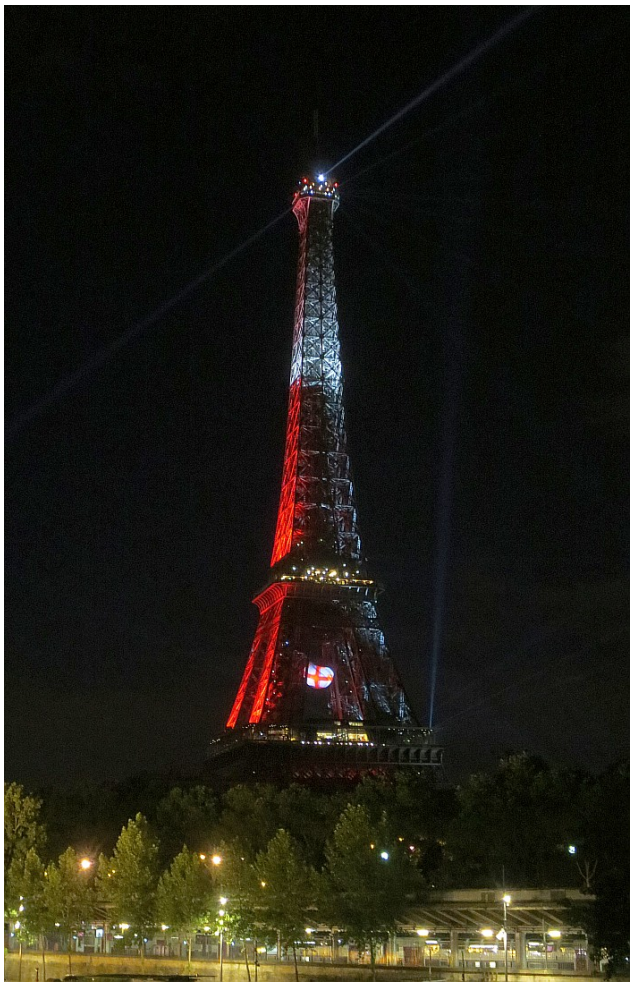
On the widespread index

Vladimir Batagelj

IMFM Ljubljana and IAM UP Koper

Second European Conference on Social Networks
June 14-17, 2016, Paris

Outline



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (June 17, 2016 at 08 : 40):
EUSN'16 slides PDF

Problem

January 12, 2016

```
Subject How to measure the distribution of an attribute among the
        nodes of a network?
From LEVALLOIS Clément
Sender Social Networks Discussion Forum
To SOCNET@LISTS.UFL.EDU
Date Tue 11:36
-----------------------------------------------
Dear List members,

I need to have a measure of how widespread is the distribution of a
node attribute in a network. Let me explain:

My nodes have a textual attribute, let's say "preferred flavor for ice
cream"

I would like to know to what extent the flavor "raspberry" is a value
which is evenly distributed in the network, or to the contrary, just
found in one community. A low value would mean that only nodes from a
subregion of the network have this taste, a higher value would show an
even distribution of the value across the whole network. I imagine that
a difficulty is to account for the frequency of the attribute: if many
nodes of the network have "raspberry" for the value of the attribute,
it will tend to make this value distributed more widely.

Any help or pointer on this would be very much appreciated!

Thank you, Clement Levallois
```

Problem

In my reply I proposed the following:

A possible measure could be the following:
let V be the set of nodes and S the set of nodes with given attribute value. Then we define the widespread of attribute as

$$W(S) = \frac{|S \cup N(S)|}{|V|}$$

where $N(S)$ is the set of nodes neighboring some node of S. $||$ denotes the cardinality (number of elements) of the set.

After some thoughts, in my second message, I proposed some variations on this widespread measure.

Problem

1. variant – consider also the size of set S

$$W'(S) = \frac{|S \cup N(S)|.|S|}{|V|^2}$$

It attains its maximum value 1 iff $S = V$.

2. variant – dominating node

$$W''(S) = \frac{|S \cup N(S)|.|V \setminus S|}{|V|.(|V| - 1)}$$

It attains its maximum value 1 iff S is the center of a star (a single node linked to all other nodes).

Problem

```
> Indeed my formulation was not clear. Refining my statement, I think
> a possible solution can appear:
>
> - Being evenly distributed in the network would mean that the
> distance (shortest paths) between the nodes bearing this attribute
> value iscomparably close to the distance between the same number of
> nodes randomly picked from the entire set of nodes of the network.
>
> Does it make sense? 2 things:
> - it does not depend on a notion of communities
> - I might be wrong but the formulation above seems quite
> computationally intensive
>
> Clement
```

If S is a small community then usually $N(S)$ will have large intersection with S and $N(S) \setminus S$ will be relatively small – the value of $W(S)$ will be small.
We get large values of $W(S)$ when S is large or S contains hubs – nodes with very large degree.
It is very fast. The time complexity is linear in number of links.

Simple widespread index

Let us show how the proposed widespread indices can be computed in Pajek.
Let S be a selected subset of set of nodes V in a simple (no parallel links) directed network $\mathbf{N} = (V, L)$. With $N(S)$ we denote the set of neighbors of the set S:

$$N(S) = \{u \in V : \exists v \in S : (v, u) \in L\}$$

and with $N_+(S) = S \cup N(S)$. With $n = |V|$ we denote the number of nodes.
A *simple widespread index* W_0 is defined as

$$W(S) = \frac{|N_+(S)|}{n}.$$

Simple widespread index

D is a *dominating set* of a network $\mathbf{N} = (V, L)$ iff $N_+(D) = V$.

D is an *independent set* of a network $\mathbf{N} = (V, L)$ iff $D \cap N(D) = \emptyset$.

We have:

- $0 \leq W(S) \leq 1$.
- $W(V) = 1$.
- $W(S) = 1$ iff S is a dominating set.
- if $S_1 \subset S_2$ then $N_+(S_1) \subseteq N_+(S_2)$
- $|N_+(S_1)| \leq |N_+(S_2)|$ iff $W(S_1) \leq W(S_2)$.

Domination number

Related to dominating sets is a *domination number* $\gamma(\mathbf{N})$

$$\gamma(\mathbf{N}) = \min\{|D| : D \text{ is a dominating set of } \mathbf{N}\}$$

for which it holds $n \geq \gamma(\mathbf{N}) \geq \lceil \frac{n}{1+\Delta} \rceil \geq 1$, where Δ is the largest (out)degree in $\mathbf{N}$.

Even better lower bound is $\gamma(\mathbf{N}) \geq k$, where k is the smallest number such that

$$\sum_{i=1}^{k} (1 + d_i) \geq n$$

where $(d_i)_i$ is a sequence of outdegrees ordered in decreasing order.

Domination index

The problem with the definition of $W(S)$ is that it doesn't consider the size of the set S. Let D^* be a minimal dominating set. Then an alternative widespread measure could be the *domination index* defined by

$$W^*(S) = \frac{|N(S) \setminus S|}{|V \setminus D^*|} = \frac{|N(S) \setminus S|}{n - \gamma}.$$

If $L = \emptyset$ we set $W^*(S) = 0$. It is easy to see that

- $0 \leq W^*(S) \leq 1$.
- $W^*(V) = 0$.
- in a weakly connected network: $W^*(S) = 1$ iff S is a minimal dominating set.
- if $|N_+(S_1)| = |N_+(S_2)|$ and $|S_1| < |S_2|$ then $W^*(S_1) > W^*(S_2)$.

Unfortunately the problem of determining the domination number $\gamma(\mathbf{N})$ is NP-complete – there is no efficient algorithm to compute W^*.

Domination k-index

To get an efficiently computable index we could replace γ with 1 (it always holds $1 \leq \gamma$); or, even better, with k (also $k \leq \gamma$):

$$W_k(S) = \frac{|N(S) \setminus S|}{n - k}.$$

For this *domination k-index* it holds:

- $0 \leq W_k(S) \leq 1$.
- $W_k(S) = 1$ iff S is a minimal dominant and independent set with k nodes.
- $W^*(S) \geq W_k(S)$ and $W^*(S) = W_\gamma(S)$
- $W^*(S_1) > W^*(S_2)$ iff $W_k(S_1) > W_k(S_2)$
- if $|N_+(S_1)| = |N_+(S_2)|$ and $|S_1| < |S_2|$ then $W_k(S_1) > W_k(S_2)$.

Domination k-index

Note: In a network with non-empty set of arcs the nodes with zero indegree are all in any dominant set. Let D_0 be the set of all such nodes. Then we get a better lower bound for γ – it is $|D_0| + k'$, where k' is the smallest number such that

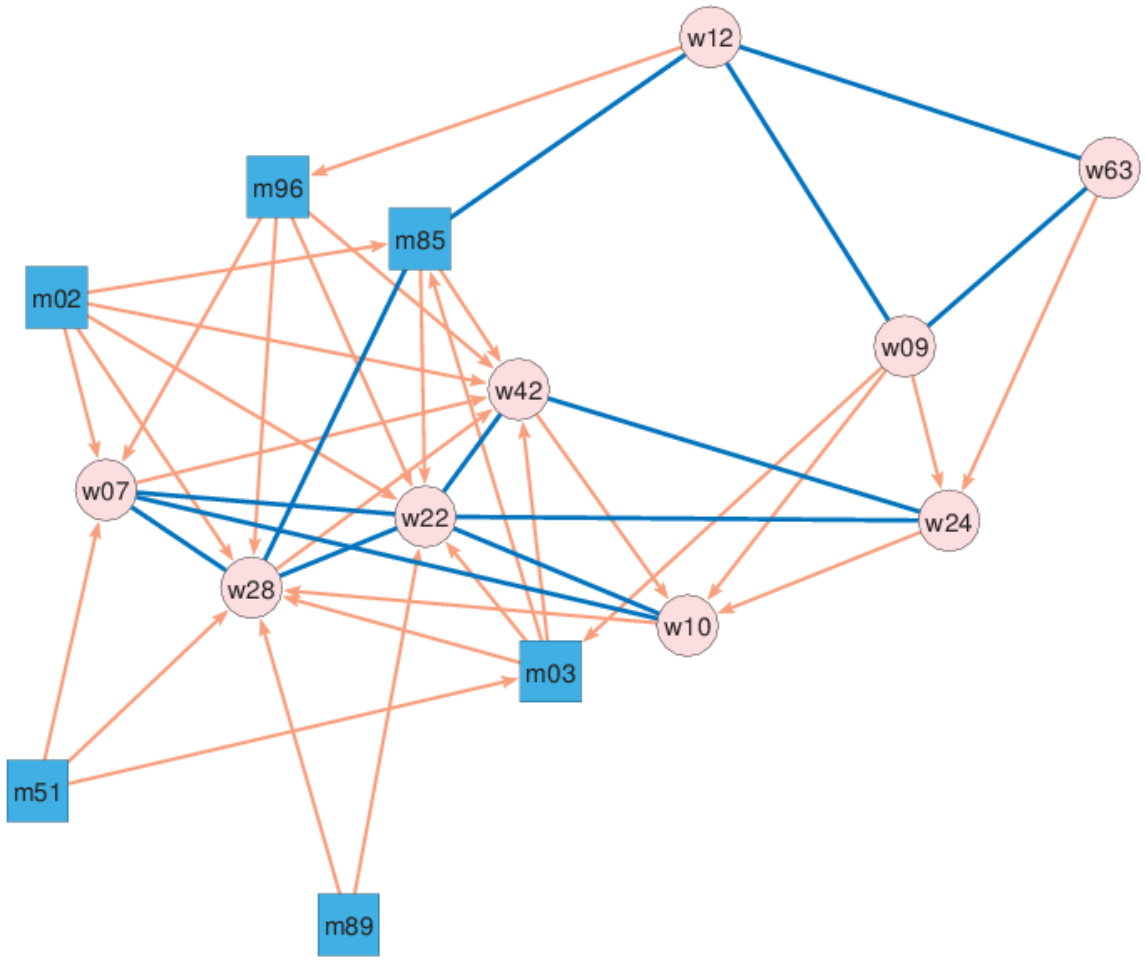
$$\sum_{i=1}^{k'}(1 + d_i) \geq n - |N_+(D_0)|.$$

Since $V \setminus D_0$ can have zero degree nodes again, we iterate the process. The final d_is are computed in the final $V \setminus D_0$.

In some real life networks we have many "leaves" – nodes with indegree 1 and outdegree at most 1. For such nodes there always exists a minimal dominant set that contains their twin node – "roots".

Pajek macro for computing both widespread indices

We will illustrate the computation on the case of the subset of boys in the class network.



Partition

In Pajek we first read the network

```
File/Network/Read [class.net]
```

We get $k = 3$. It is easy to see that $\gamma = 5$.
Next we partition the node set to boys and girls according to the node shape (square - boy; circle - girl).

```
Network/Create Partition/Vertex Shapes
Partition/Binarize Partition [1]        % 1=boy, 2=girl -> 0=girl, 1=boy
```

Clicking on the Info button for partition we learn that the group 1 contains 6 boys, and the group 2 contains 9 girls.

Let $V = \{v_1, v_2, v_3, ..., v_n\}$. We assign to its subset $S \subseteq V$ the corresponding characteristic vector $\chi(S) = [h_1, h_2, h_3, ..., h_n]$ where $h_i = 1$ if $v_i \in S$, and $h_i = 0$ otherwise.

Computing indices W and W_k

Assume that we have active in the registers the network, the partition S and the scalar k.

```
Network/Create New Network/Transform/Transpose 1-Mode [yes]
Partition/Copy to Vector
Operations/Network + Vector/Network*Vector [1,OK]
Vector/Make Partition/by Intervals/Selected Tresholds [0.5]
Vector/Create Scalar/Number                          % n
Partition/Binarize Partition [2]                     % N(S)
select partition S  as Second
Partitions/Max(First,Second)
Partition/Copy to Vector
Vector/Create Scalar/Sum
select scalar n as Second
Vectors/Divide (First/Second)
File/Vector/Change Label [W]
select partition S as First
Partition/Binarize Partition [0]
select partition N(S) as Second
Partitions/Min(First,Second)
Partition/Copy to Vector
Vector/Create Scalar/Sum                             % |N(S)-S|
select scalar n as First
select scalar k as Second
Vectors/Subtract (First-Second)                      % n-k
select n-k as Second
select |N(S)-S| as First
Vectors/Divide (First/Second)
File/Vector/Change Label [Wk]
```

Results

This sequence of commands is saved as the macro `widespread`. It expects as "inputs" a network, a subset S given as a binary partition (characteristic vector), and a scalar k. It returns both indices W and W_k (ZIP).

For the `class` network we have $n = 15$, $k = 3$ and $\gamma = 5$.

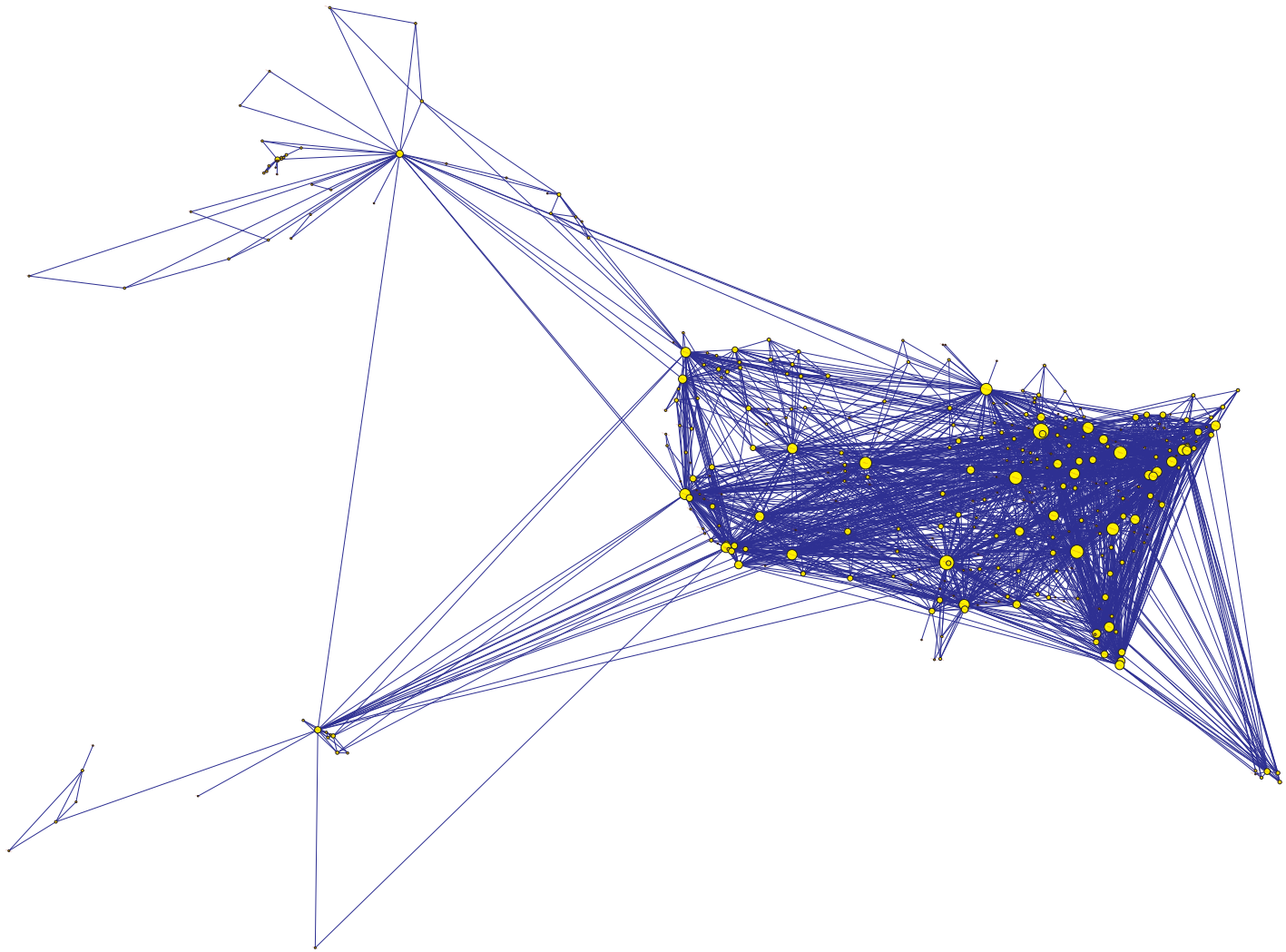
S	*Boys*	*Girls*
$\lvert S \rvert$	6	9
$\lvert S \cup N(S) \rvert$	11	12
$\lvert N(S) \setminus S \rvert$	5	3
$W(S)$	$\frac{11}{15} = 0.73333$	$\frac{12}{15} = 0.8$
$W_3(S)$	$\frac{5}{12} = 0.41667$	$\frac{3}{12} = 0.25$
$W^*(S)$	$\frac{5}{10} = 0.5$	$\frac{3}{10} = 0.3$

US Airports

As a non-toy example, let us consider the US Airports network. It consists of 332 airports and 2126 edges among them. There is an edge linking a pair of airports iff in the year 1997 there was a flight company providing flights between those two airports.

For this network it turns out that using the second approach mentioned in the note we can relatively easy determine its domination number γ.

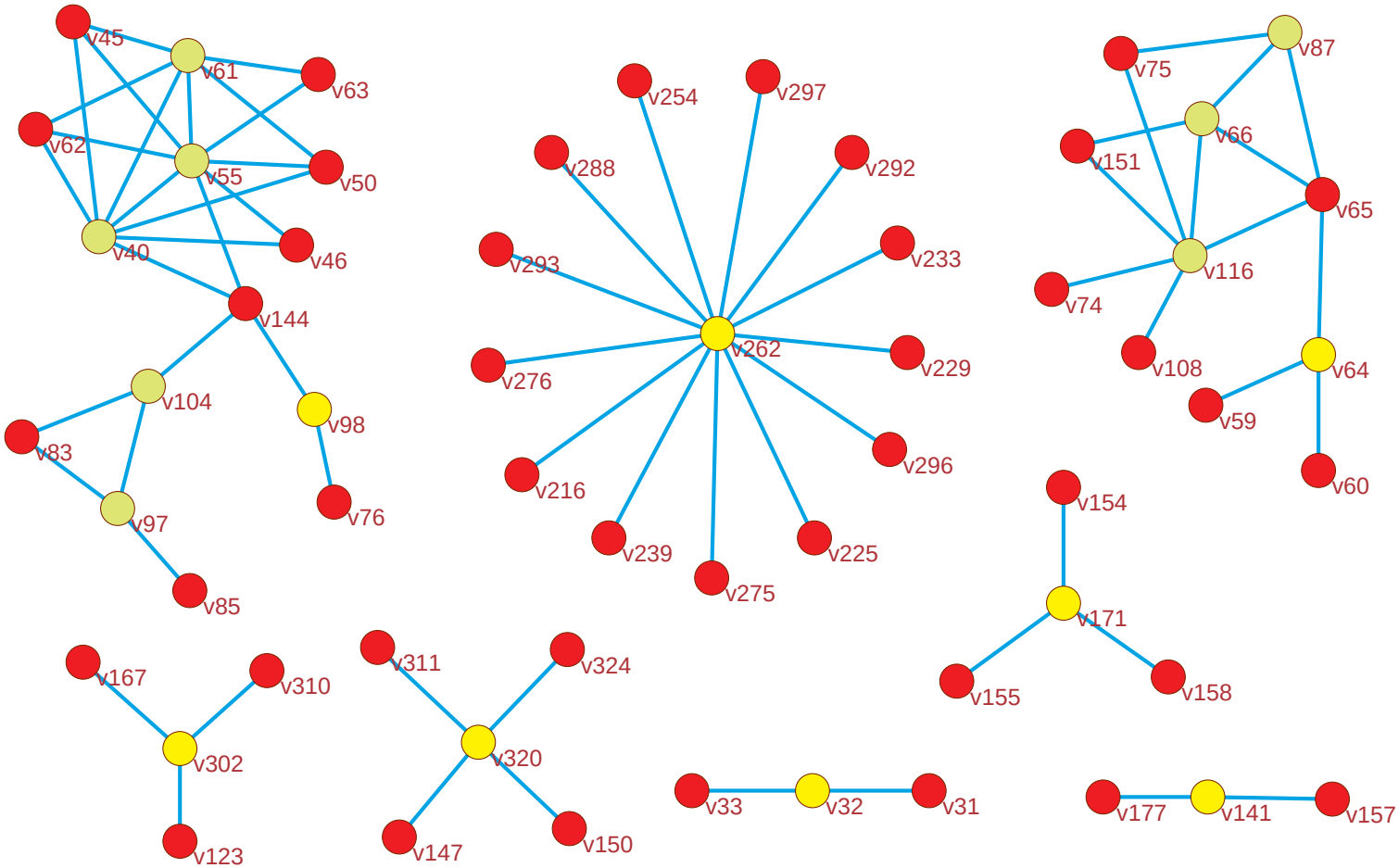
US Airports links 1997



Determining the domination number γ

```
read network USair97.net
Network/Create new network/Transform/--/set all values to 1
Network/Create new network/Transform/.../default labels
Network/Create Partition/Degree/All
Partition/Binarize [1]                % Leaves
Partition/Copy to vector
Operations/Network+Vector/Network*Vector [1]
Vector/Make Partition/by Intervals/Selected Thresholds [0.5]  % r
Partition/Binarize [2]                % Roots
Partition/Copy to Vector
Operations/Network+Vector/Network*Vector [1]
select r as the second vector
Vectors/Add (First+Second)
Vector/Make Partition/by Intervals/Selected Thresholds [0.5]
Partition/Binarize Partition [1]   % Outsiders
Partition/Copy to Vector
Operations/Network+Vector/Network*Vector [1]
Vector/Make Partition/by Intervals/Selected Thresholds [0.5]
Partition/Binarize Partition [2]   % OutNeighbors
select Outsiders as the first and second partition
Partition/Add (First+Second)
select  OutNeighbors as the second partition
Partition/Add (First+Second)
Operations/Network+Partition/Extract [1-*]
Operations/Network+Partition/Transform/Remove lines within clusters [1]
draw network+partition, Kawada-Kawai/Separate components
```

Outsiders



Determining the domination number γ

To determine a minimal domination set D^* we have to add to the set of Roots a minimal set D_o that covers (a node is covered iff it is in the set or is a neighbor of some node from the set) the set of Outsiders (green or yellow). The red nodes are neighbors of Outsiders that are covered by Roots.

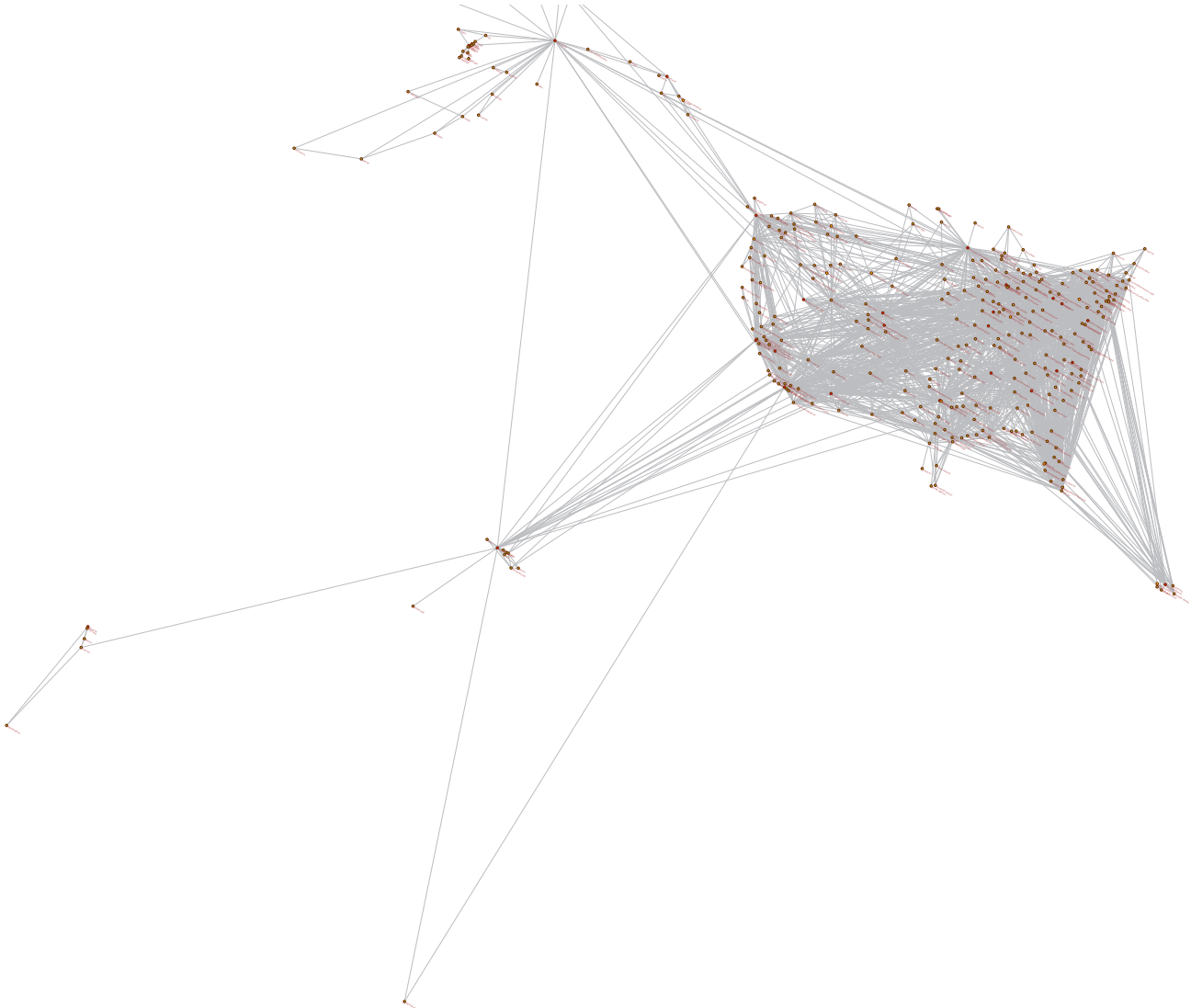
The solution is not unique. For example to cover the outsider 32 we can select any of the nodes 32, 31, and 33. To cover nodes 87, 66, 116 and 64 with a single node we have to select the node 65. Here is a minimal set

$$D_o = \{32, 55, 65, 97, 98, 141, 171, 262, 302, 320\}$$

The set of Roots contains 26 nodes + additional 10 nodes from D_o gives a minimal domination set D^* with $\gamma = 36$ nodes.

We manually add nodes from D_o to the Roots partition.

US Airports 1997 / minimal dominant set



Computing indices W and W_k

The set 5_*airports* consists of 5 nodes

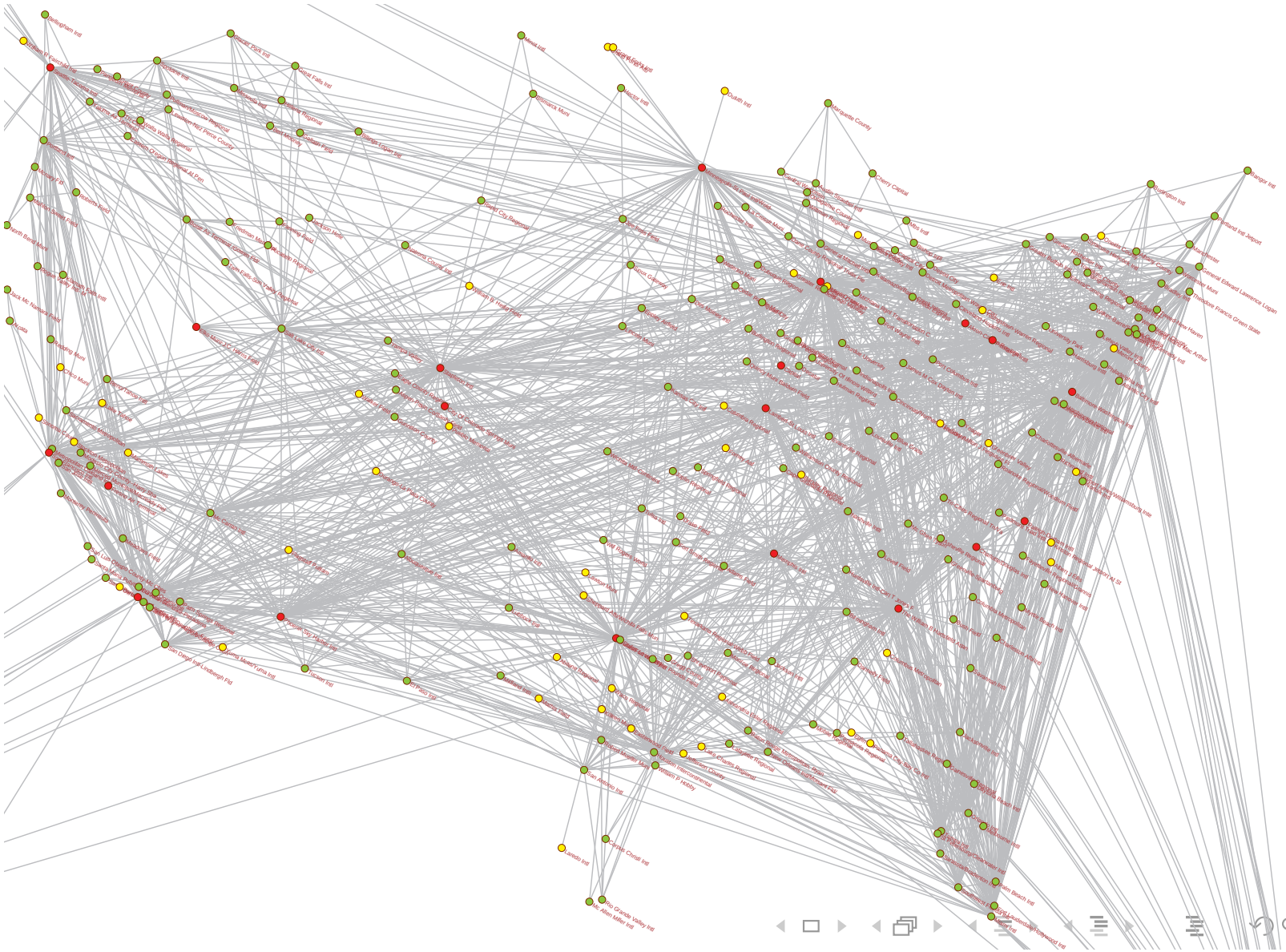
$$5\_airports = \{8, 118, 248, 255, 261\}$$

that correspond to airports: Anchorage Intl, Chicago O'hare Intl, Los Angeles Intl, The William B Hartsfield Atlanta, Dallas/Fort Worth Intl.

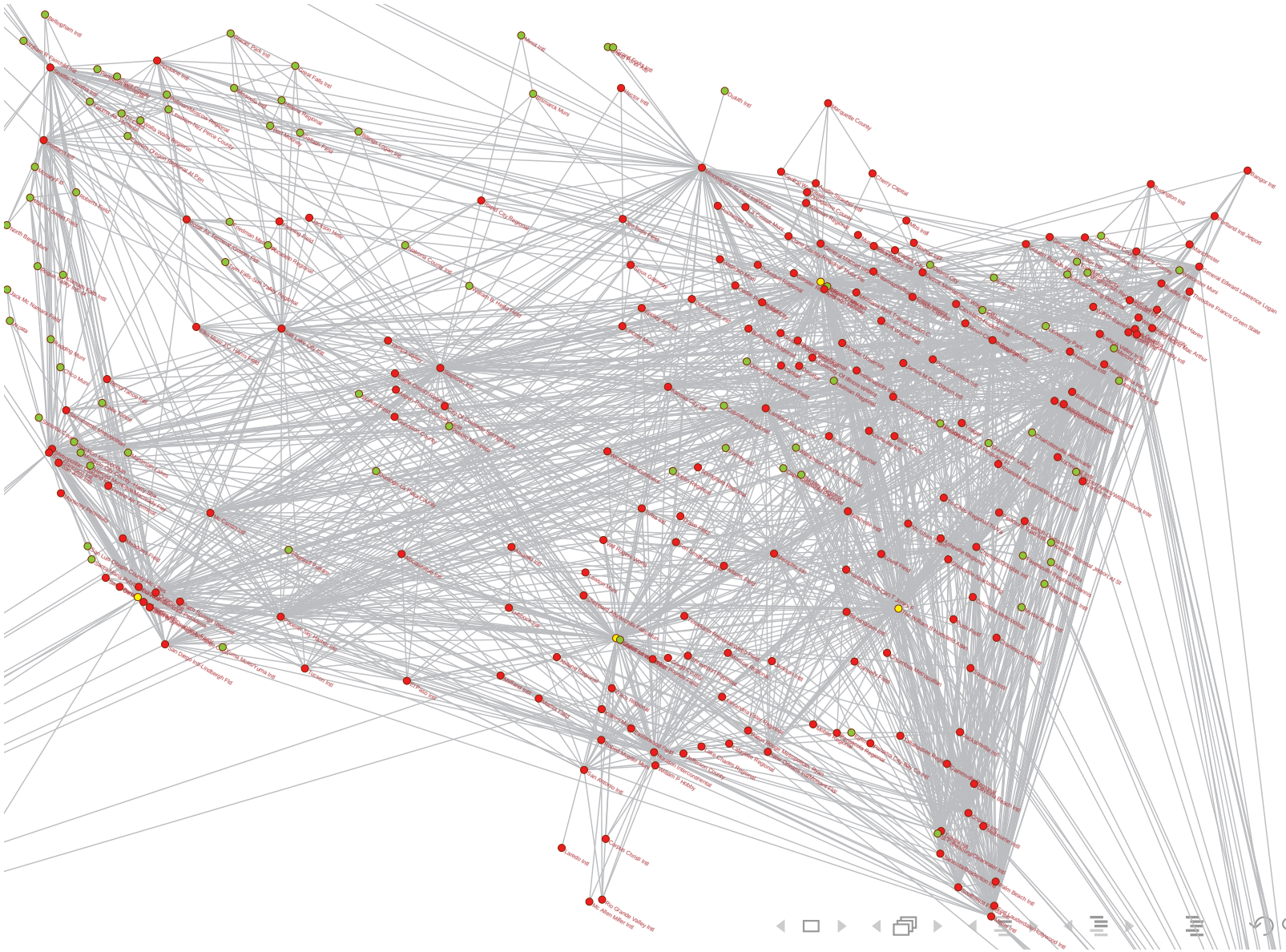
For the `USair97` network we have $n = 332$ and $\gamma = 36$.

S	*Leaves*	5_*airports*
$\lvert S\rvert$	55	5
$\lvert N(S)\rvert$	26	218
$\lvert S \cup N(S)\rvert$	81	218
$\lvert N(S) \setminus S\rvert$	26	213
$W(S)$	$\frac{81}{332} = 0.243976$	$\frac{218}{332} = 0.656627$
$W^*(S)$	$\frac{26}{296} = 0.087838$	$\frac{213}{296} = 0.719595$

US Airports1997 / C = *Leaves*



US Airports 1997 / $C = 5\_airports$



References I


Vladimir Batagelj, Andrej Mrvar: Pajek manual.


Wouter De Nooy, Andrej Mrvar, Vladimir Batagelj: Exploratory Social Network Analysis with Pajek; Revised and Expanded Second Edition. Structural Analysis in the Social Sciences, Cambridge University Press, September 2011.


Wikipedia: Dominating set.


Wikipedia: Independent set.