Triads

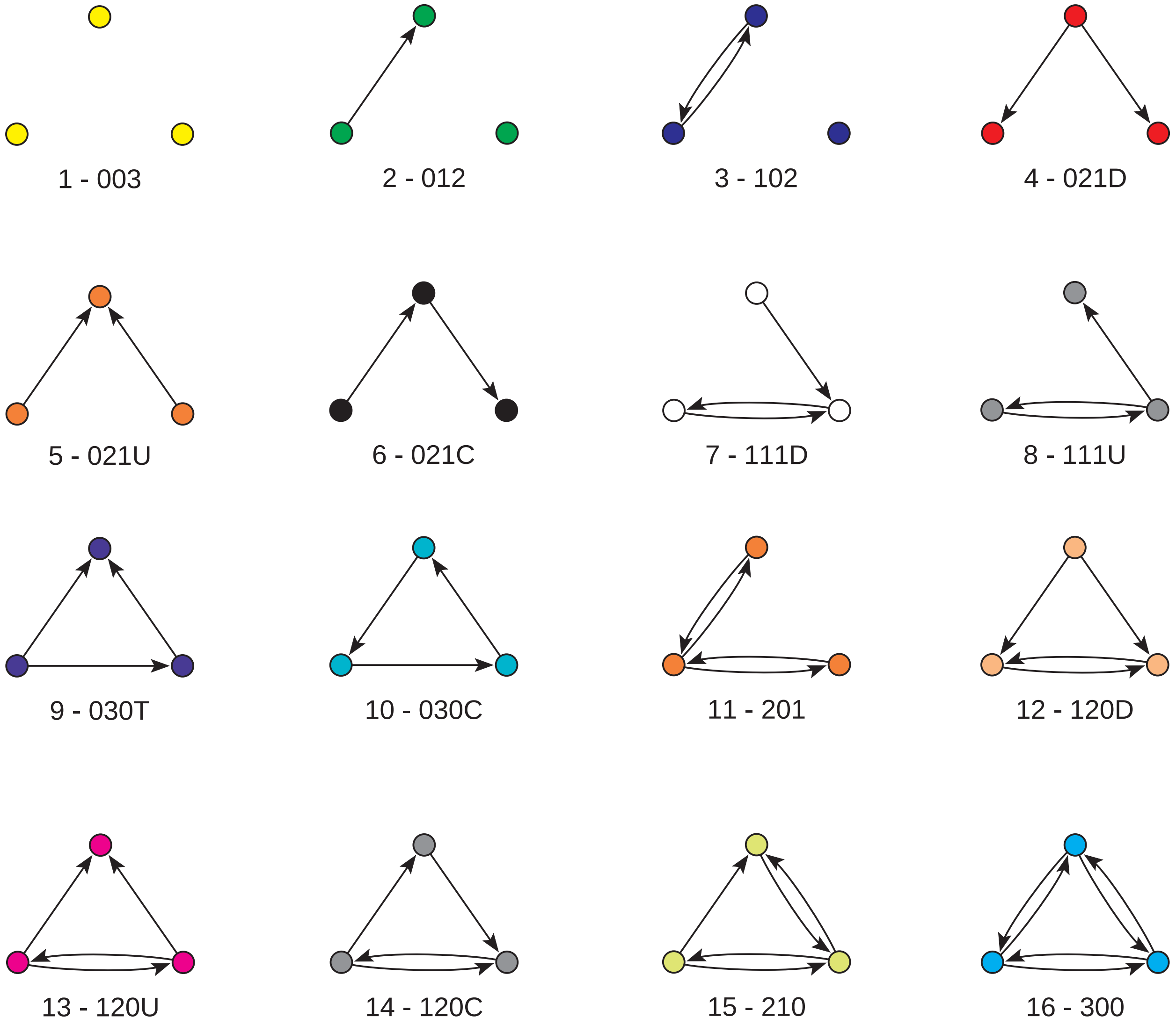
If we select three vertices and the lines among them in the directed network we always get one of the 16 combinations. The subgraph on three vertices is called a *triad*.

There exist two different labeling of triads – first is simple numbering 1..16, the second type of label consists of three numbers xyz, where:

- x – number of pairs of vertices connected by bidirected arcs;
- y – number of pairs of vertices connected by a single arc;
- z – number of unconnected pairs of vertices.

When the three numbers are not enough to distinguish among triads additional letter is used:
D-Down, **U**-Up, **C**-Cyclic, **T**-Transitive.

All possible triads



(In)transitivity of triads

The *triad*, involving units u, v and w is ***transitive***, if whenever exist arcs $u \rightarrow v$ and $v \rightarrow w$, exists also $u \rightarrow w$.

---

triads $\boxed{9, 12, 13, 16}$ are *transitive*

triads $\boxed{6, 7, 8, 10, 11, 14, 15}$ are *intransitive*

triads $\boxed{1, 2, 3, 4, 5}$ do not contain arcs to meet the conditions of transitivity (they are vacuously transitive)

Some intransitive triads are *'more intransitive'*, some are *'less intransitive'*, e. g.
in triad 15 only one arc is missing to be transitive
in triad 11 two arcs are missing

A directed network is transitive, if every triad in it is transitive.

We can count all triads in Pajek using
Network/Info/Triadic Census.

In network exclique.net there are 480 transitive, 638 intransitive triads, but for most triads (2942) transitivity cannot be checked.

Cliques

Subset of vertices in a network is called a ***clique***, if every vertex from the subset is connected to all other vertices in the subset (clique is a special type of a ***core***). A clique (in social networks) represent a subset of persons who are connected as much as possible.

Searching for cliques is computationally much more expensive than searching for cores. Therefore we will only search for cliques of size 3 or 4 at most in smaller networks.

3-clique is triad number 16. Using counting of triads we can find how many 3-cliques exist in a network.

But if we want to find all occurrences of 3-cliques we must use the general procedure for searching fragments in network.

This procedure finds any interesting smaller network (fragment) in a larger network.

In our case the fragment is a clique on 3 vertices.

The 3-clique can be generated in Draw window like explained in the beginning (interactive definition of networks).

But it is easier to generate a directed complete network on 3 vertices:
Network/Create New Network/Complete Network/Directed
We define this network as a fragment by selecting it as the first network.

Then we select the original network as the second network.

Be careful: Algorithm for searching fragments *distinguishes* between edges and bidirected arcs. If we have some edges in the network, we must transform them to bidirected arcs first using
Network/Create New Network/Transform/Edges $->$ Arcs.

After we selected fragment (first network) and original network, where we want to find such fragments (second network), we run searching for fragments using Networks/Fragment (First in Second)/Find.

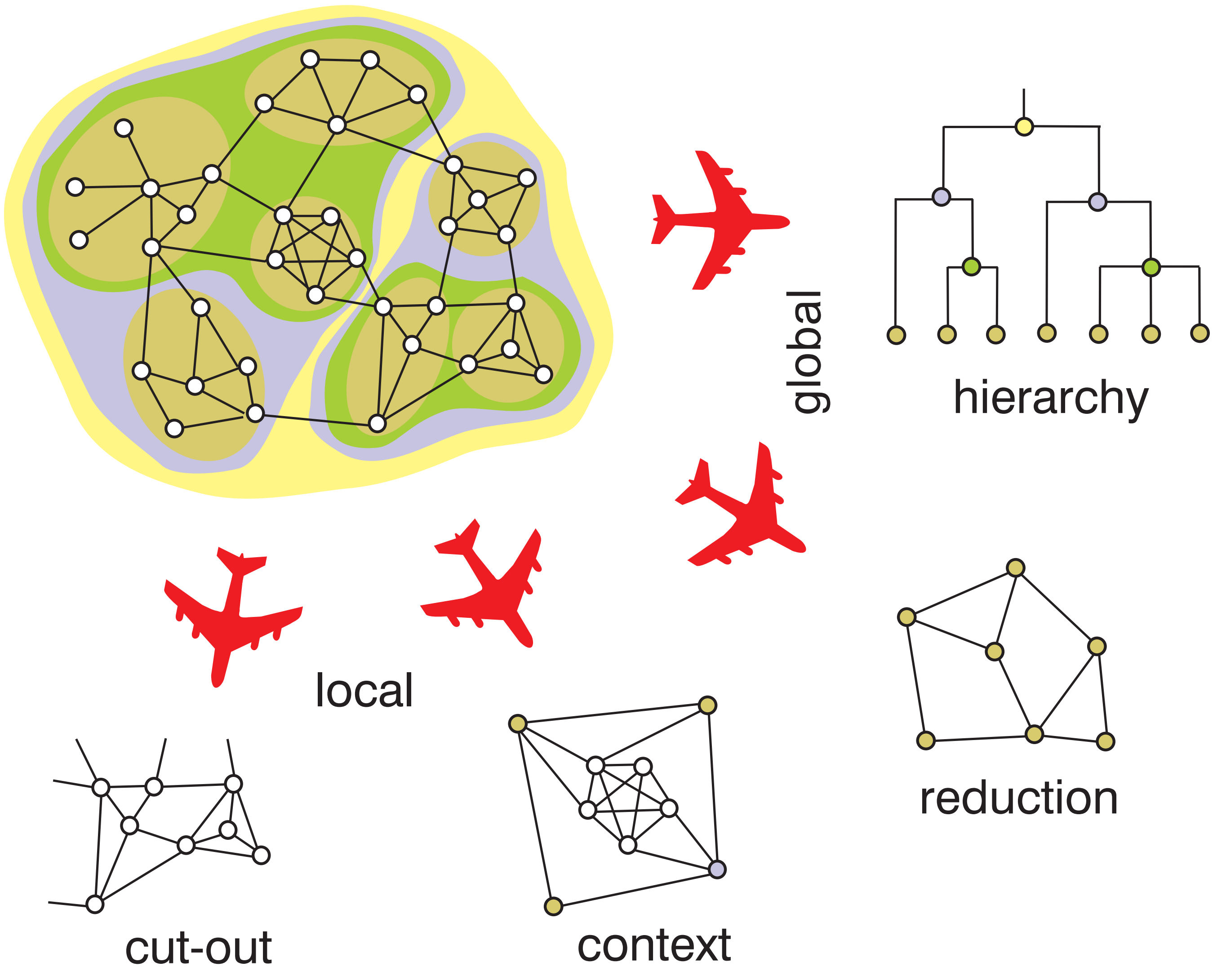
We get three objects as a result:

1. Subnetwork containing only all occurrences of fragments (cliques).
2. Partition with values 0 and higher: value 0 means that the correspondent vertex does not belong to any fragment (clique), value a means that vertex belongs to a fragments (cliques).
3. Hierarchy with all fragments.

If original network is small (and not too dense), we can try to find 4-cliques in a similar way.

Examples: advice.net, exclique.net.

Global and local views



We can look at network as a whole (*global view*) or we examine only a part of it in details (*local view*).

Data: Shr1.net, shr.clu.

Global views

There exist two global views: *reduction* and *hierarchy*.

Reduction means that we consider all vertices in a cluster as a new vertex – all vertices in a cluster are shrunk to one compound vertex. Even more general global view is hierarchy – vertices in network are hierarchically clustered to larger and larger clusters and in this way we get the tree (hierarchy). Later we walk in the tree and according to our position see the network in more details (if we are close to the leaves of the tree) or less details (if we are close to the root of the tree) – vertices, which belong to the same cluster in the tree are shrunk to new, compound vertex.

Local views

There exist two local views as well. The first is called *cut-out* – a smaller part of the network is extracted from the network and we consider it separately. The second local view is called *context* – we extract a smaller part (cluster) from the network, but we show at the same time, how the extracted part is connected to the rest of the network.

Context is combination of local and global view: one cluster is shown in details, vertices in other clusters are shrunk.

Example: Import and export among countries

Import and export in 1994 among 80 countries are given. Import/export is given in 1000$, but imports from countries which are lower than 1% of the total import of a country are not considered. File: import.net.

Two partitions are given:

- partition according to continents – (cont.clu).
  1 – Africa, 2 – Asia, 3 – Europe, 4 – North America, 5 – Oceania, 6 – South America.
- partition according to development (develop.clu)
  1 – highly developed countries, 2 – developed, 3 – less developed, 4 – undeveloped, 999998 – unclassified.

Local views: select a continent and look at import/export among countries in this continent.

Global view: import/export among and inside continents.

Context: Detailed view of import/export among countries from one continent and their trade to other continents.

Global and local views in Pajek

Local view – extracting selected cluster (already known from previous chapters):

Operations/Network+Partition/Extract SubNetwork.

Matrix shadowing in EPS:

File/Network/Export as Matrix to EPS/Original

Global views are obtained by shrinking:

Operations/Network+Partition/Shrink Network.

Pajek asks for minimum number of lines, that must exist among shrunk vertices that will generate a line among shrunk vertices. We can use the default value (1). After that we are asked, which cluster we do not want to shrink. If we want to shrink all clusters, we input number of cluster that does not exist (in our case for example 0, or 7, or . . .).

Similarly the ***context*** is obtained: we select the cluster number that we do not want to shrink.

Matrix shadowing in EPS:

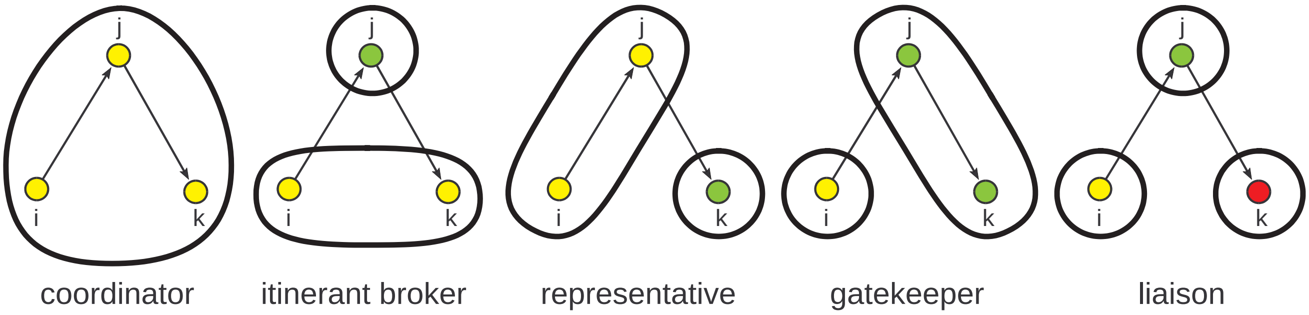
File/Network/Export as Matrix to EPS/Original

In the case of network of students local views can represent subnetworks of only boys or only girls, global view the communication among this two groups.

Brokerage roles

In a network in which groups are defined (e.g. men and women) each unit can be involved in the following brokerage roles:

- coordinator,
- itinerant broker,
- representative,
- gatekeeper,
- liaison.



In Pajek, first the appropriate partition must be defined (e.g. according to gender) using: Network/Create Partition/Vertex Shapes. We can check the result using Draw/Network + First Partition. Then use the command: Operations/Network+Partition/Brokerage Roles to compute the brokerage roles. As a result we get five vectors (one vector for each role).

Examples

1. Count all types of triads in network exclique.net.
   Are there many transitive triads?
   Find all cliques of size 3 and of size 4.

2. Construct all views on import/export among countries network:

   - (sub)network for one continent,
   - shrunk network representing import/export among continents,
   - contextual view: import/export of countries from one continent to other continents

   Use networks import.net and partition according to continents cont.clu.