Szemerédi-type clustering of peer-to-peer streaming system

Vesa Pehkonen and Hannu Reittu
VTT Finland

MODELING, ANALYSIS, AND CONTROL OF COMPLEX NETWORKS

Cnet 2011, 9.Sept. San Francisco

agenda:

- review of Szemerédi's Regularity Lemma
- Peer-to-peer streaming system
- Szemerédi-type clustering of p2p-system

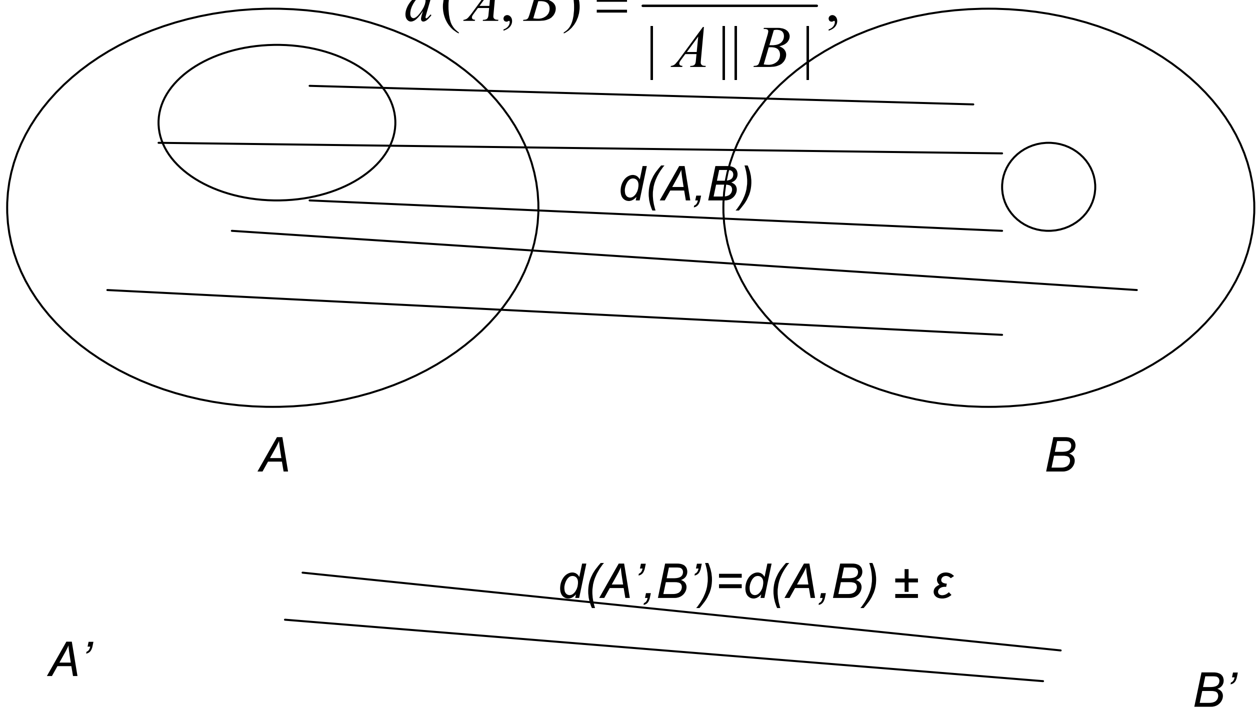
Szemerédi's Regularity Lemma (SzRL)

- a fundamental result in graph theory
- Szemerédi, E.:'Regular partitions of graphs', 1978

an ‘ε-regular pair’ *(A,B) is a graph*:

- *A* and *B* are disjoint node sets
- link density (*d(A,B)*) between pair is almost uniform

$$d(A,B)=\frac{e(A,B)}{|A||B|},$$



for all: $|A'| \geq \varepsilon|A|$, $|B'| \geq \varepsilon|B|$

SzRL

for every $\varepsilon > 0$, and for every natural number m

- $\exists$ $N=N(\varepsilon,m)$ and $M=M(\varepsilon,m)$, natural numbers

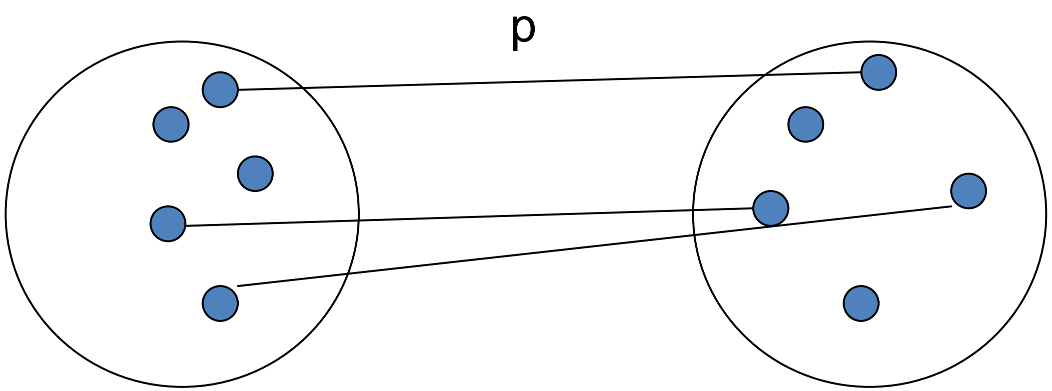
such that:

- any graph, G, with at least N nodes can be partitioned into k classes of almost equal sizes: $(V_1, V_2, \ldots, V_k)$
- $m \leq k \leq M$ and all but at most εk^2 pairs (V_i, V_s) are ε-regular

(then $(V_1, V_2, \ldots, V_k)$ is called an <u>ε-regular partition</u> of G)

note:

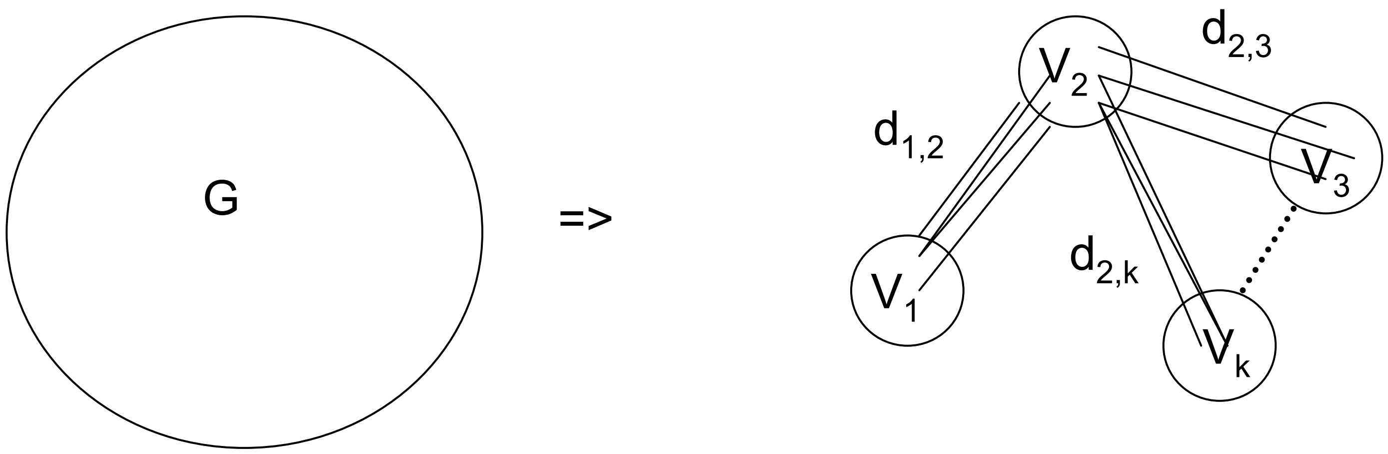
- a large random bipartite graph is 'quite' regular



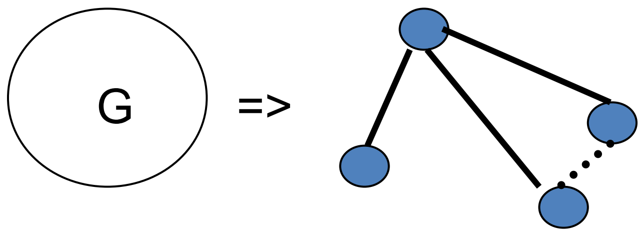
- link is drawn with prob. p, and not drawn with prob. 1-p
- link density is close to p
- regularity ≈ uniformity (of links)

SzRL ≈>

a large enough graph can be well approximated by bounded number of pseudo-random bipartite graphs



- G => Sz. graph



- space of all graphs is totally bounded or precompact

some remarks on SzRL

- most effective for large and dense graphs
- sparse version (Scott 2010)?
- ‘must’ in extremal graph theory
- interesting new results:
  - O(n) time algorithm to find reg. partitions (E. Fischer et al, 2010)
  - Sz. graph can be found in constant time! (E. Fischer et al, 2010, T. Tao, 2009)
  - <u>Theorem</u>: a graph property, *P,* is effectively testable ⇔ *P* can be verified from corresponding Sz. graphs (N. Alon, et al 2005)
  - Sz. graph tolerates substantial noise, spectral methods, M. Bolla 2005

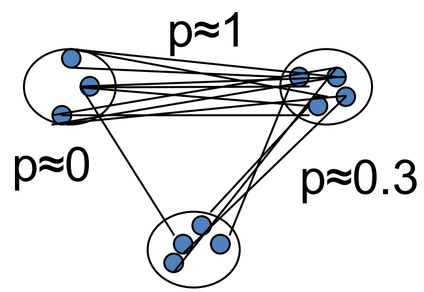
although SzRL does not guarantee reg. partitioning for any ‘real life’ graph:

- because of terrific upper-bounds (M) like:

$$2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}} \quad \frac{1}{\varepsilon^5}$$

- regular partition can still be meaningful
- a reasonable model in many cases:

- Sz. graph as a model
  - fit real life graph with such model in max likelihood sense
  - try to find partitioning where pairs look as random as possible => as regular as it can get(?)
  - try to cluster nodes in a way that pairs are like random bipartite graphs:

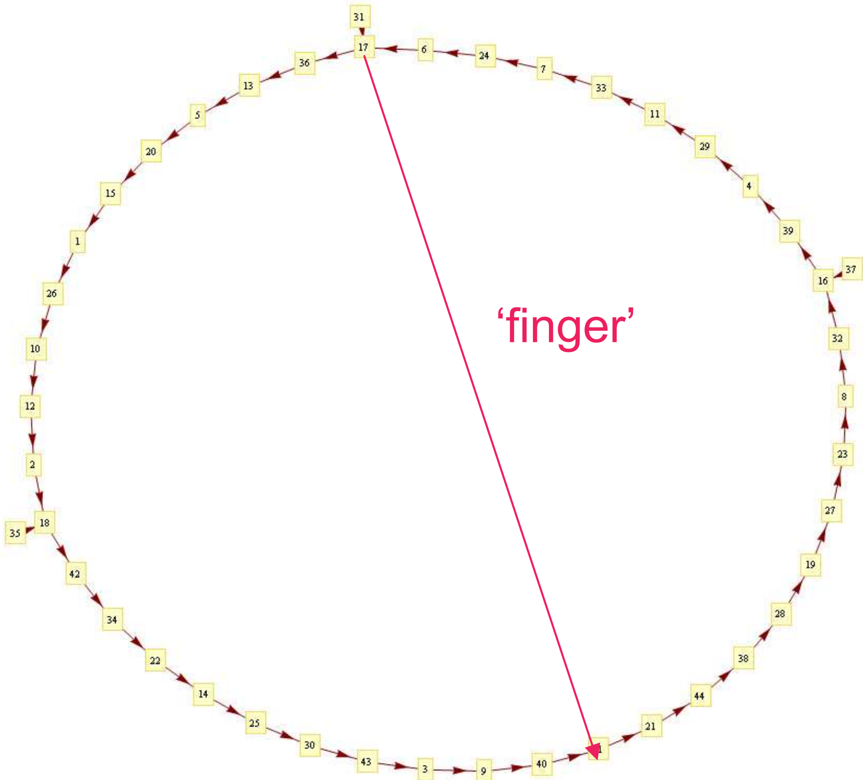


we adopted the last approach from:

- Nepusz-Bazsó-Négyessy-Tusnády, “Reconstructing cortial networks…” Springer 2008.
- network of brain areas with neural connections as links
- with uncertain links
- Sz. clustering => predictions about such uncertain links!

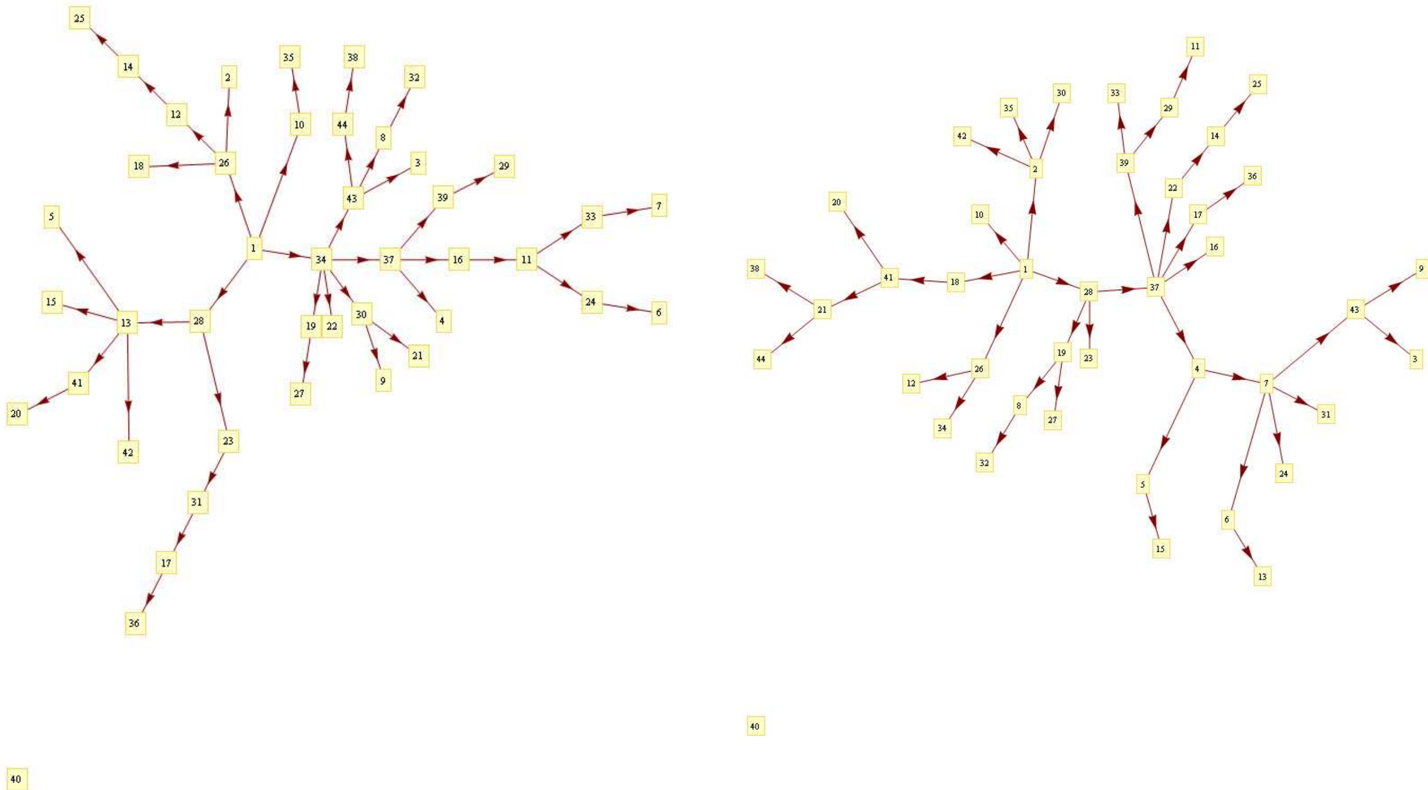
our experimental peer-to-peer streaming system, tests in PlanetLab

- 48 nodes organizing a Chord-network

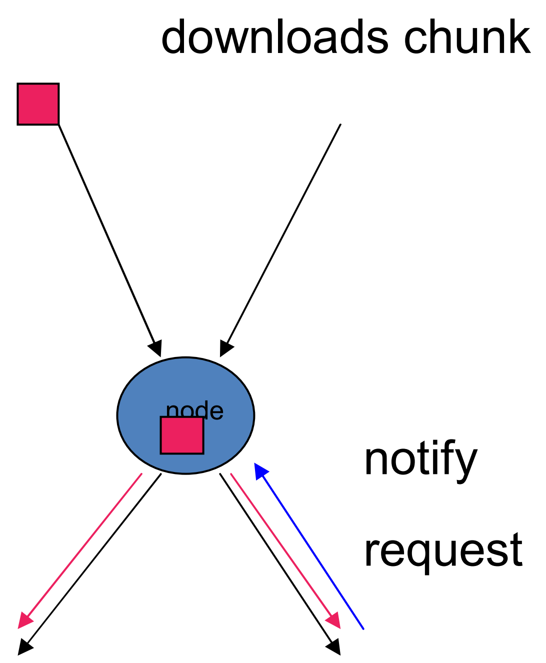


one node, the ‘seed’, streams chunks of the file to its antifinger neighbours -> new neighbours…

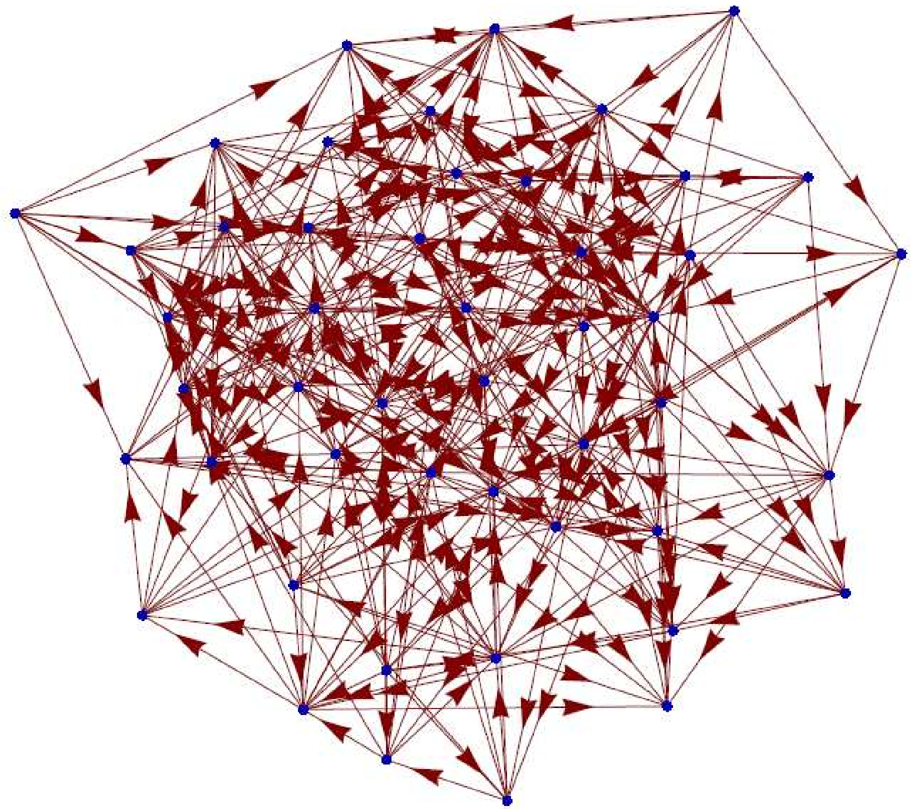
a case of some two chunks:



a chunk is downloaded, neighbours are notified, one of them requests the chunk, and a peer accepts request and uploads the chunk further – a push scheme



after thousands of chunks were streamed: 'who downloaded from whom'-graph



we would like to find peers downloading-uploading patterns

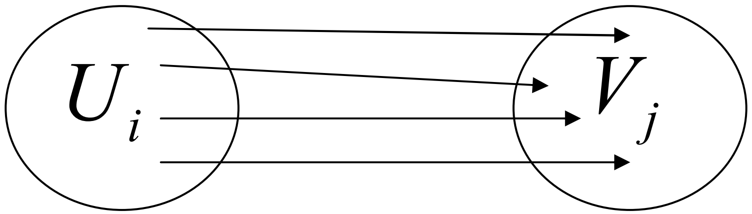
- similar peers in the same class
- similarity in some Sz. clustering sense (?)
- without any aprior assumptions what the classes should be

in directed case we want to find two partitions ('in' and 'out' clusters)

- in clusters: $\Upsilon = (V_1, V_2, \ldots, V_{Kin})$,
- and out clusters $\Omega = (U_1, U_2, \ldots, U_{Kout})$

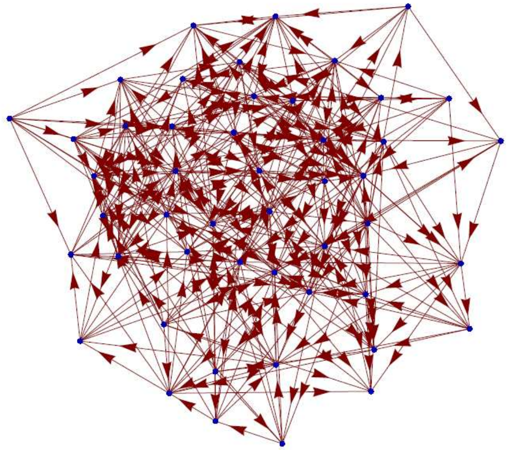
$$V = V_1 + \ldots + V_{Kin} = U_1 + \ldots + U_{Kout}$$

- consider links from U_i to V_j



define a directed and weighted graph:

- for link (i, j)



- weight: $w_{i,j} = \dfrac{n_{i,j}}{\sum_{\alpha} n_{i,\alpha}}$

- with $n_{i,j}$ = number of chunks node i gets from j during a long session

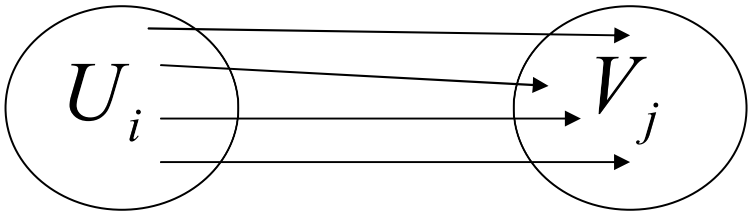
define a random graph model, $\mathcal{G}_W$

- with link probability: $P((i,j) \in E_w) = w_{i,j}$
- with independent links
- with the same nodes set as the original graph, V

with respect to in- and out-clusters define the weight density matrix:

$$(P)_{i,j} := p_{i,j} = \frac{\sum_{\alpha \in U_i, \beta \in V_j} w_{\alpha,\beta}}{|U_i||V_j|}$$

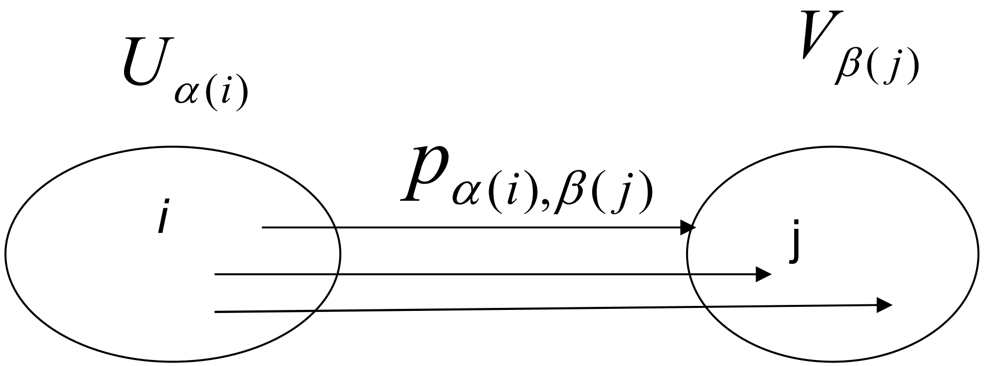
$$1 \leq i \leq Kout, \quad 1 \leq j \leq Kin,$$



two random graphs: $\mathcal{G}_W$ and $\mathcal{G}_P$

- with link probabilities:

$w(i, j)$ and $p_{\alpha(i),\beta(j)}$ with



find partitions (Ω, Y)

- in such a way that the expected 'log-likelihood' is maximal:

$$l(\Omega, \mathrm{Y}) := E_{\mathcal{G}_w} \log P(G \mid \mathcal{G}_P)$$

- where prob. of a graph $G(E,V)$:

$$P(G \mid \mathcal{G}_P) = \prod_{(i,j) \in V \times V : (i,j) \in E} p(i,j) \prod_{(i,j) \in V \times V : (i,j) \notin E} (1 - p(i,j))$$

=>

$$l(\Omega, \Upsilon) = \sum_{(i,j)\in V\times V} \left(w_{i,j} \log p_{u(i),v(j)} + (1 - w_{i,j}) \log(1 - p_{u(i),v(j)}) \right)$$

where $i \in U_{u(i)}, j \in V_{v(j)}$

with fixed $|\Omega| = Kout, |\Upsilon| = Kin$

$$\max_{\Omega, \Upsilon} l(\Omega, \Upsilon) := l^*(Kout, Kin)$$

(use EM-algorithm)

$$|\Omega| = Kout, |\Upsilon| = Kin???$$

- from *AIC*, Akaike information criterion:
- minimze: $AIC = -2l^*(Kout, Kin) + 2K$
- where *K* is number of parameters: $K = Kin \times Kout + 2|V| + 2$
- => $AIC = -2l^*(Kout, Kin) + 2Kin \times Kout + 4|V| + 4$
- Min *AIC* => $|\Omega| = Kout, |\Upsilon| = Kin$

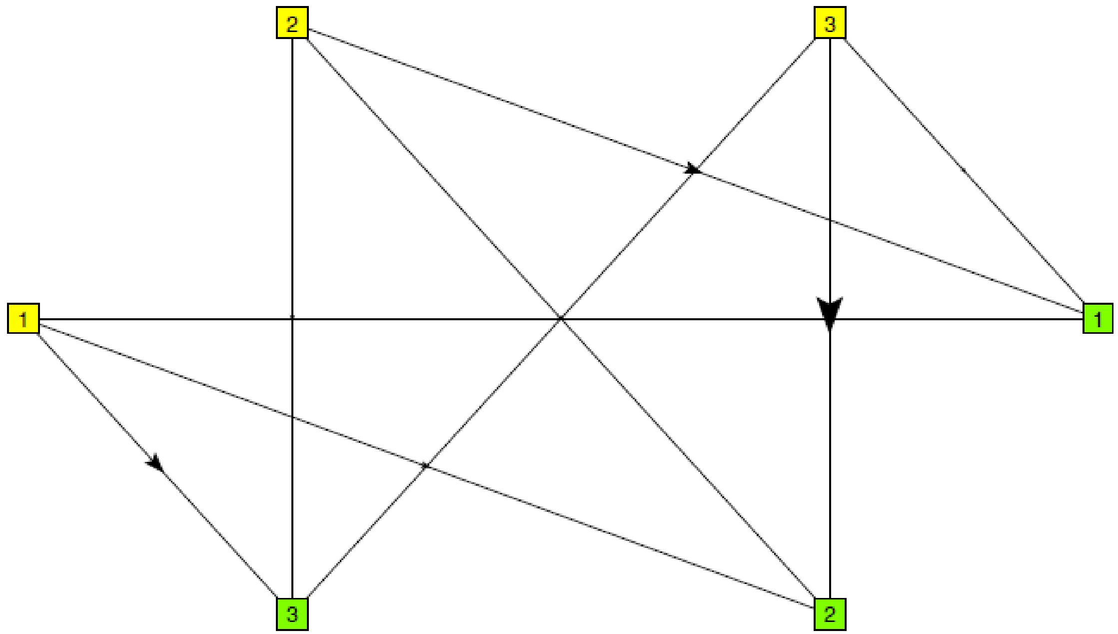
clustering algorithm:

- use EM (expectation-maximisation)
- FOR: $$Kin = 1,\ldots,m; Kout = 1,\ldots.m'$$
- start from random partitions
- **A**: calculate *P-matrix*
- for each node: calculate exp. log-likelihood considering that the node belongs to a given pair of in- and out-cluster; find max over all such cluster pairs; (*P* is fixed)
- redistribute nodes in clusters, replace node to cluster pair that gives the max of expected log-likelihood
- Go To **A** and iterate until the clusters saturate
- calculate *AIC*
  - continue FOR, take clustering corresponding to min *AIC*

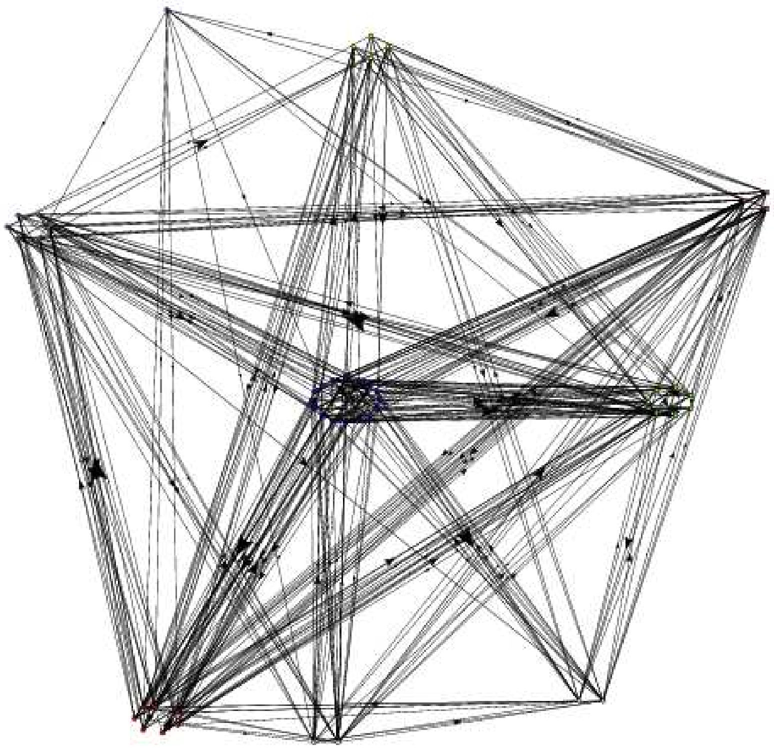
we end up with:

$$Kin = Kout = 3$$

- *Sz-graph:*

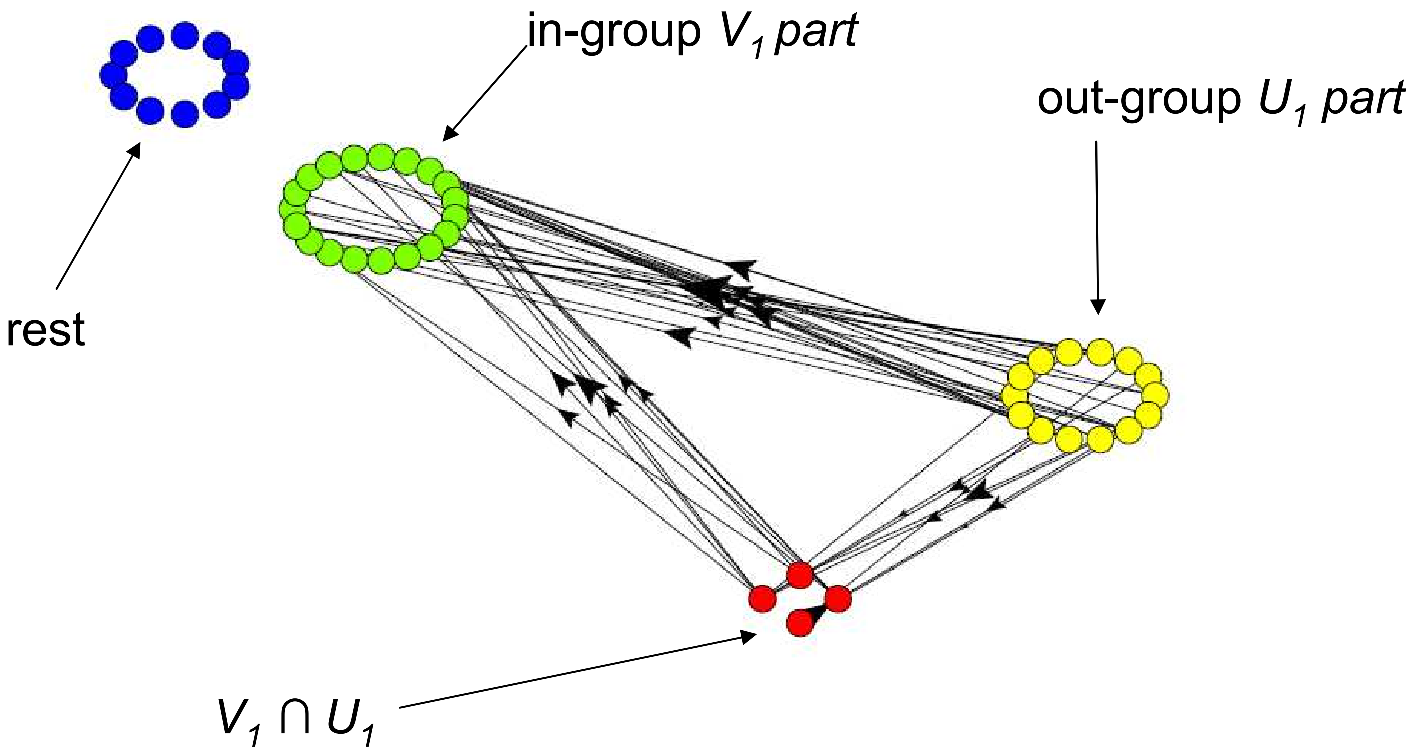


=> the same in more details

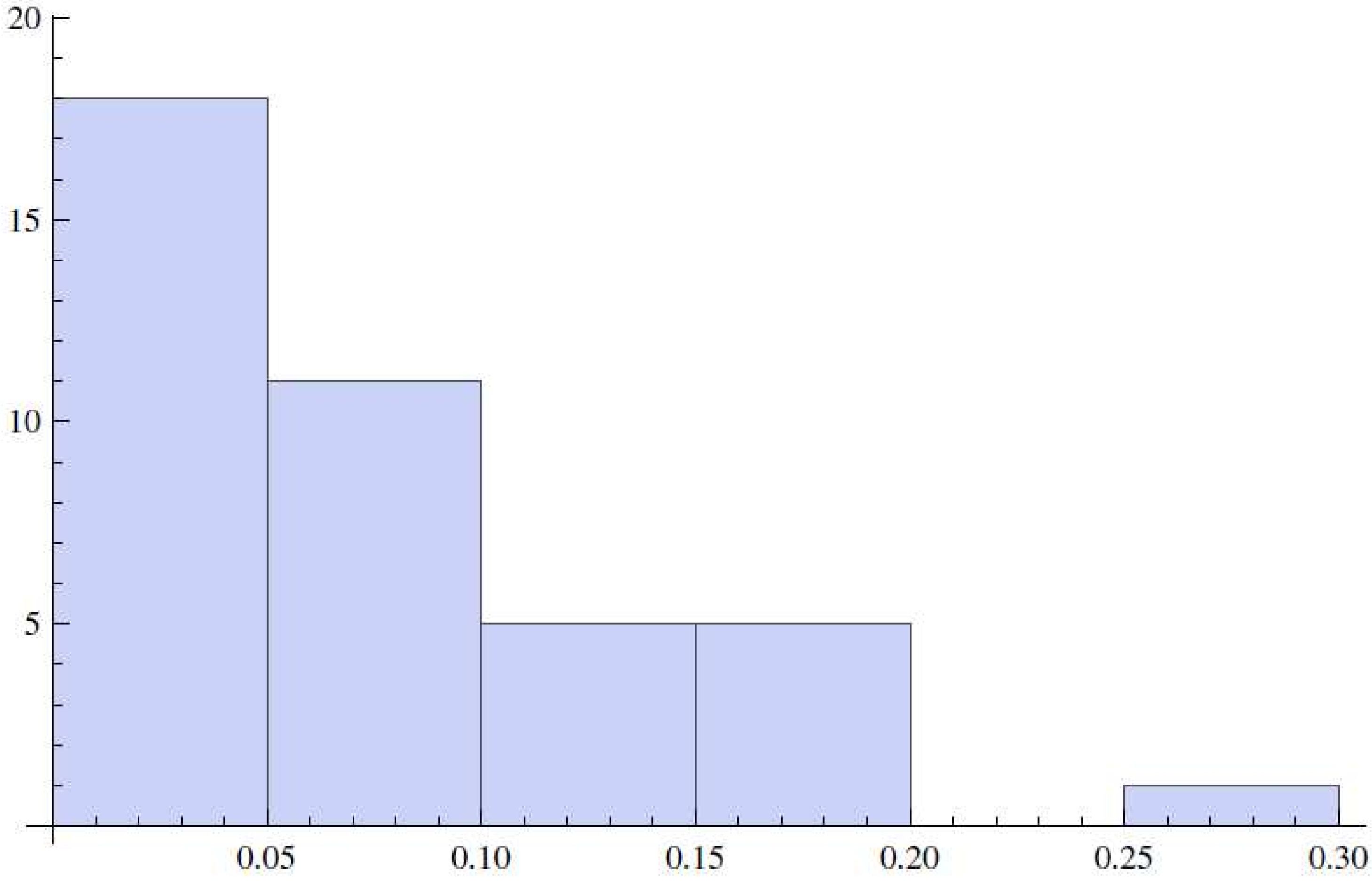


some apparent pattern were found

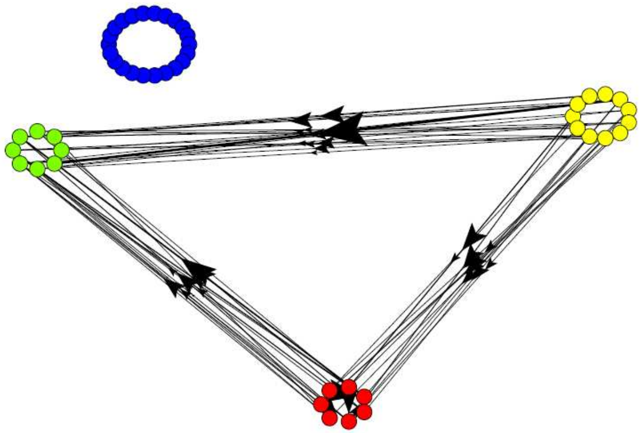
- cluster pair '(1,1)':

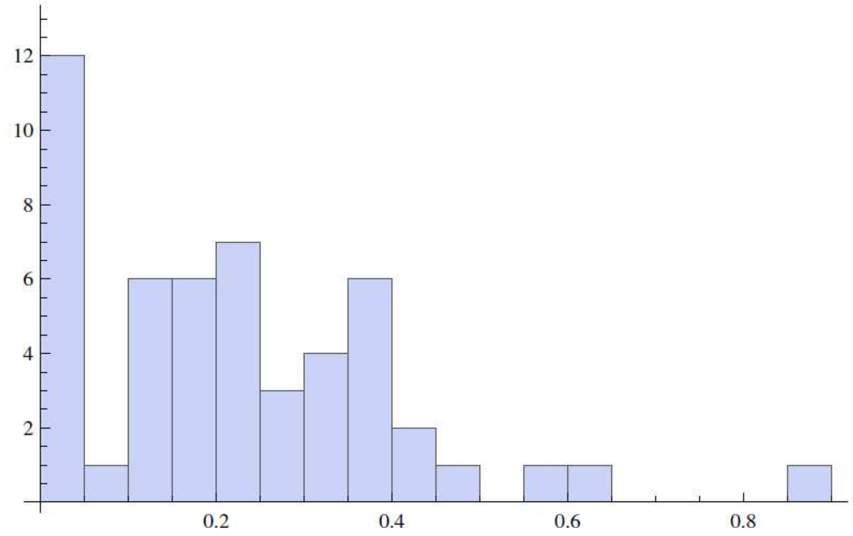


histogram of the weight distribution for pair (1,1)
very small weights, mean 0.07

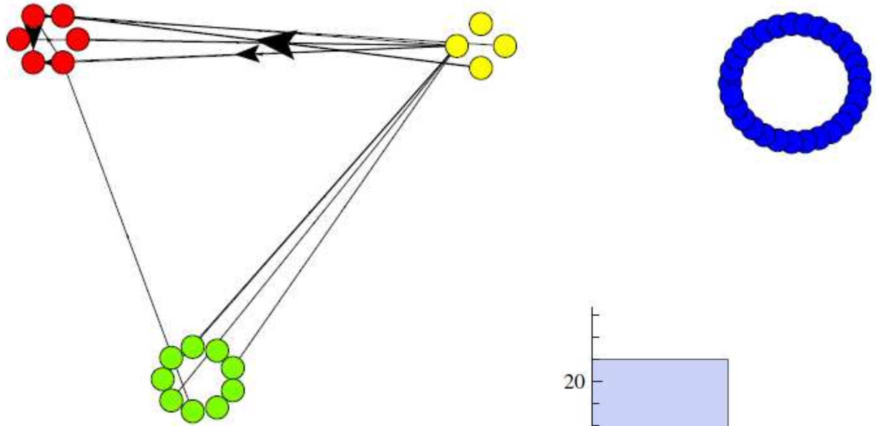


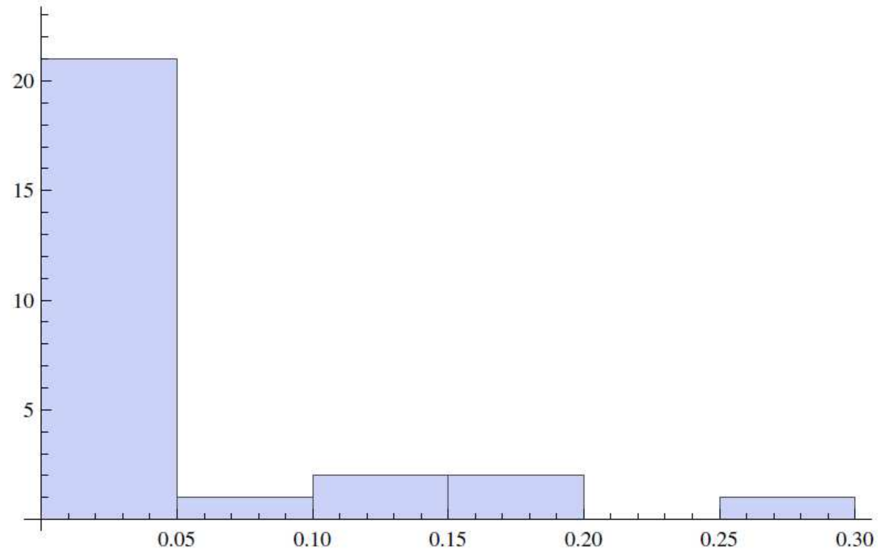
another pair (1,3)
with high weights, mean 0.22



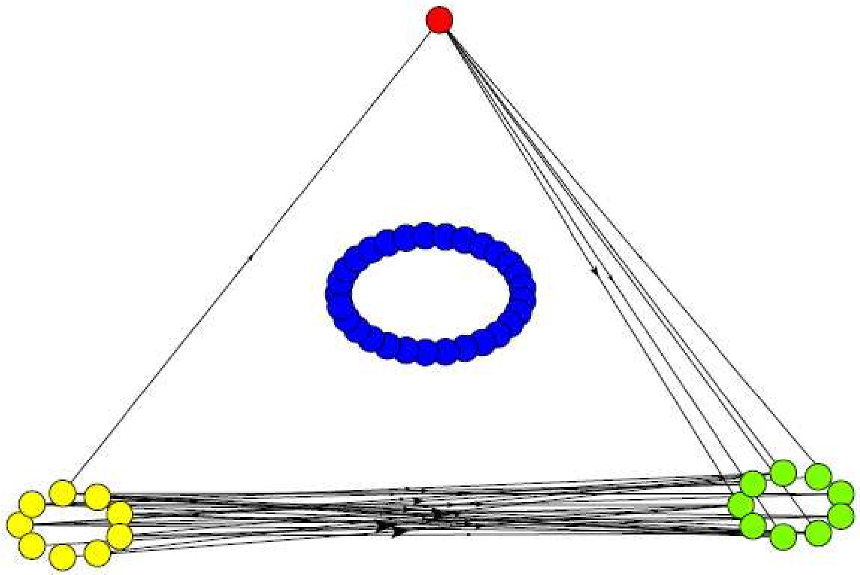


pair (3,1)
almost zero weights

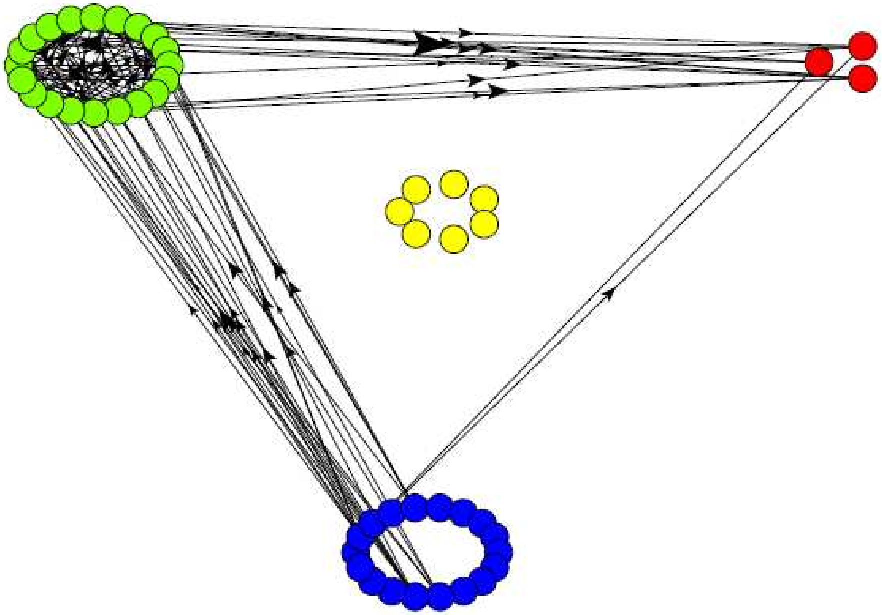


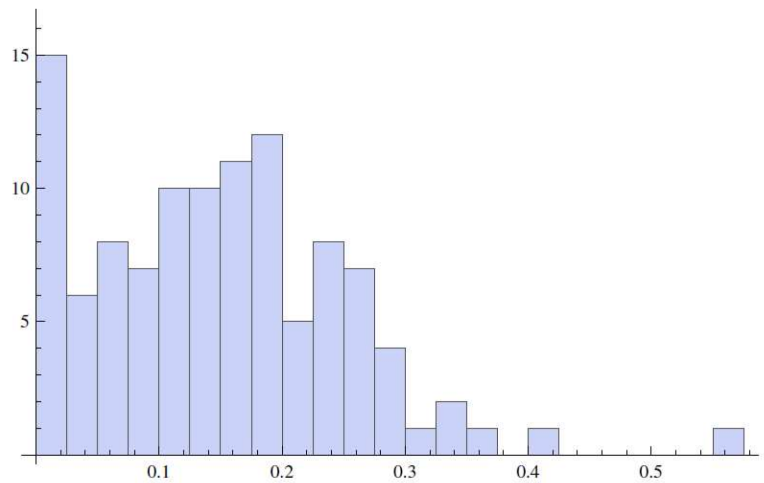


pair (3,2)
client-server type

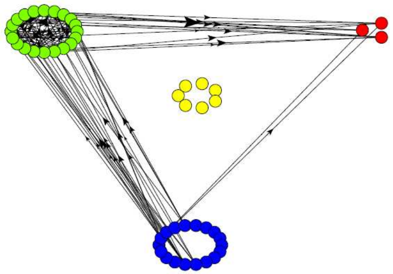


Pair (2,1)
with large intersection group
high link weights

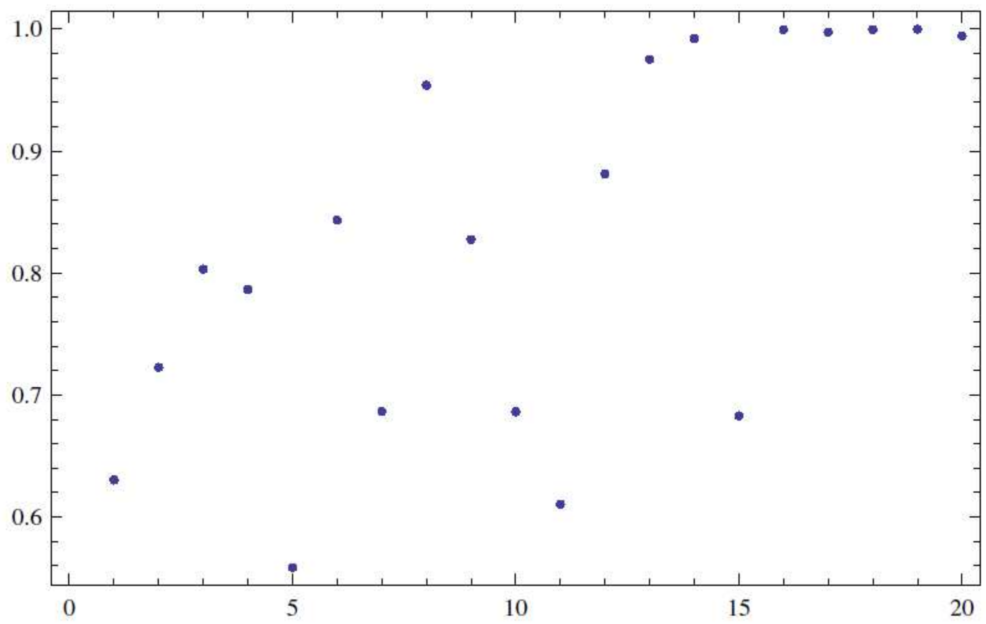




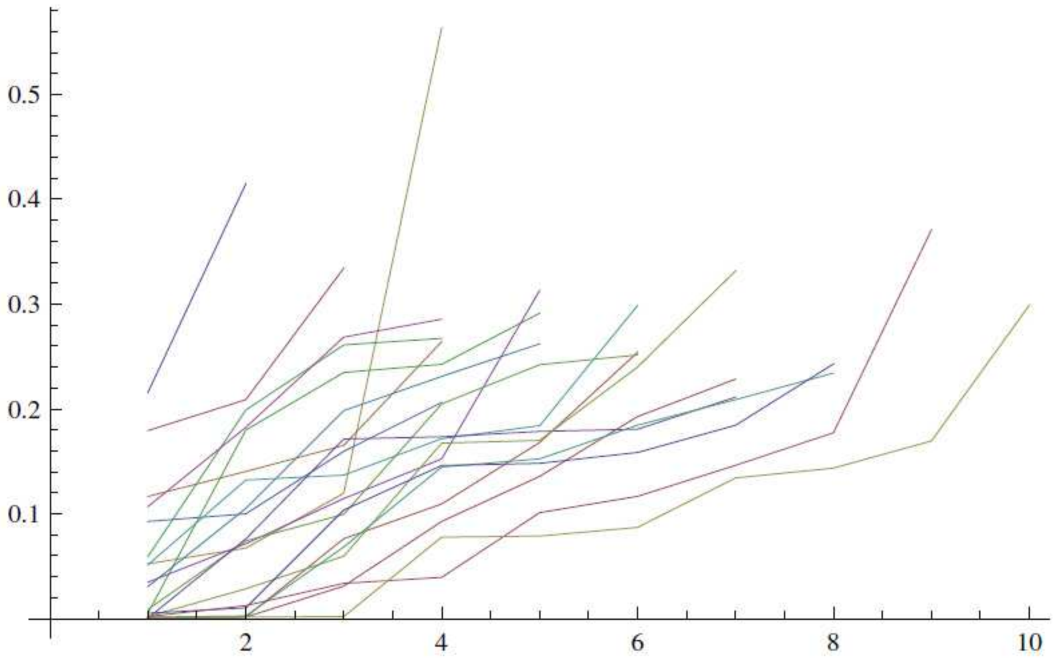
pair (2,1)
most peer-to-peer like



Many peers get most of chunks within this pair:

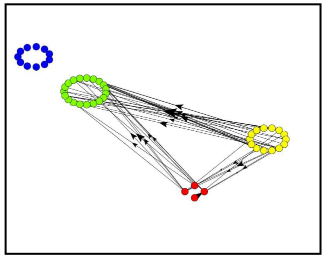
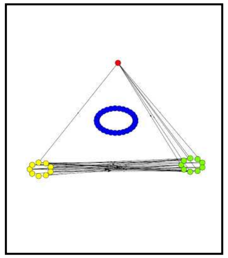
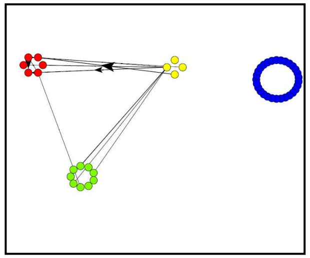
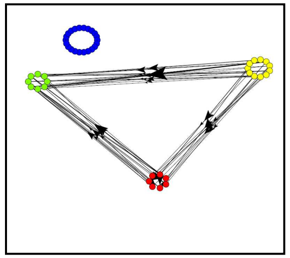
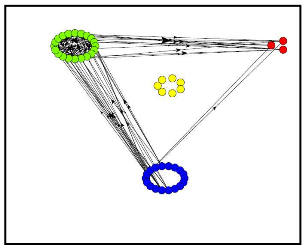


profiles of weights for (2,1)



comments:

- more interesting results for larger networks?
- use in metabolic networks
- MC-algorithm would be a better choice
- seems to have some potential

Thank You!