Network

Introduction

Let $\mathbf{U} = \{X_1, X_2, \ldots, X_n\}$ be finite set of units. Connections among units are described using one or more *binary relations* $R_t \subseteq \mathbf{U} \times \mathbf{U}, \quad t = 1, \ldots, r$, which determine *network* $\mathcal{N} = (\mathbf{U}, R_1, R_2, \ldots, R_r)$.

Example: Relation can represent friendship, negative relation, kinship relation (. . .is a child of. . ., . . .is a daughter of . . ., . . .is married to. . .), citations. . .

In the following we will use only one relation R mostly.

$X_i R X_j$ is read as:
unit X_i is in relation R with unit X_j.

Example: if R corresponds to relation 'liking', then $X_i R X_j$, means that person X_i likes person X_j.

Relation can have additional properties:

- *reflexive relation*: $\forall x \in \mathbf{U} : xRx$
- *irreflexive relation*: $\forall x \in \mathbf{U} : \neg xRx$
- *symmetric relation*: $\forall x, y \in \mathbf{U} : (xRy \Rightarrow yRx)$ (marriage)
- *asymmetric relation*: $\forall x, y \in \mathbf{U} : \neg(xRy \wedge yRx)$ (is a son of)
- *antisymmetric relation*: $\forall x, y \in \mathbf{U} : (xRy \wedge yRx \Rightarrow x = y)$
- *transitive relation*: $\forall x, y, z \in \mathbf{U} : (xRy \wedge yRz \Rightarrow xRz)$
- *intransitive relation*: $\forall x, y, z \in \mathbf{U} : (xRy \wedge yRz \Rightarrow \neg xRz)$
- *comparable relation*: $\forall x, y \in \mathbf{U} : (x \neq y \Rightarrow xRy \vee yRx)$
- *strictly comparable relation*: $\forall x, y \in \mathbf{U} : (xRy \vee yRx)$
- *partial ordering*: relation is partial ordering, if it is reflexive, antisymmetric and transitive,;
- *equivalence relation*: relation is equivalence relation, if it is reflexive, symmetric and transitive.

Network defined using relation R, can be represented in different ways:

- Representation using corresponding **binary matrix** $\mathbf{R} = [r_{ij}]_{n \times n}$, where

$$r_{ij} = \begin{cases} 1 & X_i R X_j \\ 0 & \text{otherwise} \end{cases}$$

  Sometimes r_{ij} is a real number, expressing the strength of relation R between units X_i in X_j.

- **list of neighbours**
  Network can be described by telling for each unit list of all other units with which the unit is in relation.

- description by a **graph** $G = (V, L)$ where V is set of vertices and L set of (directed or undirected) lines. Vertices represent units of a network, lines represent relation.

  Graph is usually represented by a picture, where vertices are drawn as small circles, directed lines are drawn as arcs and undirected lines as edges connecting the corresponding two vertices.

$X_i R X_j \Rightarrow$ there exists directed line from unit X_i to X_j in corresponding graph. Vertex X_i is called *initial*, vertex X_j is called *terminal* vertex.

Line where initial and terminal vertex are the same is called *loop*.

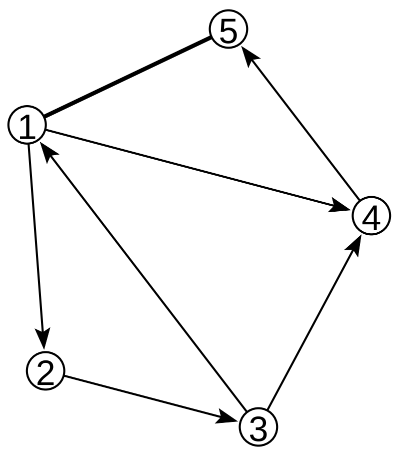
If directed lines between two vertices exist in both directions, they are sometimes replaced by a single undirected line.

Sometimes we do the opposite: undirected line is replaced by two directed lines in opposite directions.

We will denote number of vertices in graph with n, and number of directed lines with m.

Matrix, graph and list of neighbours

	1	2	3	4	5
1	0	1	0	1	1
2	0	0	1	0	0
3	1	0	0	1	0
4	0	0	0	0	1
5	1	0	0	0	0



Arcslist

1 2 4

2 3

3 1 4

4 5

Edgeslist

1 5

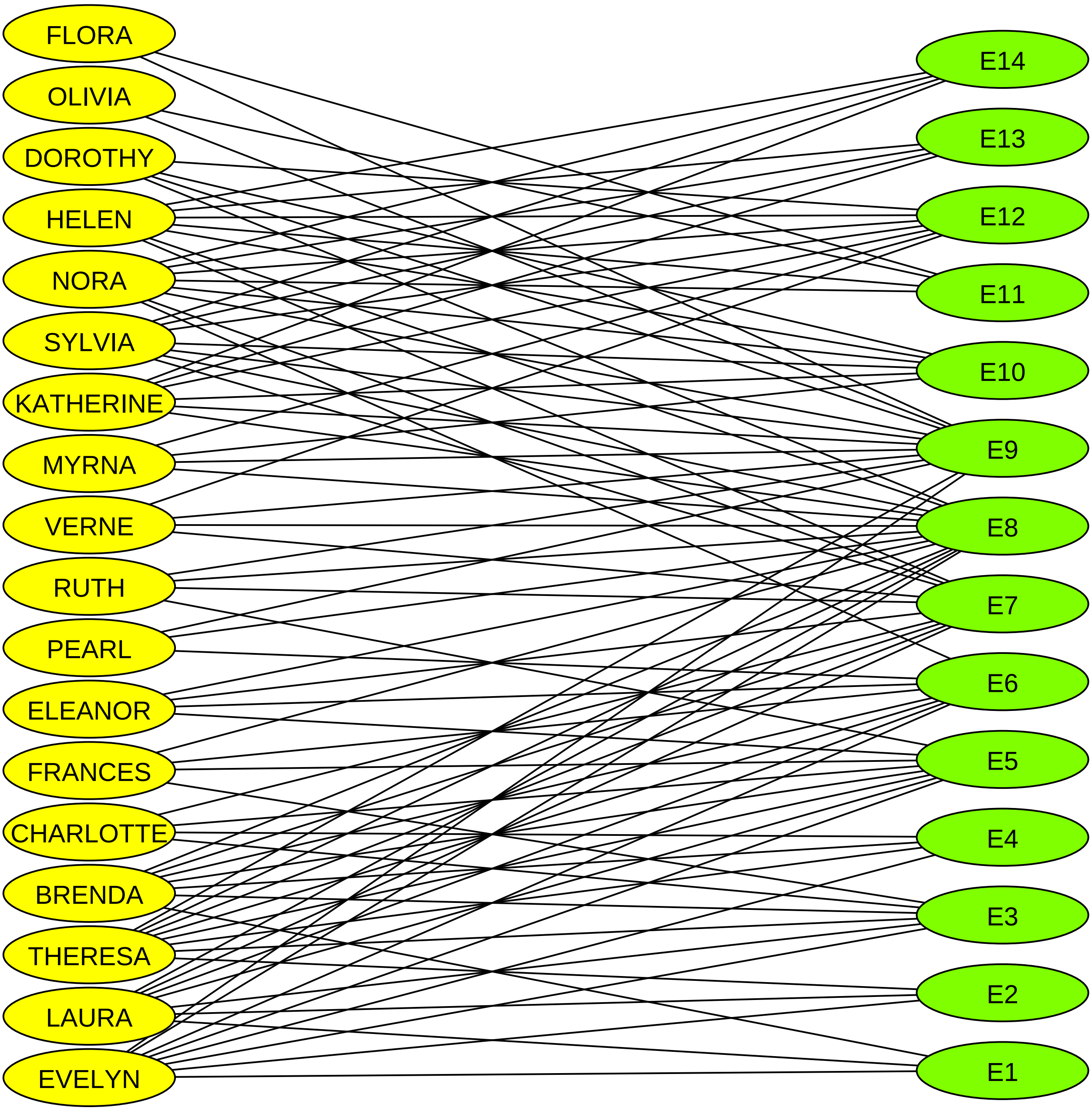
Types of networks

- *undirected network* – relation is symmetric – all lines are undirected – edges, $L = E$.
- *directed network* – relation is not symmetric – all lines are directed – arcs, $L = A$.
- *mixed network* – arcs and edges exist in a corresponding graph – $L = A \cup E$.
- *two-mode network*

  Two-mode network consists of two sets of units (e. g. people and events), relation connects the two sets, e. g. participation of people in social events.

  Corresponding graph is called *bipartite graph* – lines connect only vertices from one two vertices from another set – inside sets there are no connections.

Example: participation of women in social events: davis.net – example from UCINET dataset:

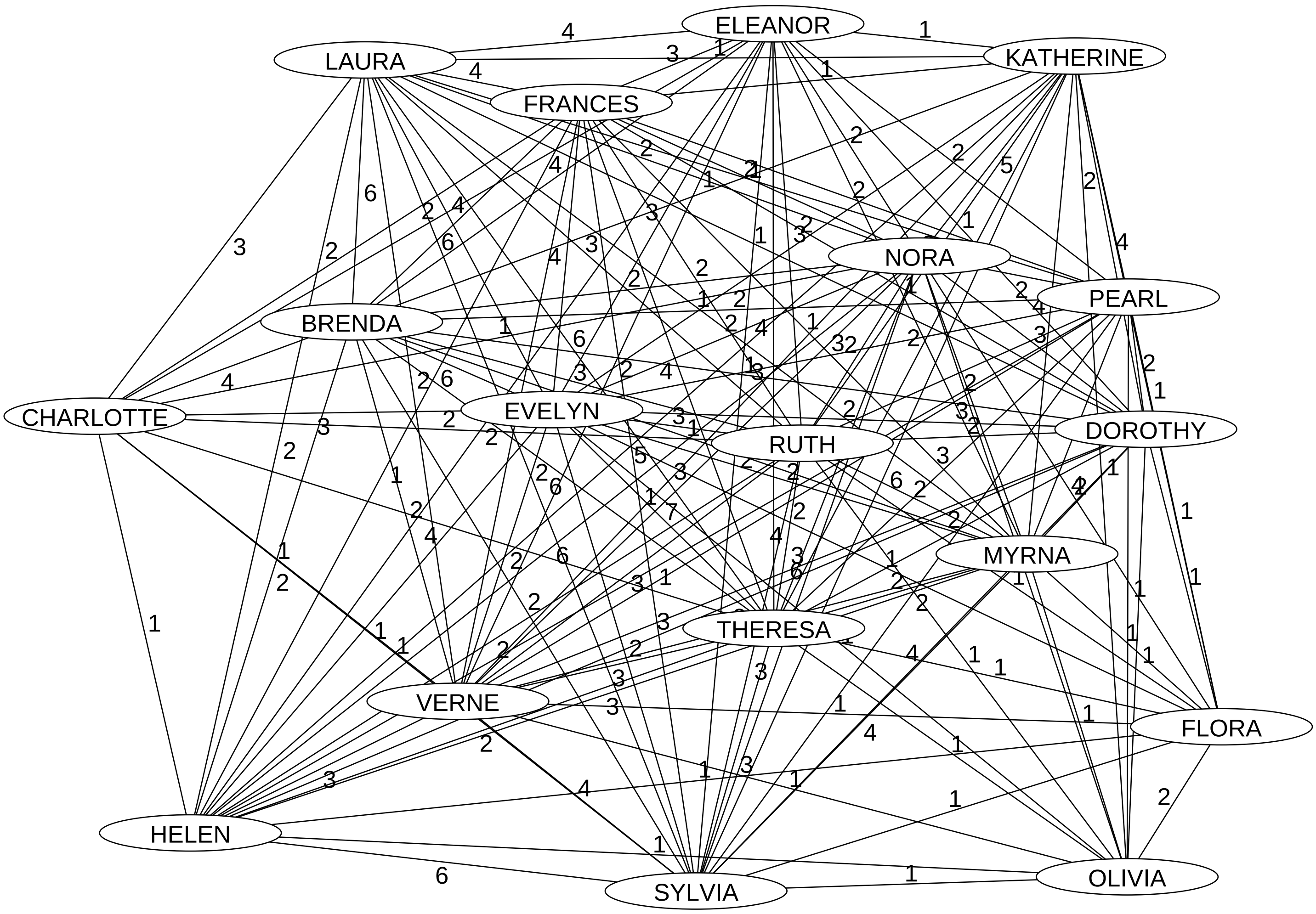


Two-mode network can be transformed to ordinary network, where units of a new network are only units from first or only units from second set.

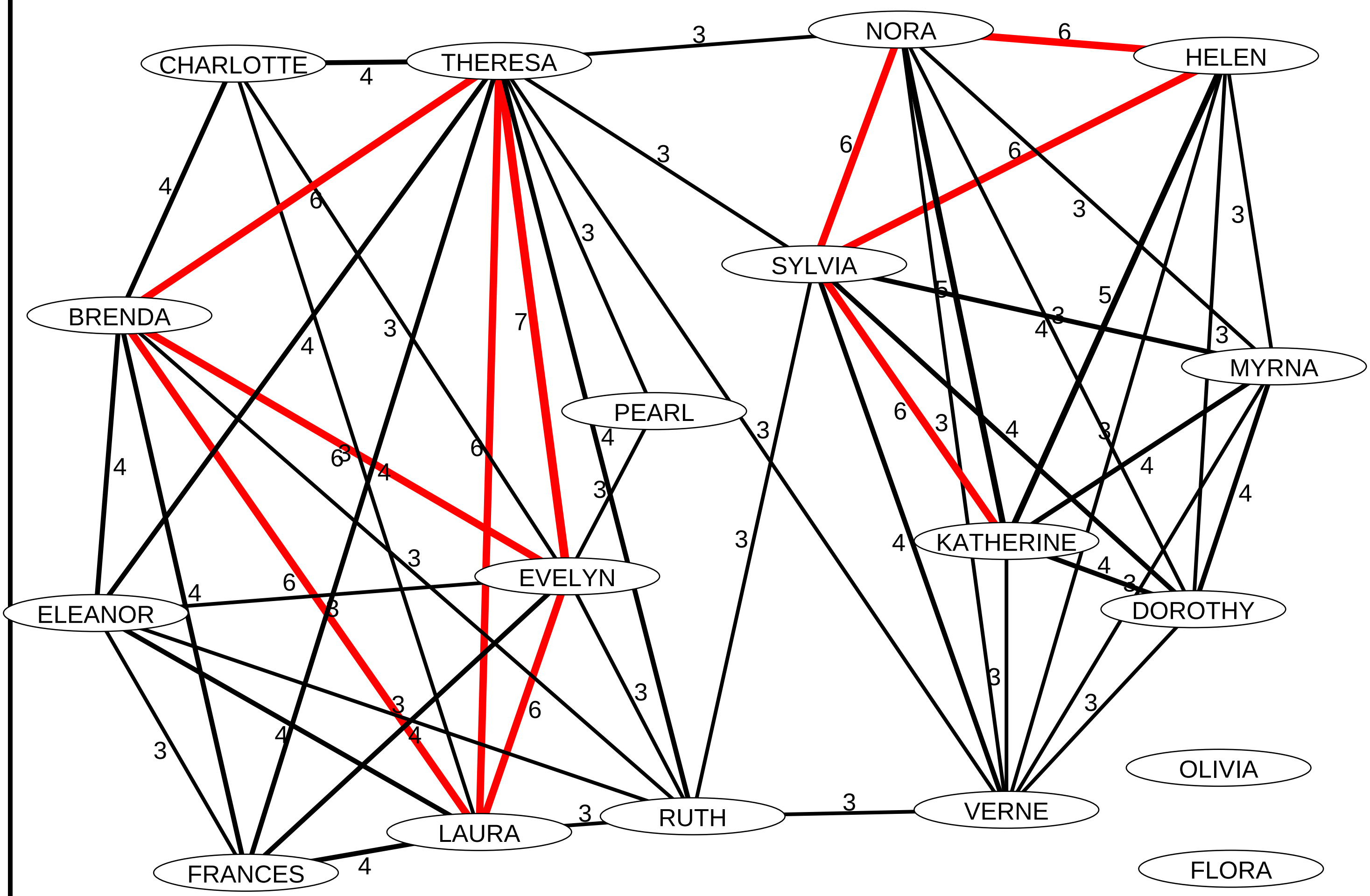
If we transform network from the previous example into network where units are women, we get ordinary network where two women are in relation (in corresponding graph undirected line will exist) if they took part in at least one common event. Value of line is the number of common events.

If we transform network into ordinary network, where units are events, the two events will be in relation (in corresponding graph undirected line will exist), if there exists a women who took part in both events. Value of line is the number of women taking part in the two events.

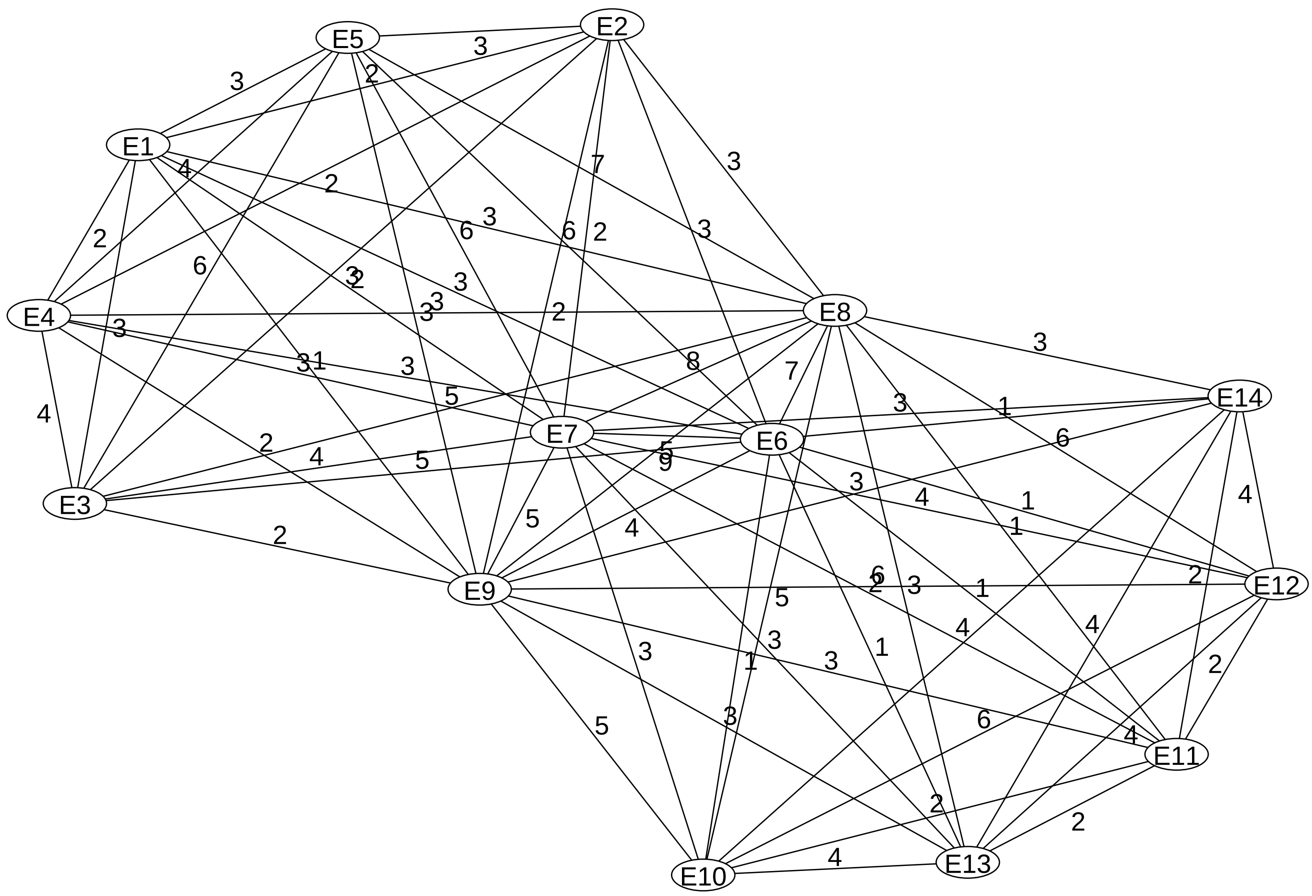
Network of women and number of common events



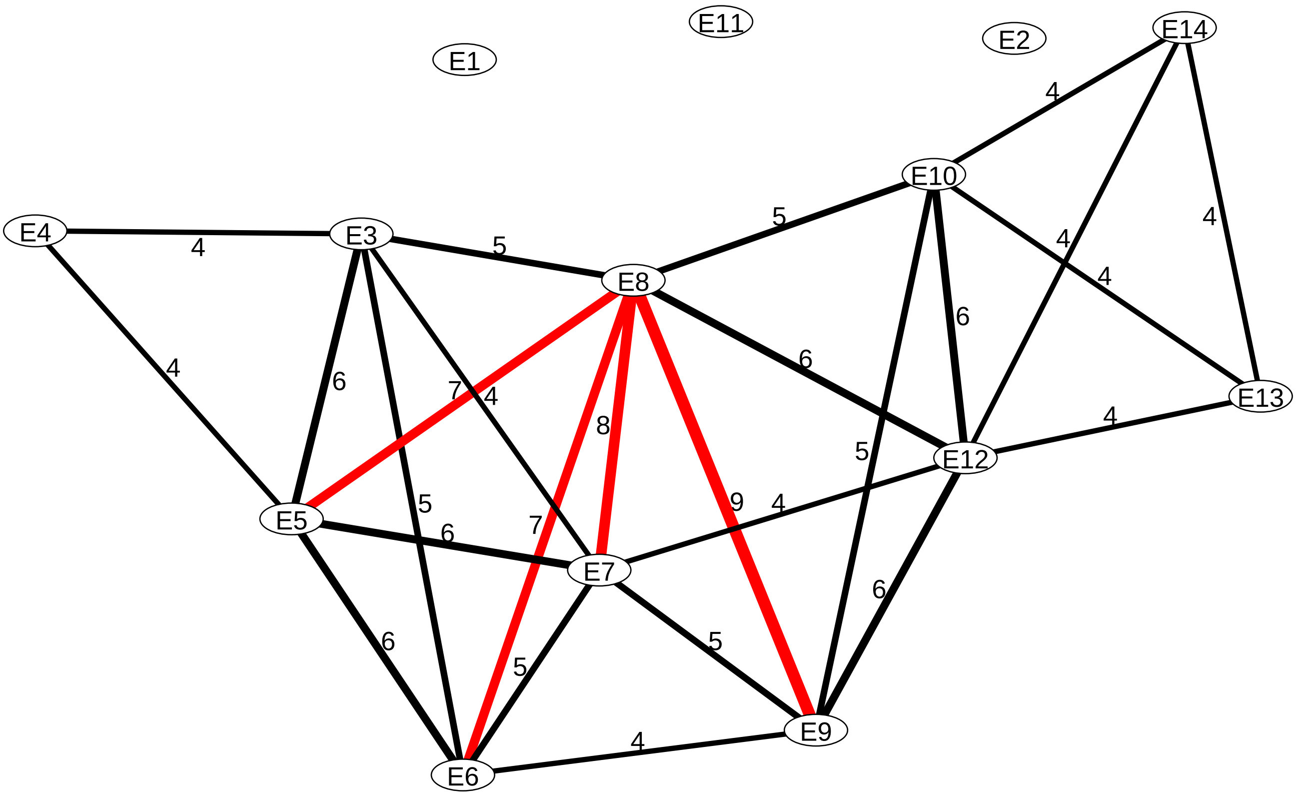
Number of common events at least 3



Network of events and number of women taking part in connected events



Number of common women at least 4



Small and large networks

Networks with some 10 units and lines are called *small networks*, while networks with some 1000 units and lines are called *large networks*.

Dense and sparse networks

Network is called *sparse*, if the number of lines in corresponding graph is of the same order as the number of vertices ($n \approx km$). Large networks that are sparse can still be efficiently analysed with some algorithms. In the real life we often find very large but sparse networks.

In general, number of lines can be much higher than number of vertices. Such networks are called *dense*.

If every unit is connected to every other unit the number of lines is n^2 (number of elements in matrix).

If every unit is connected to every other unit except to itself (graph without loops), the number of lines is $n(n-1)$ (number of elements in matrix without diagonal).

According to that the density of a network can be defined:

For networks with loops:

$$Density1 = \frac{m}{n^2}$$

For networks without loops:

$$Density2 = \frac{m}{n(n-1)}$$

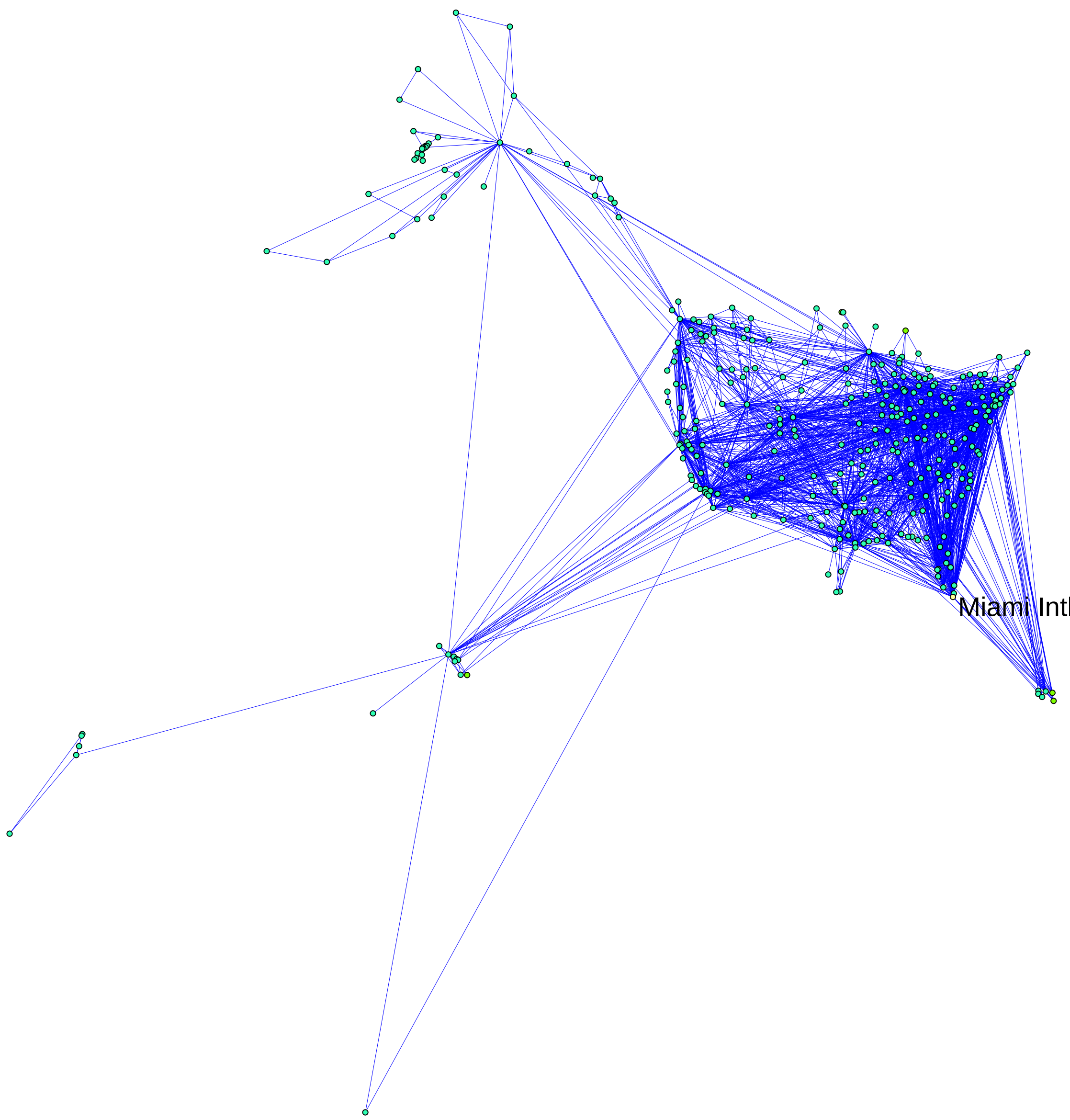
If at most one line can exist among any two vertices the density is a real number between 0 and 1.

Density of a network is one of the measures using which we can compare different networks.

Example of large networks

- social networks
  - connections among people (friendship);
  - relation among political parties;
  - trade among organizations, countries;
  - genealogies;
  - citation networks;
  - computer networks (local networks, Internet, links among home pages);
  - telephone calls;
- flow charts in computer science;
- Petri nets;
- organic molecule in chemistry;
- connections among words in text;
- transportation networks (airlines, streets, electric networks...).

Airlines among 332 American airports (332 vertices, 2116 lines)



Pajek

`Pajek` is a program package for Windows 95/98/NT, which enables analyses of *large networks*. Program is freely available at:
http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Analyses in Pajek are performed using six data structures:

1. network,
2. partition,
3. cluster,
4. permutation,
5. vector,
6. hierarchy.

First look at networks only.

Network

Default extension is .NET

Network can be defined in different ways on input file. Look at three of them:

1. List of neighbours (Arcslist / Edgeslist)

*Vertices 5

1 ”a”

2 ”b”

3 ”c”

4 ”d”

5 ”e”

*Arcslist

1 2 4

2 3

3 1 4

4 5

*Edgeslist

1 5

Explanation:

Data must be prepared in input (ASCII) file. Program *NotePad* can be used for editing. In DOS environment the editor is called *Edit*. Much better of both is a shareware editor *TextPad* (http://www.textpad.com/).

Words, starting with *, must always be written in first column of the line. They are declaring start of definition of vertices or lines.

Using *Vertices 5 we define network with 5 vertices. This must always be the first statement in definition of network.

Definition of vertices follows after that – to each vertex we give a label, which is displayed between ”and ”. If labels of vertices are equal to their sequential numbers this part can be omitted.

Using *Arcslist list of directed lines from selected vertices are declared (1 2 4 means, that there exist two lines from vertex 1, one to vertex 2 and another to vertex 4).

Similarly *Edgeslist, declares list of undirected lines from selected vertex.

At the end of the file no additional empty lines are allowed.

2. Pairs of lines (Arcs / Edges)

*Vertices 5

1 ”a”

2 ”b”

3 ”c”

4 ”d”

5 ”e”

*Arcs

1 2 1

1 4 1

2 3 2

3 1 1

3 4 2

4 5 1

*Edges

1 5 1

Explanation:

This is the most general format. In this case every line is defined separately in new line - initial and terminal vertex of every line are given. Directed lines are defined using *Arcs, undirected lines are defined using *Edges. The third number in rows defining lines gives the value of the line. Lines from 2 to 3 and from 3 to 4 have value 2, all others have value 1.

In the previous format (Arcslist / Edgeslist) values of lines cannot be defined – the format is suitable only if all values of lines are 1.

If values of lines are not important the third number can be omitted (all lines get value 1).

3. Matrix

*Vertices 5

1 ”a”

2 ”b”

3 ”c”

4 ”d”

5 ”e”

*Matrix

0 1 0 1 1

0 0 2 0 0

1 0 0 2 0

0 0 0 0 1

1 0 0 0 0

Explanation:

In this format directed lines (arcs) are given in the matrix form (*Matrix). If we want to transform bidirected arcs to edges we can use
Net/Transform/Arcs to Edges/Bidirected only

Only necessary elements to define network were described so far. Additionally, Pajek enables precise definition of elements used for drawing networks (coordinates of vertices in space, shapes and colors of vertices and lines, ...).

Among internal formats (.NET) Pajek recognizes several other formats: UCINET DL; Vega; GEDCOM and some chemical formats: BS (Ball and Stick), MAC (Mac Molecule) and MOL (MDL MOLfile).

Interactive definition of networks

Simple networks can be defined inside program Pajek as well without definition in input file:

First empty network (network without lines) is built, later lines are added.

Select: Net/Random Network
Option is primarily intended to generate random networks on given number of vertices, where each vertex get some lines to other vertices. The number of lines from each vertex is limited by lower and upper bound.

If the lower and upper number of lines are limited to 0, we get empty network on selected number of vertices.

In our example with 5 vertices we input:

How many vertices: 5
Min number of lines from vertex: 0
Max number of lines from vertex: 0

Then we select Draw/Draw (or press Ctrl+G). The network is represented by an layout in new window. Since coordinates of vertices are not known yet, all vertices are positioned in upper left corner. Vertices can be moved to other position by clicking with left mouse button on selected vertex, holding the mouse button down and moving the mouse.

After that we define for every vertex all lines connected to the vertex: We click with right mouse button on selected vertex. New window is opened, where with double click on Newline vertices that are neighbours to selected vertex are entered.

If arc is going from selected vertex to vertex 2, we input -2, if arc is coming from vertex 2, we enter $+2$, if the two vertices are connected by an edge we enter only 2.

Value of line is entered by clicking with right mouse button on the highlighted selection in the same window and entering the desired value.

Line can be deleted by double clicking with the left mouse button the highlighted selection in the same window with the left mouse button.

The Draw window can be refreshed using command Redraw.

Some commands in Draw window

All commands are described in the html file (Manual) that is available on the Pajek homepage.

- **Options/Mark vertices using** – selecting the way vertices are marked in the picture
- **Options/Lines** – visibility or nonvisibiliy of arcs and edges, selecting the way lines are marked in the picture
- **Options/Size** – selecting size of vertices, size of font, size of arrows and width of lines
- **Options/Colors** – selecting background color, color of vertices, lines, font. . .

Determining layouts of networks

The real imagination of a network is often obtained only by the picture of it. Several automatic and hand drawing routines are included in program Pajek.

Automatic layout generation

- **Energy** – Idea: network is represented like a physical system, we are searching for the state with minimal energy. Two algorithms are included:
  - **Energy/Kamada-Kawai** – slower
  - **Energy/Fruchterman-Reingold** – faster, drawing in plane or space (2D or 3D), selecting the repulsion factor

  Using **Options/Interrupt** – we select how many time (in seconds) the picture can be optimized.
- **EigenValues** – Selected 2 or 3 eigenvectors are computed and this eigenvectors become the coordinates of vertices. Nice pictures are obtained usually if there are some symmetricity in network (social networks are not of that type, usually).

Manual drawing

We can move the vertex by clicking with left mouse button on selected vertex and moving the mouse.

Pictures in space: picture can be rotated by pressing (holding down) keys x, X, y, Y, z, Z (the key stands for axis of rotation, small/capital letters mean positive or negative orientation).

Any axis of rotation can be selected using **Spin/Normal**. After that, rotation around the selected axis is executed by holding down s or S.

The part of the network can be zoomed by selecting the part of the picture using the right mouse button. The complete picture is obtained using Redraw.

If we want to save the changes of the network (in our case the picture), we select the icon for saving network or **File/Network/Save**, choose the type of presentation and the name of the file. Types of presentation were defined in the beginning (input formats). If the name is equal to the name of the network that already exists, the old network is lost.

If we select Arc/Edges presentation, the new file will look like (the difference is only that coordinates of vertices are added):

```
*Vertices 5
1 "a"    0.1672   0.3272   0.5000
2 "b"    0.2029   0.7394   0.5000
3 "c"    0.6144   0.8334   0.5000
4 "d"    0.8328   0.4794   0.5000
5 "e"    0.5565   0.1666   0.5000
*Arcs
1 2 1
1 4 1
2 3 2
3 1 1
3 4 2
4 5 1
*Edges
1 5 1
```

Output formats

The obtained pictures in plane or space, can be stored in output formats, that are suitable for including pictures in text processors (e. g. Word) or examining using special viewers. Saving in output formats is available in option **Export**. The possible formats are:

- Bitmap (BMP) – Picture can be included in Word (Insert Picture). It is just a snapshot of the picture – can be quite large in size.
- EPS – Picture can be shown using program GSView or included in text editors (e. g. LaTeX). To some extend picture can be included to Word too.
- VRML - Virtual Reality Modeling Language – for pictures in space. VRML is recognized by Internet browsers like Netscape and Explorer using special plug-in CosmoPlayer:
  http://cosmosoftware.com/
  The main advantage of the representation is that the layout is not statical but dynamical – we can travel in the picture, rotate the picture, select different views…
- MDL MOLfile (Chime viewer)

http://www.mdli.com/download/chimedown.html
The format is primarily intended for space pictures of molecule, but can be used for drawing undirected networks too.

- Kinemages (*KINEmatic iMAGES*)
  Freely is available program Mage:
  http://www.prosci.org/Kinemage/MageSoftware.html
  which is more powerful than Chime – simple animation can be done too.

Basic information about network

Basic information about network can be obtained by

Net/Info/General

which is available in the main window of the program. We get

- number of vertices
- number of arcs
- number of directed loops
- number of edges
- number of undirected loops
- both densities of lines ($Density1$ in $Density2$)

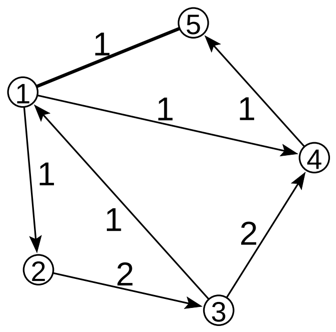
Additionally we must answer the question:

Number of arcs/edges with highest/lowest value

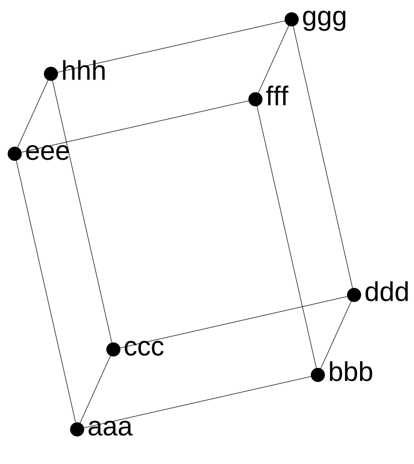
where we enter the number of lines with the highest/lowest value that we want to output. If we enter 10, 10 lines with the highest value will be displayed. If we enter -10, 10 lines with the lowest value will be displayed.

Examples

1. Build the following network interactively in Pajek:



2. Prepare the input file ex1.net with the description of network in the shape of cube, with the labels of vertices: aaa, bbb, ccc, ddd, eee, fff, ggg and hhh: Read the network, draw it in plane and space, and save it again.



3. Read the network that is stored in the file ex2.net and try all algorithms for automatic layout generation.