A comparaİson of Communİty Detectİon Algorİthms on Artİfİcİal Networks

Günce Keziban Orman

Galatasaray University C.S. Department
TUBITAK UEKAE
keziban.orman@uekae.tubitak.gov.tr

Vincent Labatut

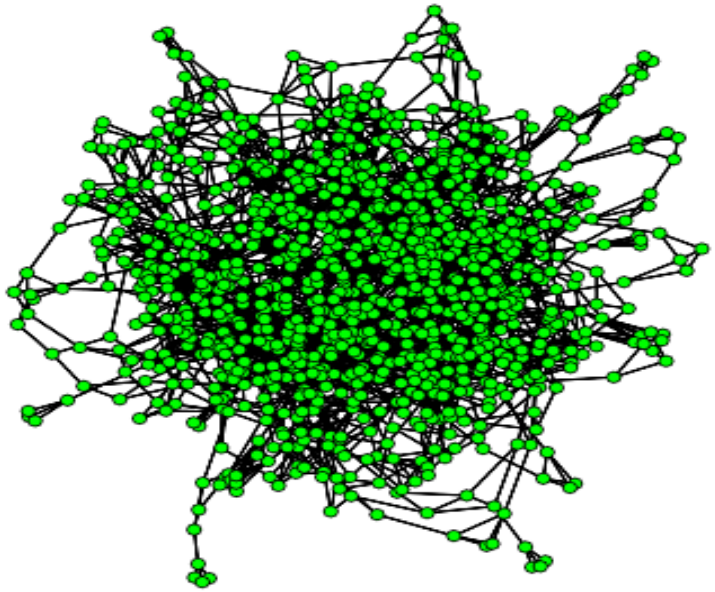
Galatasaray University C.S Department
vlabatut@gsu.edu.tr

6/27/2009

1. Introduction
    1. Complex networks
    2. Community detection
    3. Testing algorithms
2. Method
    1. Selected algorithms
    2. Lancichinetti *et. al* generative model
    3. Normalized mutual information
3. Results & Discussions
4. Conclusion

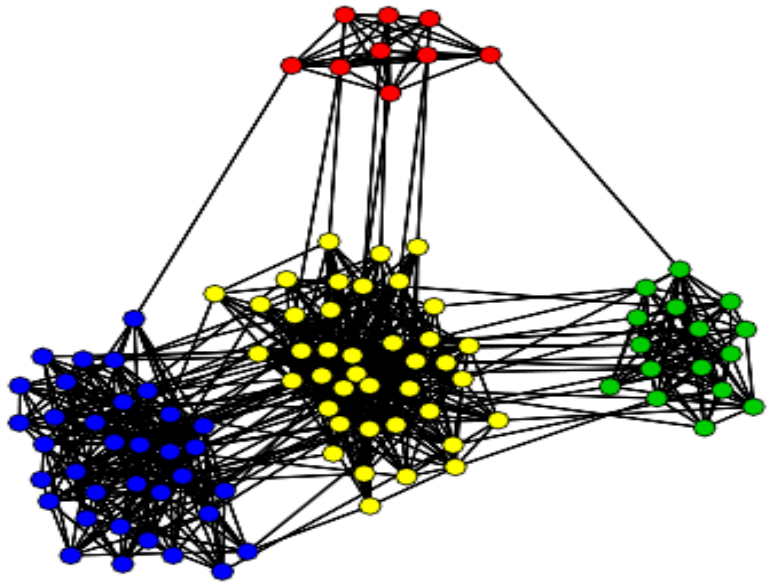
1.1 Complex Networks

- Large graphs with non-trivial topological features
- Model systems of interacting objects.
- ex: Internet, www, protein web, friendship networks ...



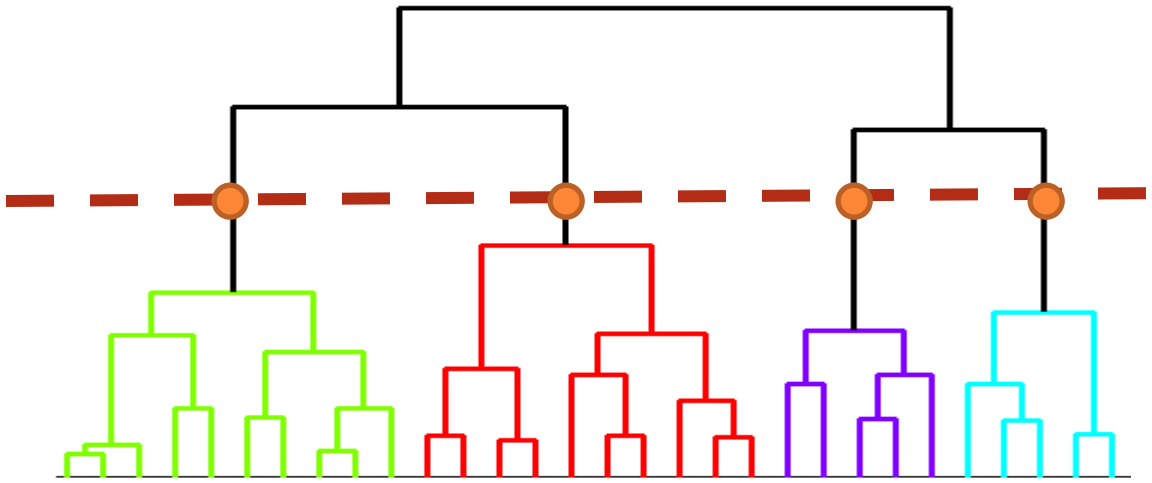
1.2.1 Community Detection Definition

- Community: group of nodes with dense inner links and sparse outer links
- Community detection: find the best graph partition according to this definition



1.2.2 Community Detection Solution Approaches

- Hierarchical Approaches
  - Divisive vs. Agglomerative
- Optimization Approaches
  1. Partition: stochastic vs heuristic
  2. Quality evaluation
- Others



1.2.3 Community Detection - Modularity

- Newman's modularity measure:

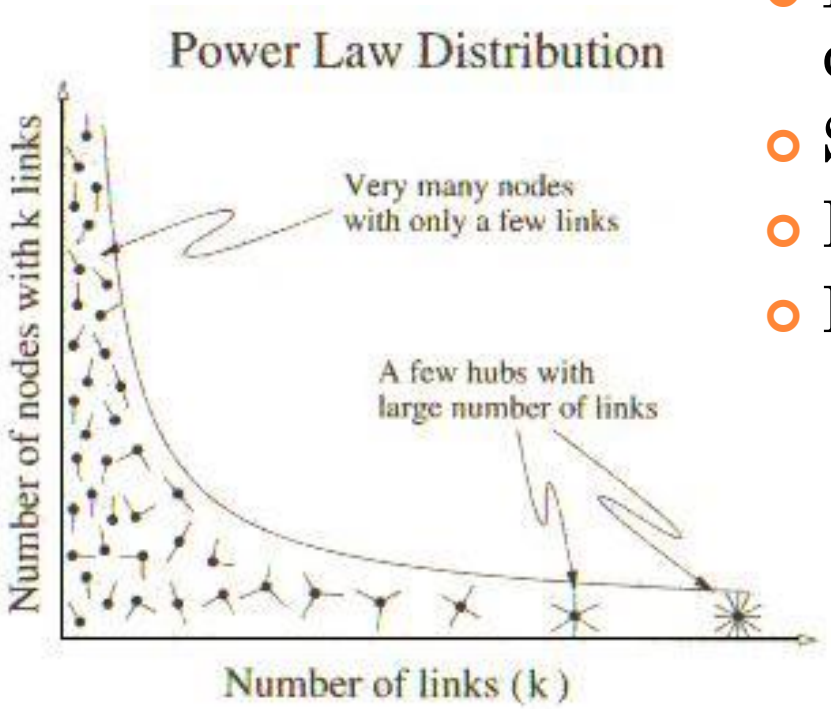
$$Q = \sum_i (e_{ii} - a_i^2)$$

  - e_{ii}: observed fraction of links inside the ith community
  - a_i: estimation of e_{ii} under the hypothesis of uniformly randomly distributed links.
- Values:
  - Q=0: networks without community structure and/or random partition
  - Q≈1: network with strong community structure and good partition
  - Q∈[0.3,0.7] is generally considered a good result

1.3 Testing Algorithms

	Real networks	Artificial Networks
Size	Usually small	No theoretical limit
Network properties	Uncontrollable, depends on the modeled system	depend on the generation model parameters
Construction	Expensive and/or difficult to build	Computer generated
Community structure	Possibly subjective or unknown community structure	Communities created and controlled

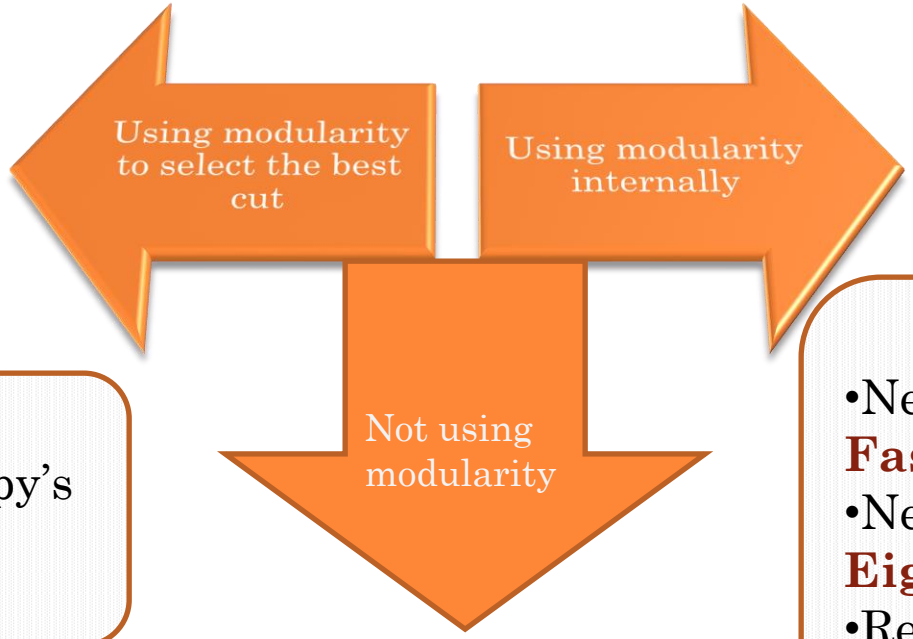
1.3 Testing Algorithms



Real Network Properties:

- Scale-free (power-law degree distribution)
- Power-law community size distribution
- Small average distance
- High transitivity
- High degree correlation

2.1 Selected Algorithms



- Pons and Latapy's **Walktrap**

- Raghavan *et al.*'s **Label Propagation**

- Newman *et al.*'s **Fastgreedy**
- Newman's **Eigenvector**
- Reichardt and Bornholdt's **Spinglass**

2.2 Lancichinetti *et. al* generative model

$$n, \beta, \gamma, \langle k \rangle, k_{max}, \mu$$

- Apply configuration model with average degree **<k>**, max. degree $\mathbf{k_{max}}$ and power-law exponent **γ**
- Draw community sizes with power-law exponent **β** and affect each node to a community
- Rewire some links in order to respect **μ**, without changing the nodes degrees

2.2 Lancichinetti *et. al* generative model

PARAMETER	VALUE
$\mathbf{n}$	{1000}
β	{1,2}
γ	{2,3}
$\langle k \rangle$	{5,15,30}
k_{max}	{15,45,90}
μ	[0.05,0.95]

2.3 Normalized mutual information (NMI)

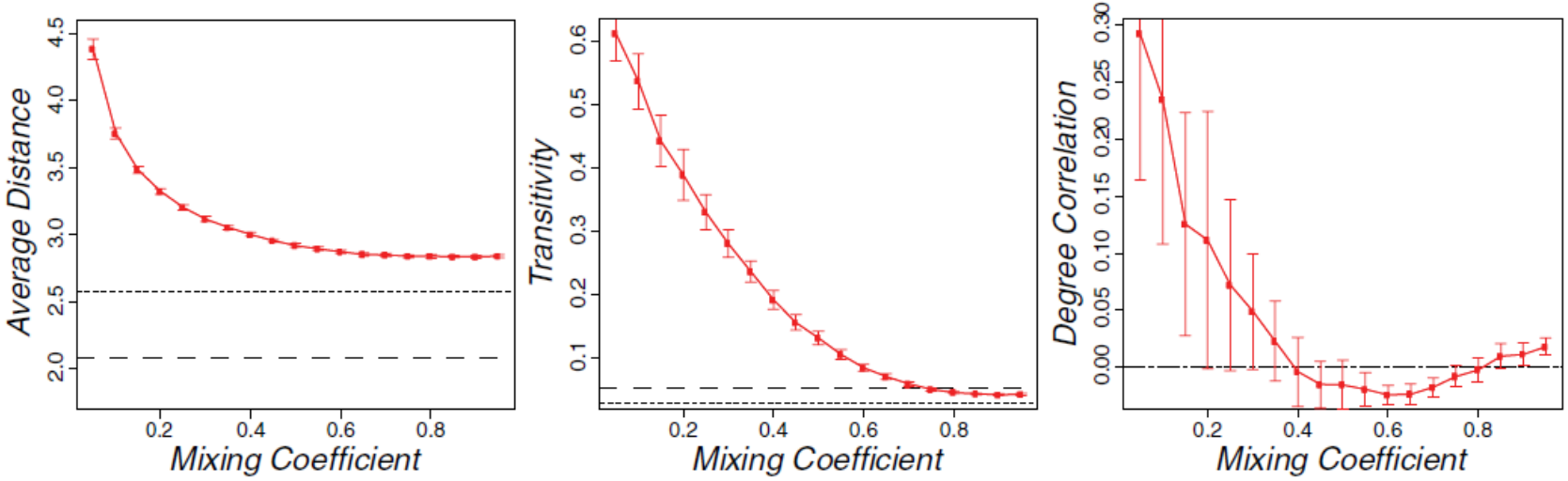
- Quality assessment for a dataset partition
  - m: confusion matrix
  - n: dataset size

$$I = \frac{-2 \sum_i \sum_j m_{ij} \log\left(n m_{ij} / m_{i+} m_{+j}\right)}{\sum_i m_{i+} \log(m_{i+}/n) + \sum_j m_{+j} \log\left(m_{+j}/n\right)}$$

- Values:
  - 0: random partition
  - 1: perfect partition

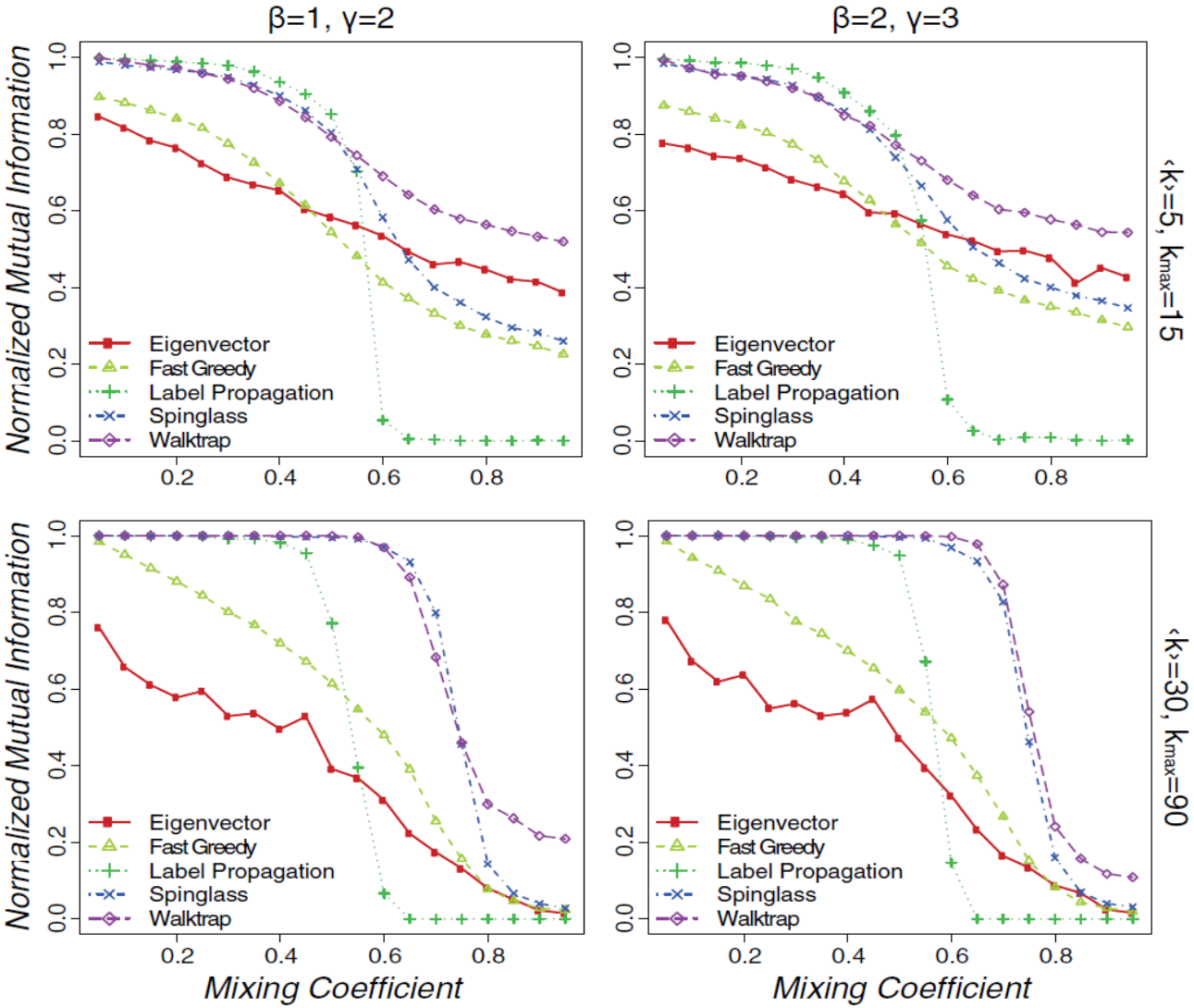
3. Results & Discussions

NETWORKS' UNCONTROLLED PROPERTIES

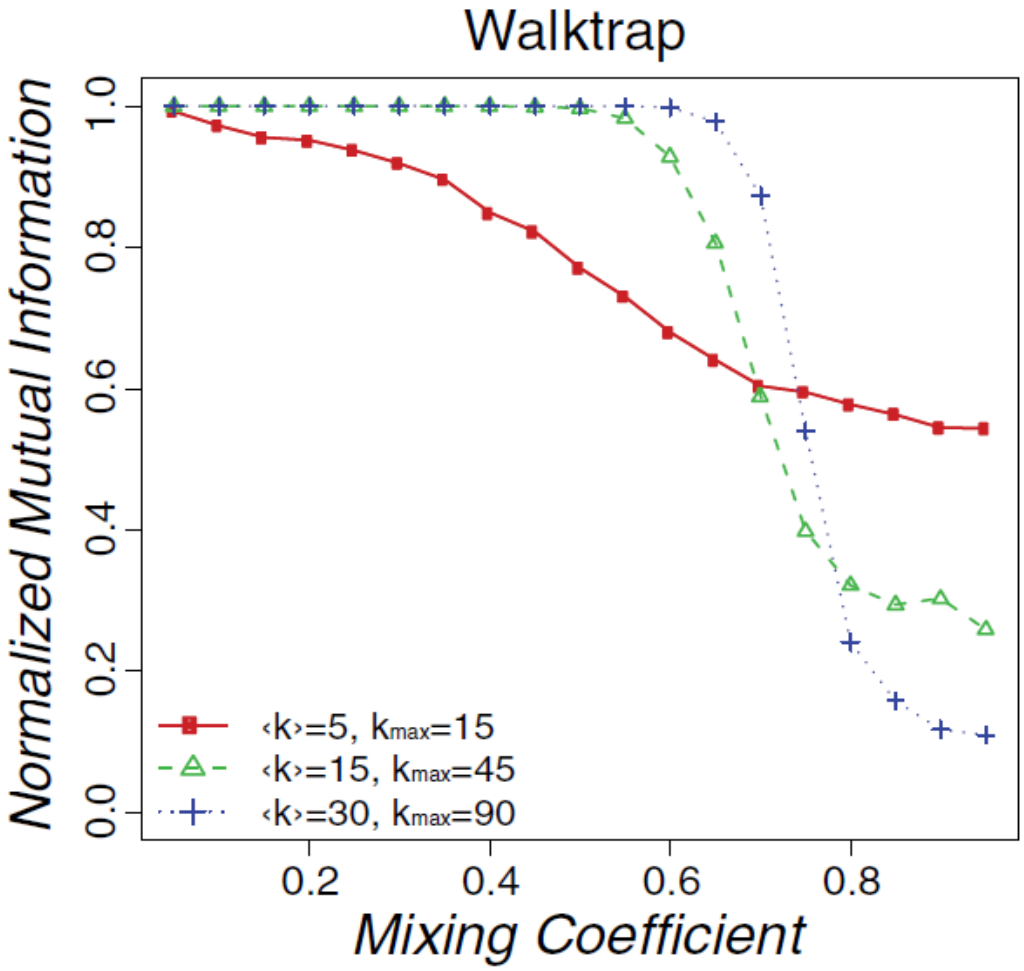


$\langle k \rangle = 30 \qquad k_{max} = 90$

3. Results & Discussions



3. Results & Discussions



3. Results & Discussions

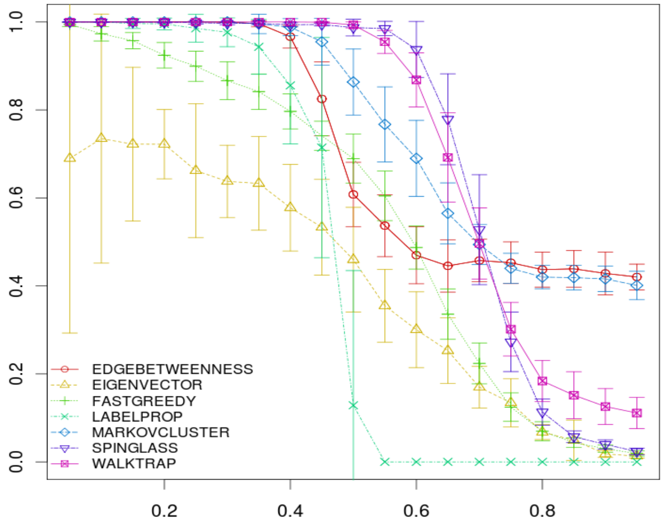
- Parameters effects:
  - β and γ: almost no effect ($\rho<0.06$)
  - $\langle k \rangle$: higher average degree improves performance
- Algorithms:
  - Partition quality:
    - WT and SG performs better
    - LP is not robust
  - Speed:
    - SG is slow,
    - LP, FG and WT are fast
    - EV lie somewhere in between

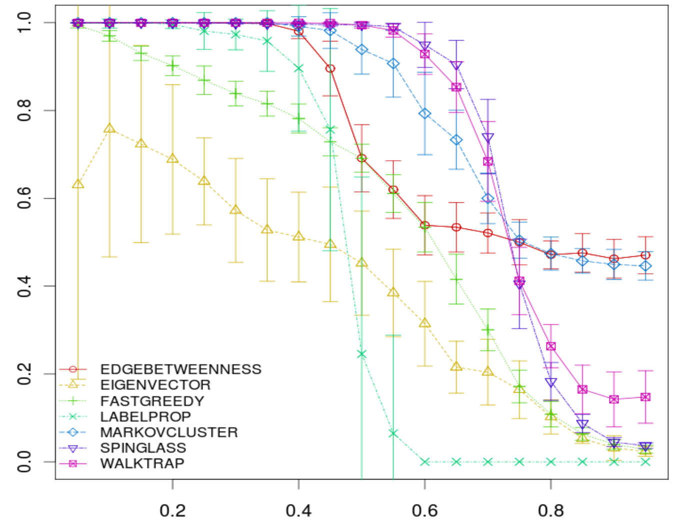
4. Conclusion

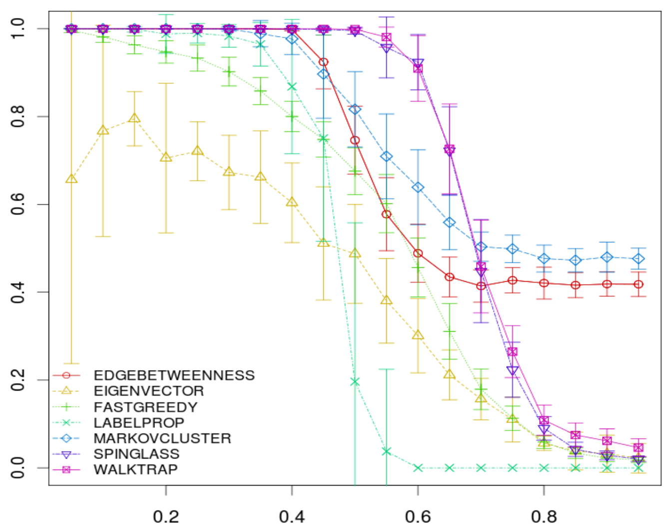
Further experiments

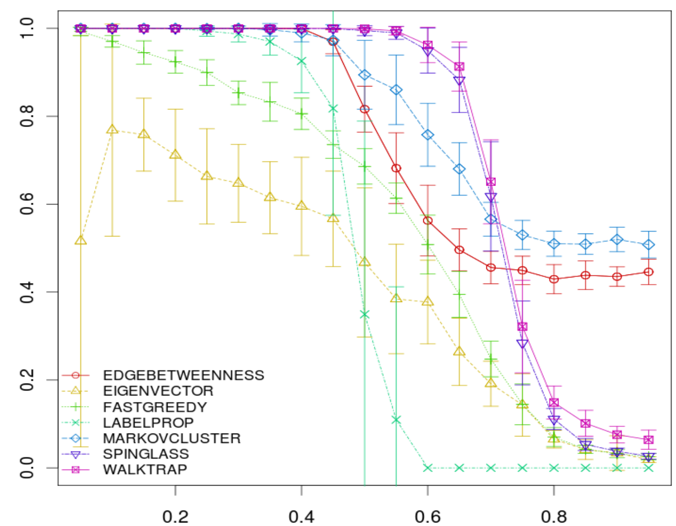
- on larger networks
- with more realistic networks
- with more algorithms
- with different performance measures

Questions


EDGEBETWEENNESS
EIGENVECTOR
FASTGREEDY
LABELPROP
MARKOVCLUSTER
SPINGLASS
WALKTRAP
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8


EDGEBETWEENNESS
EIGENVECTOR
FASTGREEDY
LABELPROP
MARKOVCLUSTER
SPINGLASS
WALKTRAP
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8

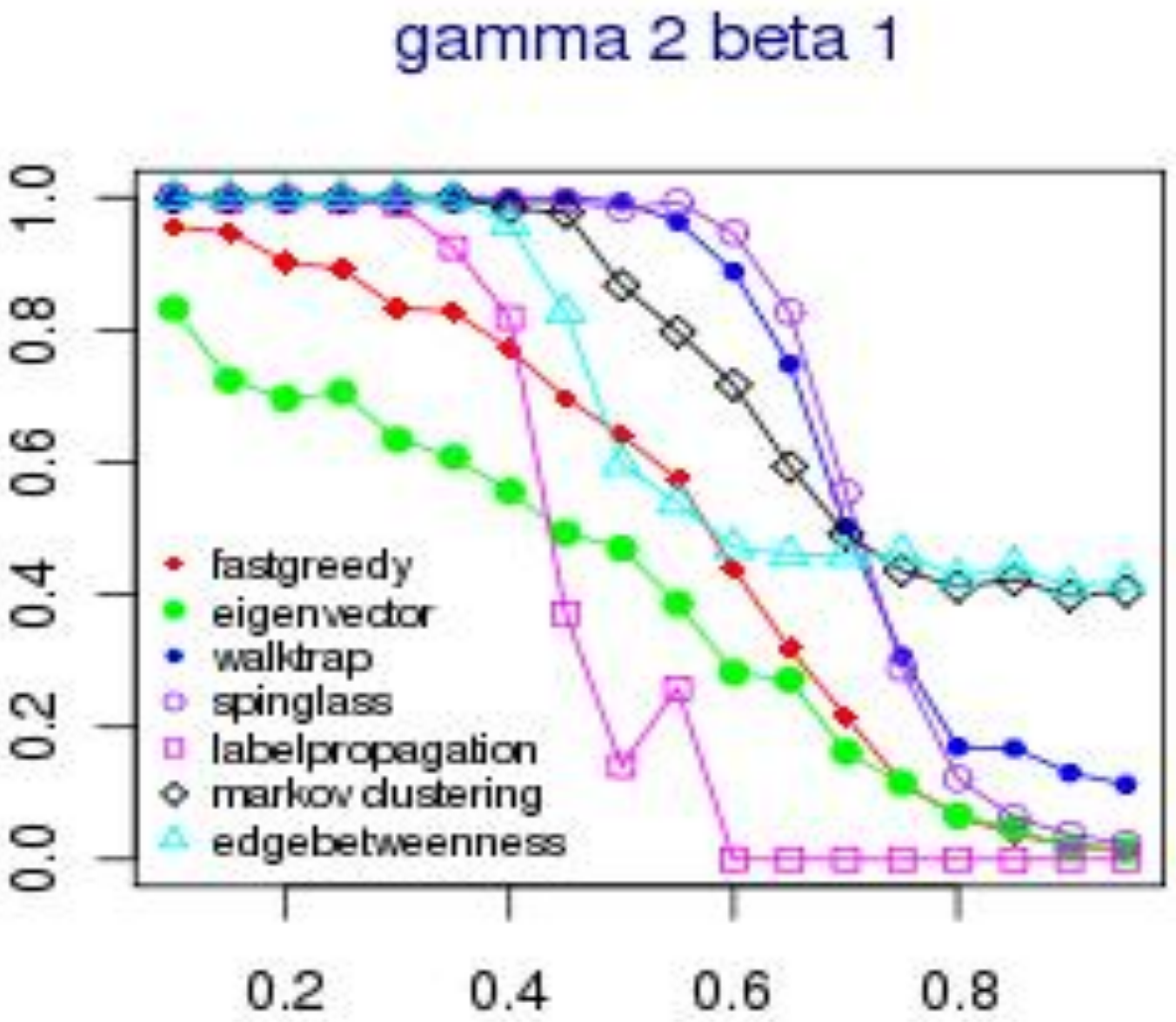

EDGEBETWEENNESS
EIGENVECTOR
FASTGREEDY
LABELPROP
MARKOVCLUSTER
SPINGLASS
WALKTRAP
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8

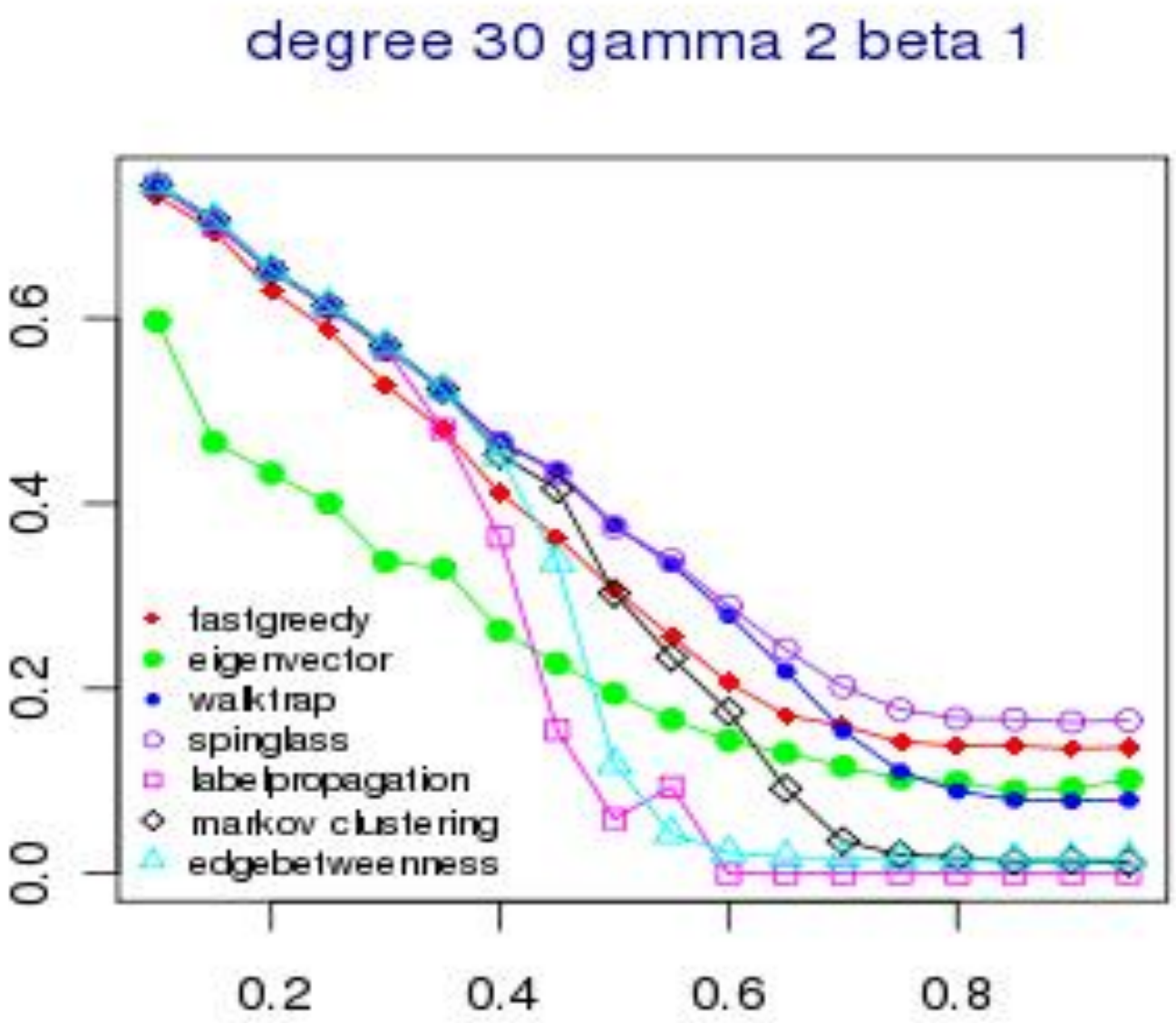

EDGEBETWEENNESS
EIGENVECTOR
FASTGREEDY
LABELPROP
MARKOVCLUSTER
SPINGLASS
WALKTRAP
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8

ALGORITHMS COMPLEXITIES

LP	FG	EV	WT	MC	SG	EB
$O(m+n)$	$O(n\log^2 n)$	$O(n^2)$	$O(n^2\log n)$	$O(nk^2n)$	*unknow*	$O(m^2n)$

n: node number; **m:** link number ; **k:** number of ressource allocated


gamma 2 beta 1
0.0
0.2
0.4
0.6
0.8
1.0
0.2
0.4
0.6
0.8
fastgreedy
eigenvector
walktrap
spinglass
labelpropagation
markov clustering
edgebetweenness


degree 30 gamma 2 beta 1
0.0
0.2
0.4
0.6
0.2
0.4
0.6
0.8
fastgreedy
eigenvector
walktrap
spinglass
labelpropagation
markov clustering
edgebetweenness