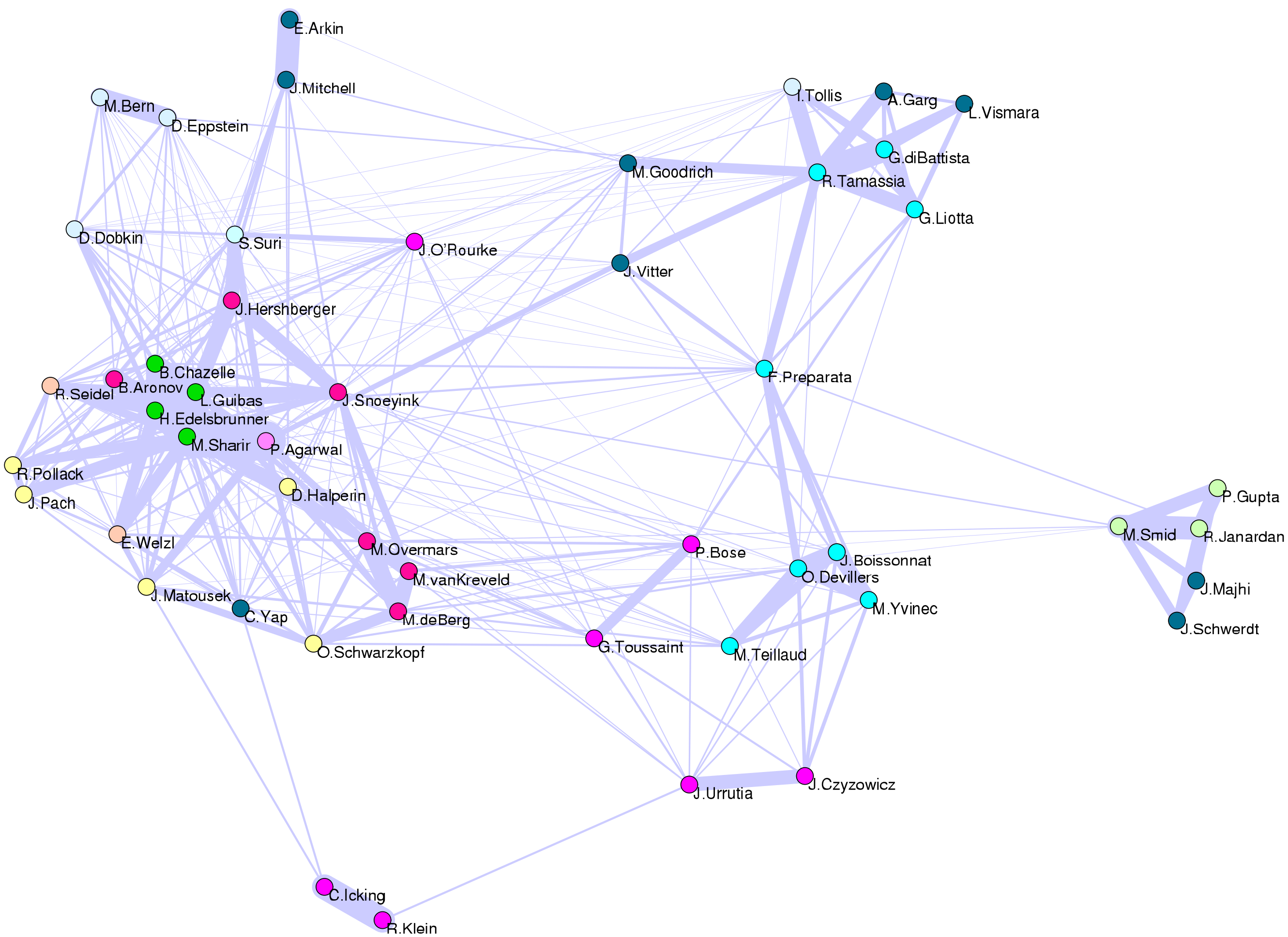
In Pajek this information is represented using *vectors* (numerical properties of vertices) and *partitions* (categorical/nominal properties of vertices). Numerical values can be assigned also to lines – *line values*.

Example: Authors collaboration network based on the *Computational Geometry Database*. Two authors are linked with an edge, iff they wrote a common paper. The weight of the edge is the number of publications they wrote together.

Problem of cleaning. Different names: Pankaj K. Agarwal, P. Agarwal, Pankaj Agarwal, and P.K. Agarwal.

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

Computational Geometry Valued Core



Sources

Vladimir Batagelj, Andrej Mrvar: Pajek.
http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Vladimir Batagelj: Slides on network analysis.
http://vlado.fmf.uni-lj.si/pub/networks/doc/

Vladimir Batagelj: Papers on network analysis.
http://vlado.fmf.uni-lj.si/vlado/vladodat.htm

Andrej Mrvar: Social network analysis.
http://mrvar.fdv.uni-lj.si/sola/info4/programe.htm

B. Jones: Computational geometry database.
http://compgeom.cs.uiuc.edu/~jeffe/compgeom/biblios.html
ftp://ftp.cs.usask.ca/pub/geometry/.