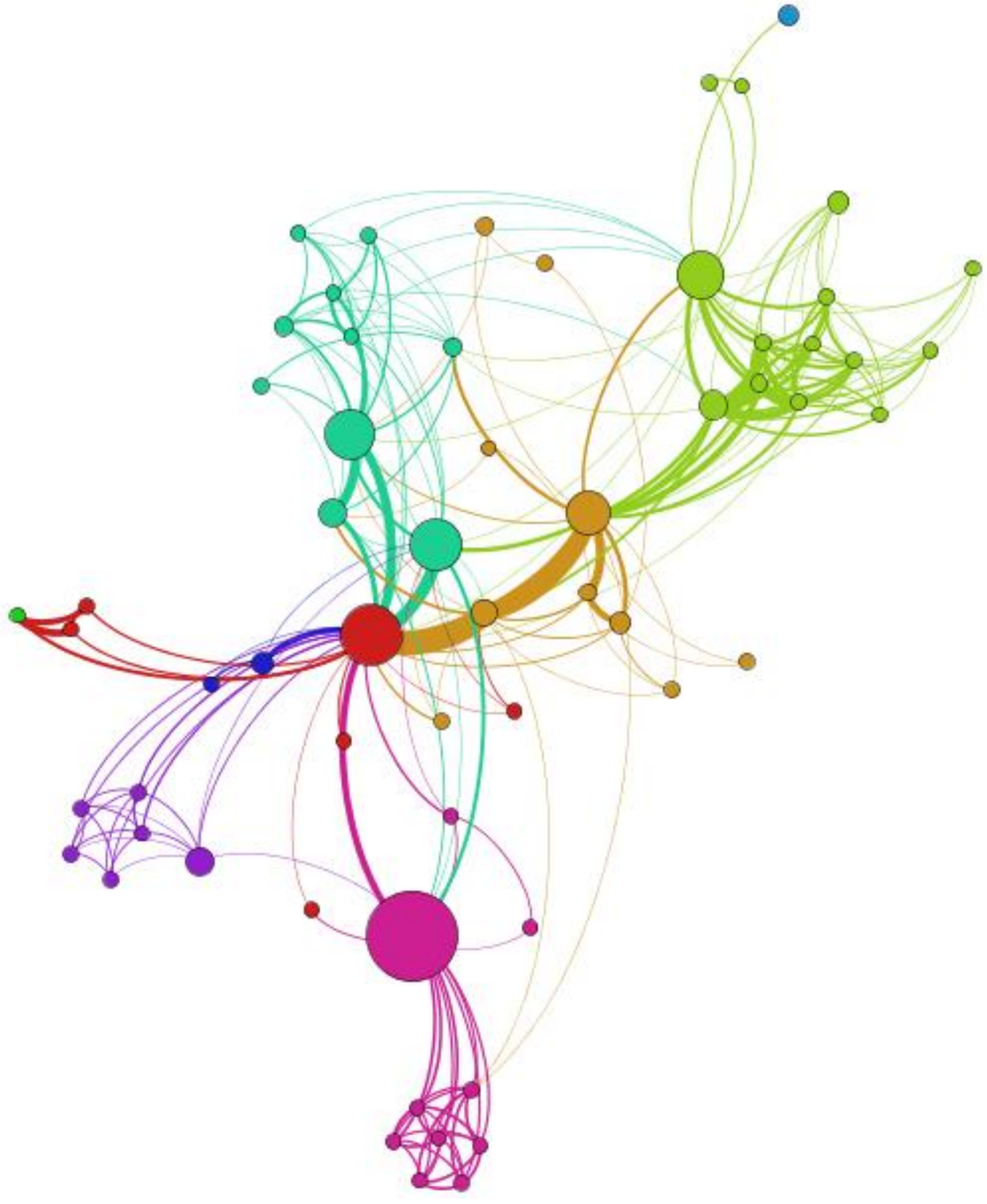
Collecting Influencers: a Comparative Study of Online Network Crawlers

Mikhail Drobyshevskiy[1,2], Denis Aivazov[1,2], Denis Turdakov[1,3],
Alexander Yatskov[1], Maksim Varlamov[1] and Danil Shayhelislamov[2]

1. *Ivannikov Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia*
2. *Moscow Institute of Physics and Technology (State University), Moscow, Russia*
3. *Lomonosov Moscow State University, Moscow, Russia*

Plan

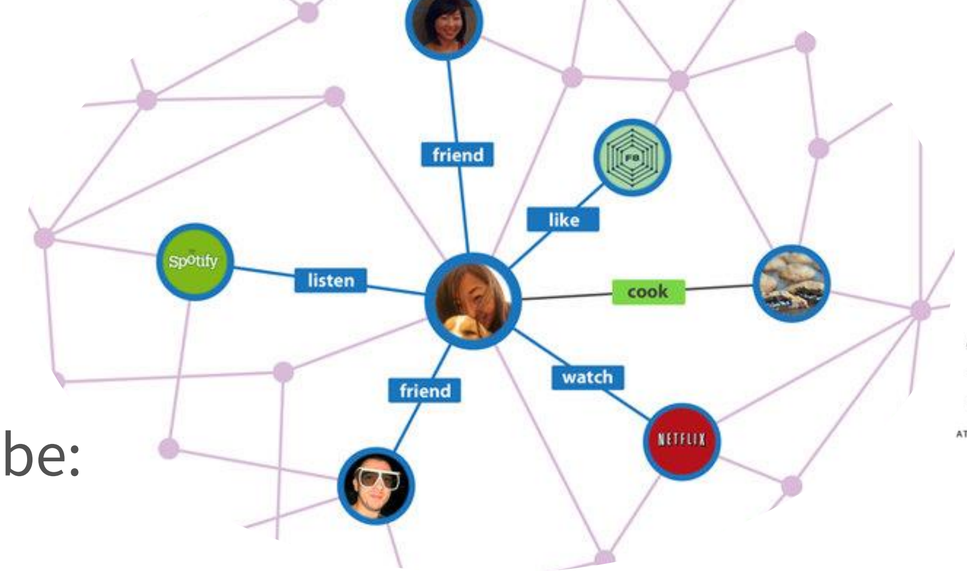
- Intro
- Types of influenced nodes
- Main crawling methods
- Comparison
- Experiment results

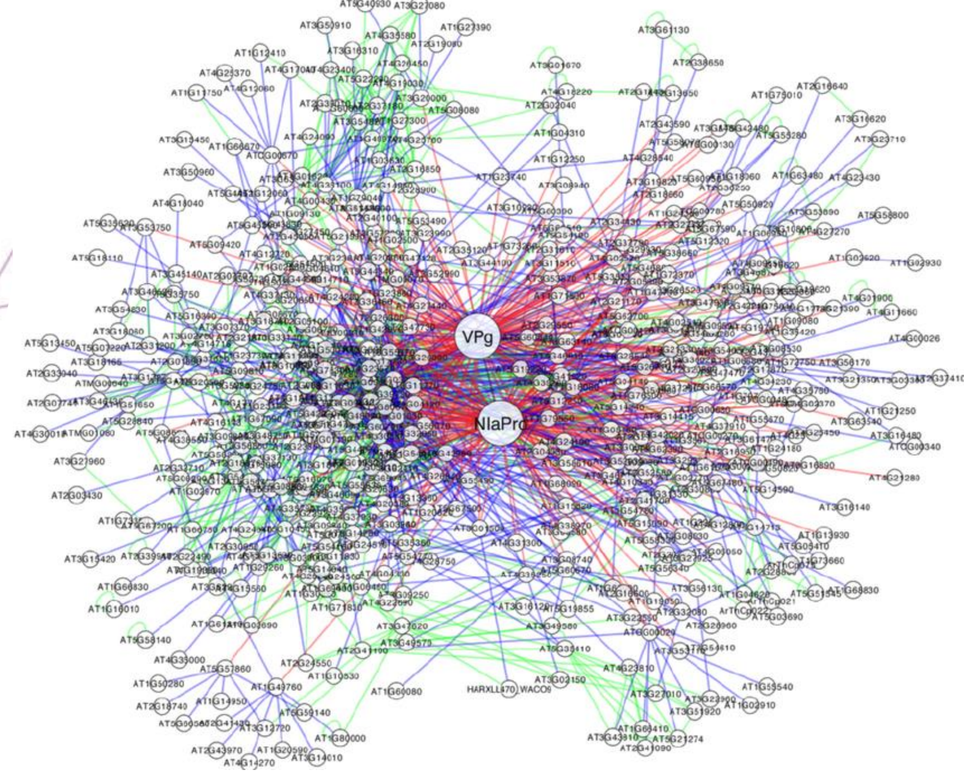


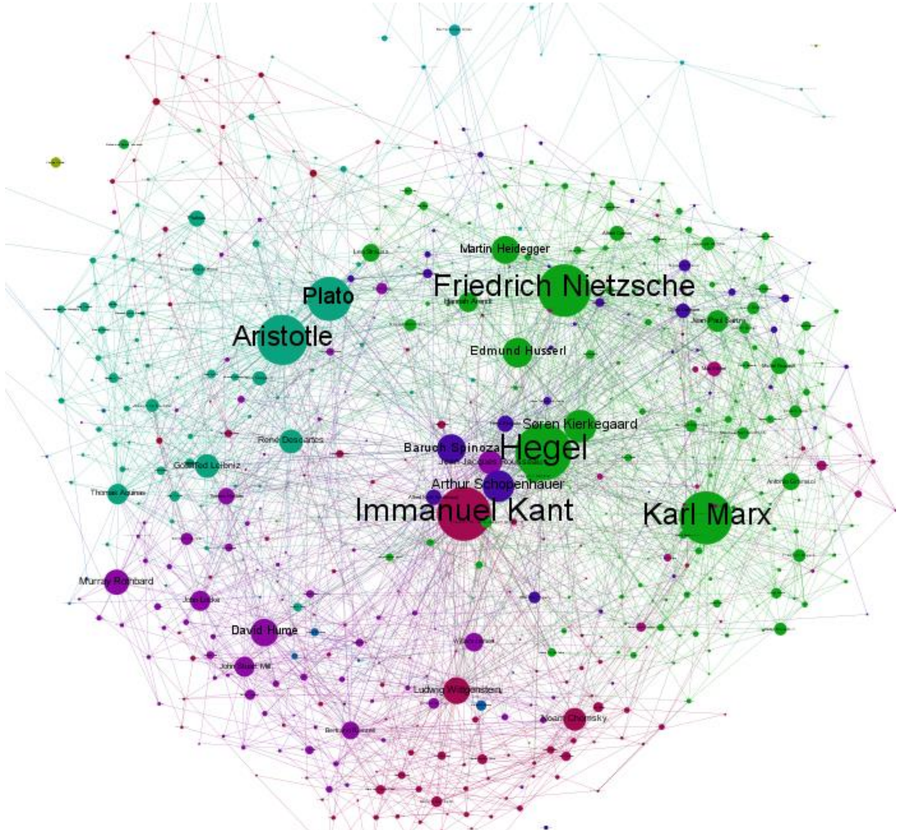
Graphs

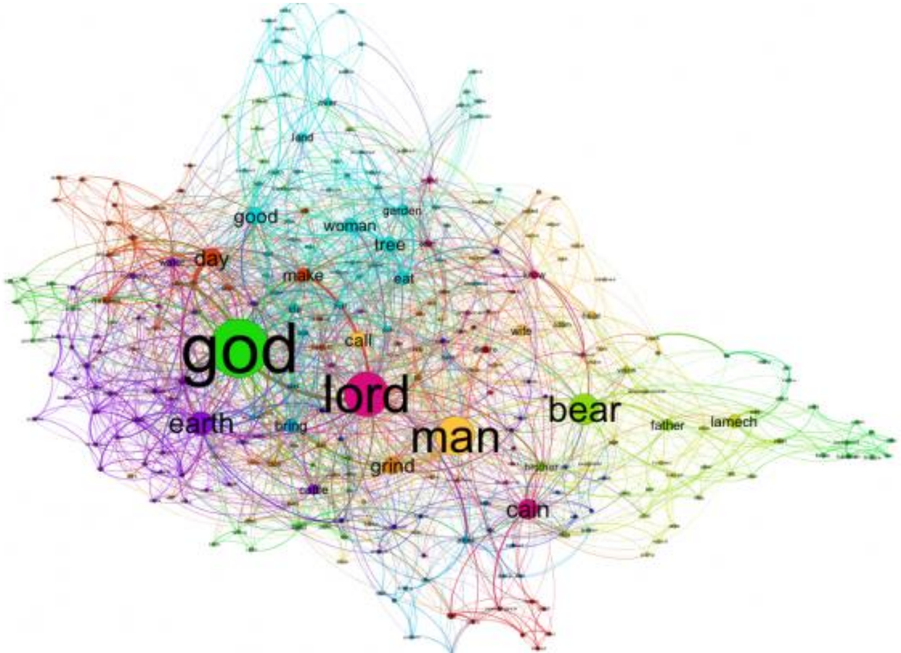
Can be used to describe:

- social connections
- cites
- proteins
- bank transaction
- web graphs
- host graphs
- word graphs
- etc.









Crawling process itself

Crawling - is a process of collecting graph

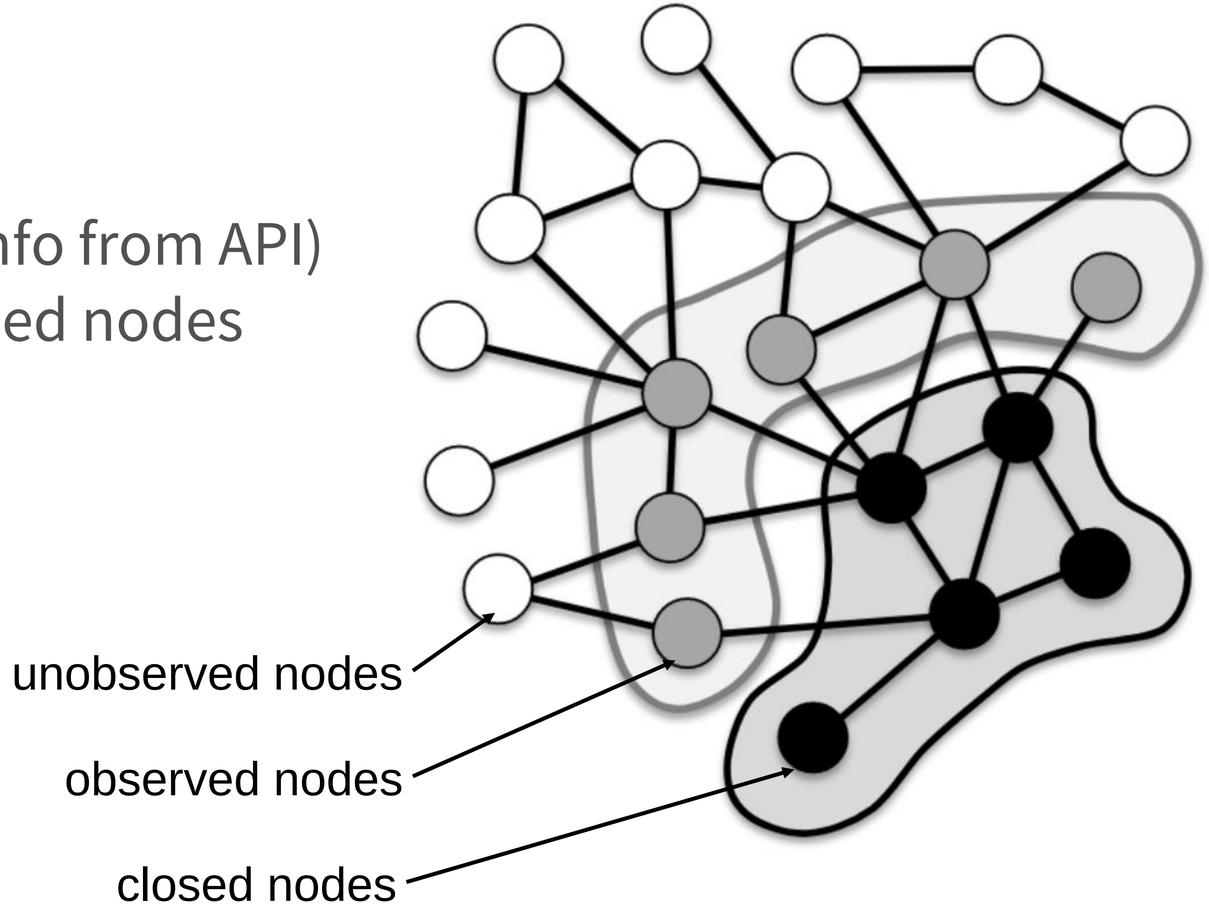
- To crawl node = to close node (get node info from API)
- Observe only friends (connections) of closed nodes

We need influencers (top 10% nodes) for:

- Adverts & Media
- Finding bottlenecks
- Understating the situation
- Making out zones of control

Constraint: bandwidth limit of API

=> need to find the fastest algorithm

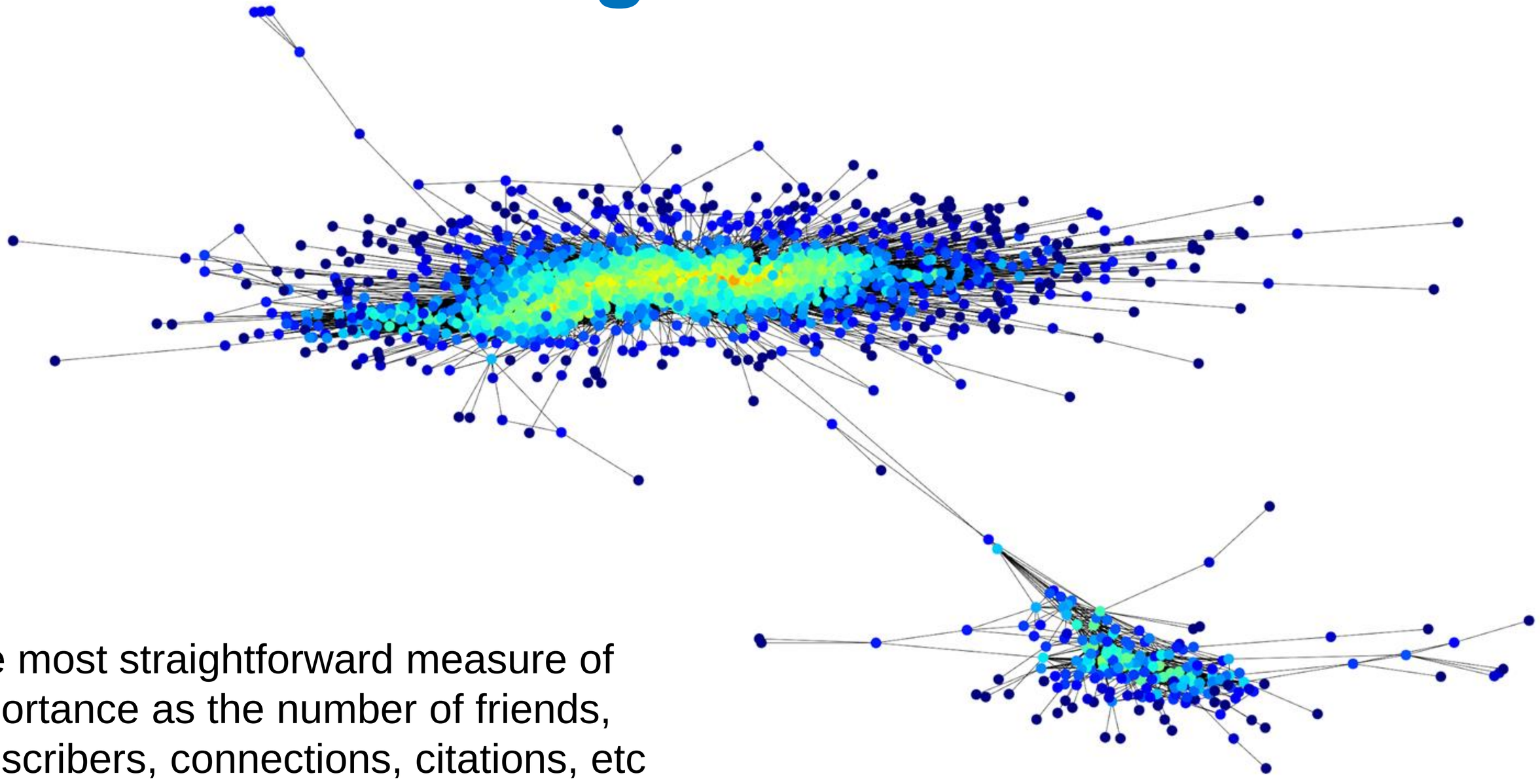


Picture from K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, “Pay few, influence most: Online myopic network covering,” // Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014, pp. 813–818.

Statement of the problem

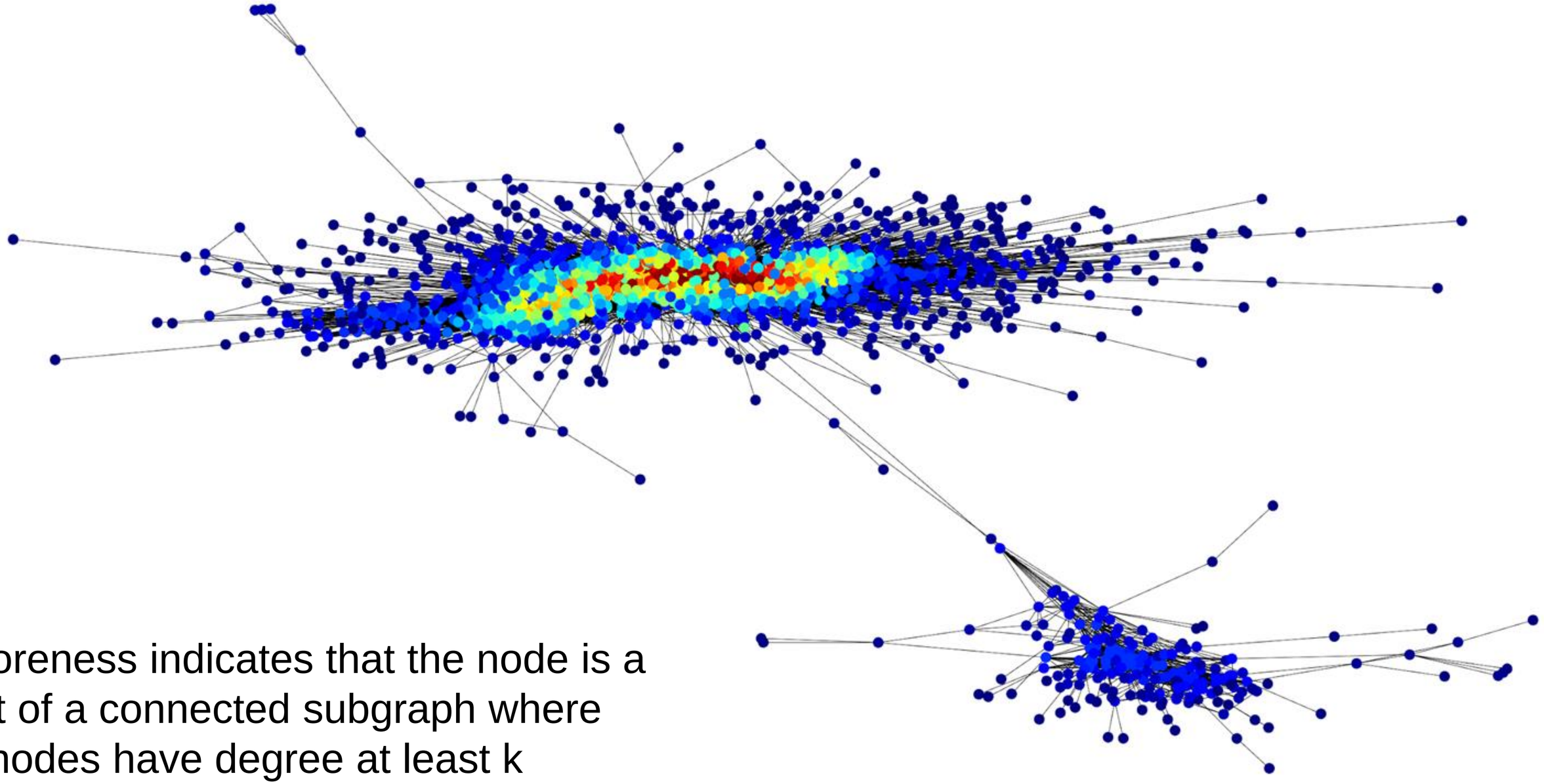
- ☐ Analyze existing crawling methods and implement them in framework
- ☐ Select different networks for crawling
- ☐ Choose how to measure “influence” of nodes
- ☐ Select a metric for comparison
- ☐ Handle the experiment and compare
- ☐ Repeat and prove (or disprove) experiments with crawling all nodes from graph

Influencers: degree



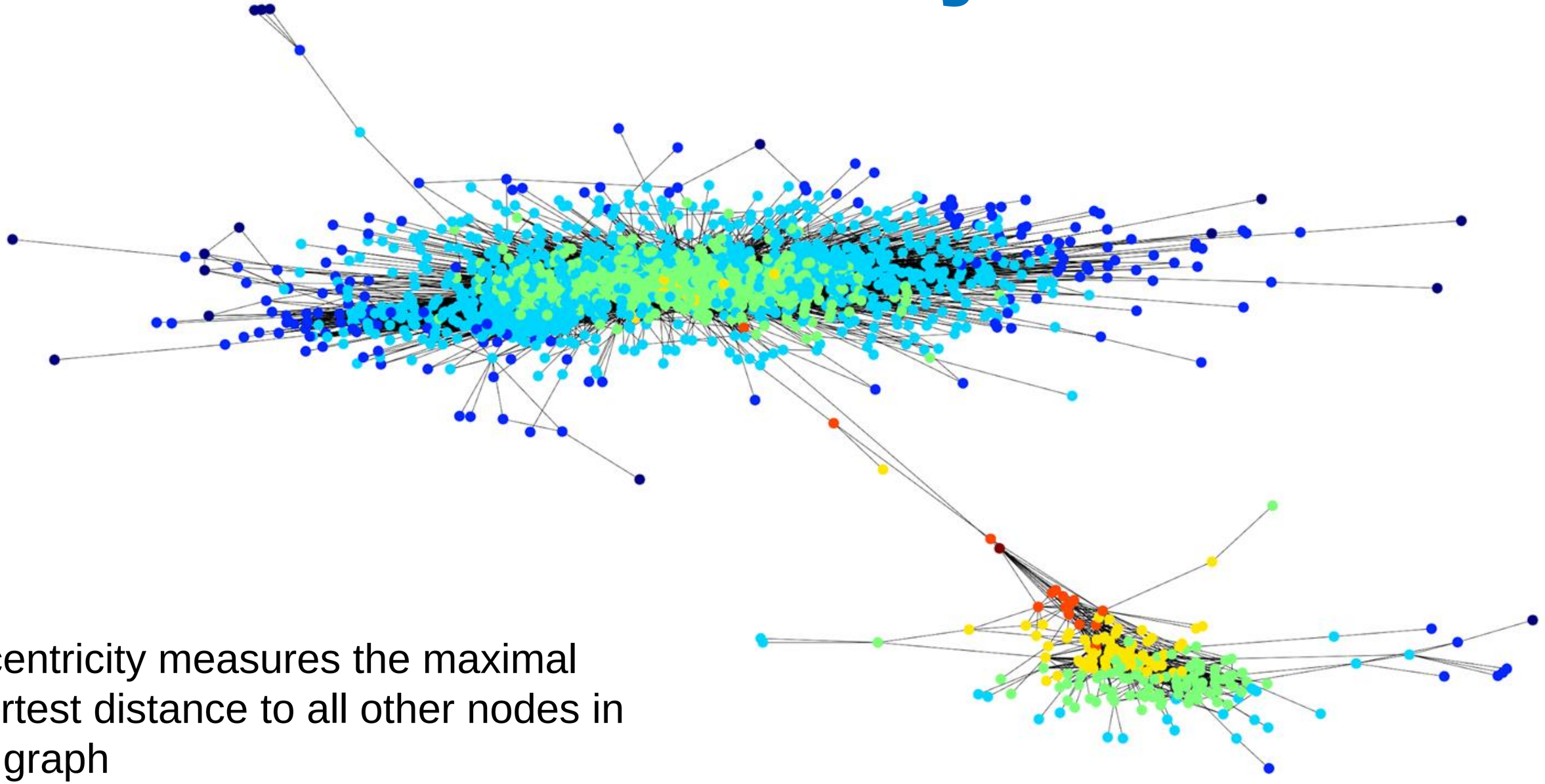
The most straightforward measure of importance as the number of friends, subscribers, connections, citations, etc

Influencers: k-coreness



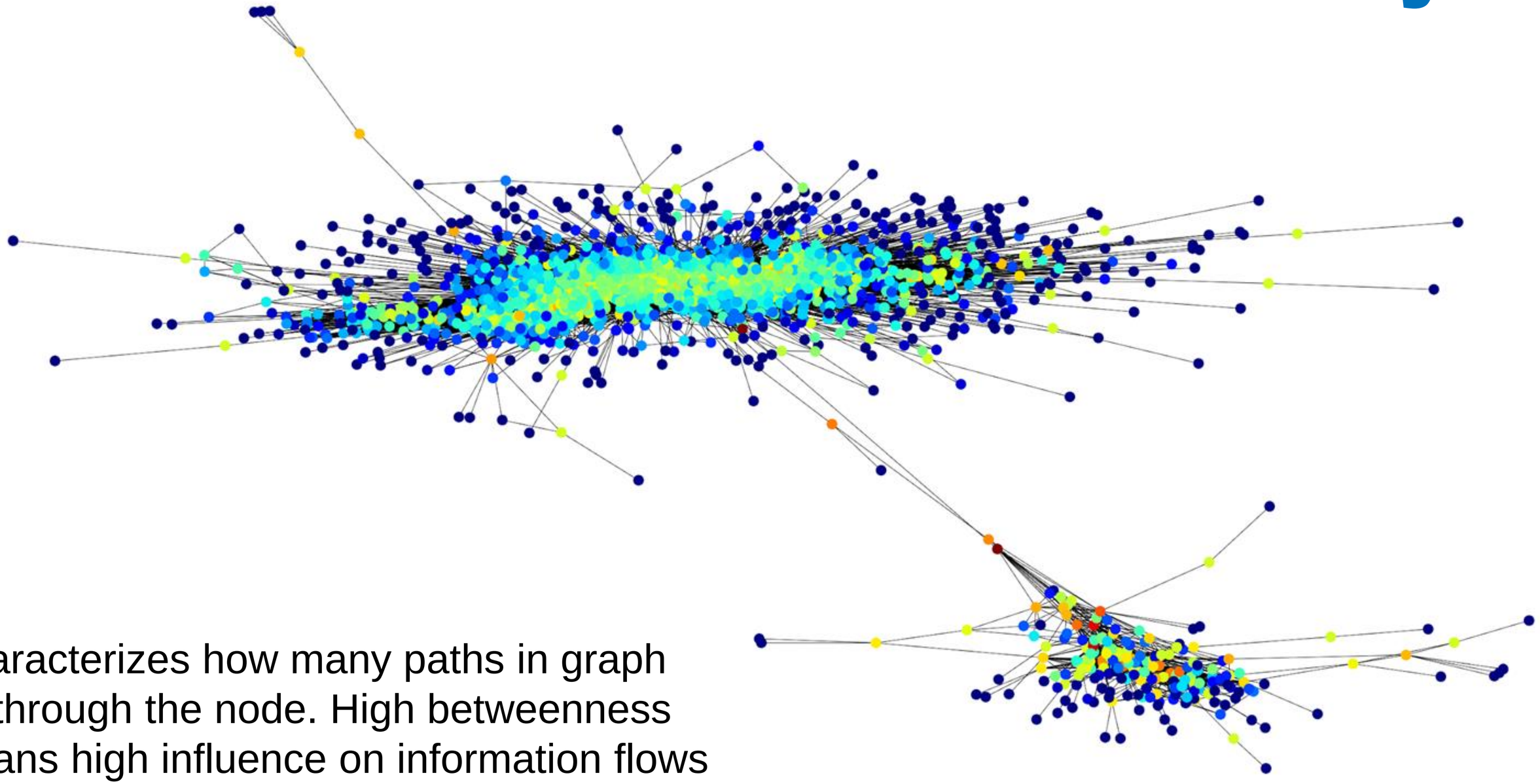
k-coreness indicates that the node is a part of a connected subgraph where all nodes have degree at least k

Influencers: eccentricity



Eccentricity measures the maximal shortest distance to all other nodes in the graph

Influencers: betweenness centrality

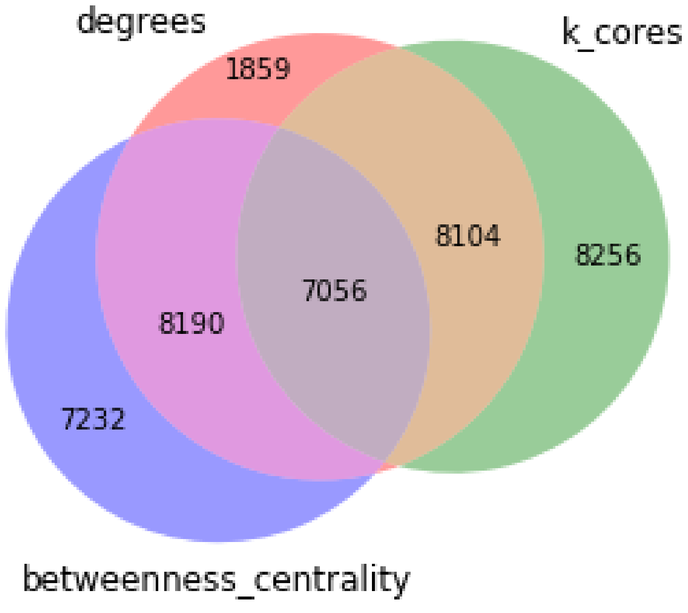


Characterizes how many paths in graph go through the node. High betweenness means high influence on information flows

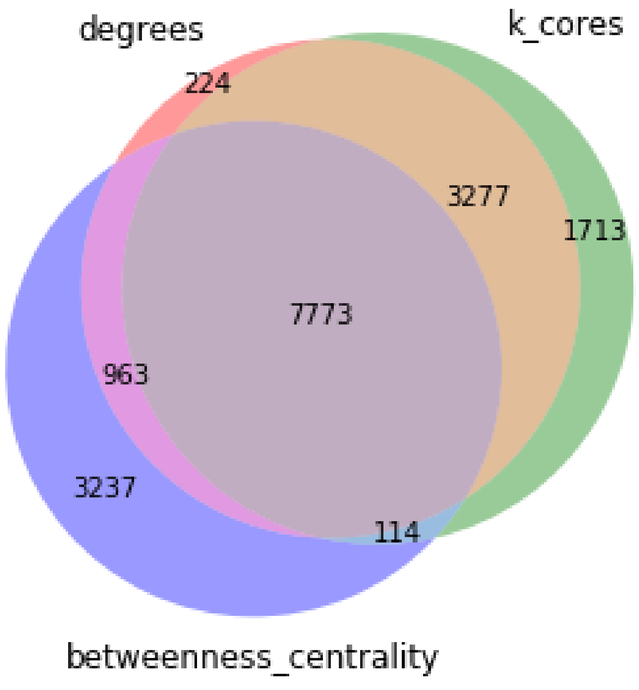
Correlation between influencers

Comparing sets of top-10% of nodes with Venn's diagram

DBLP 2010



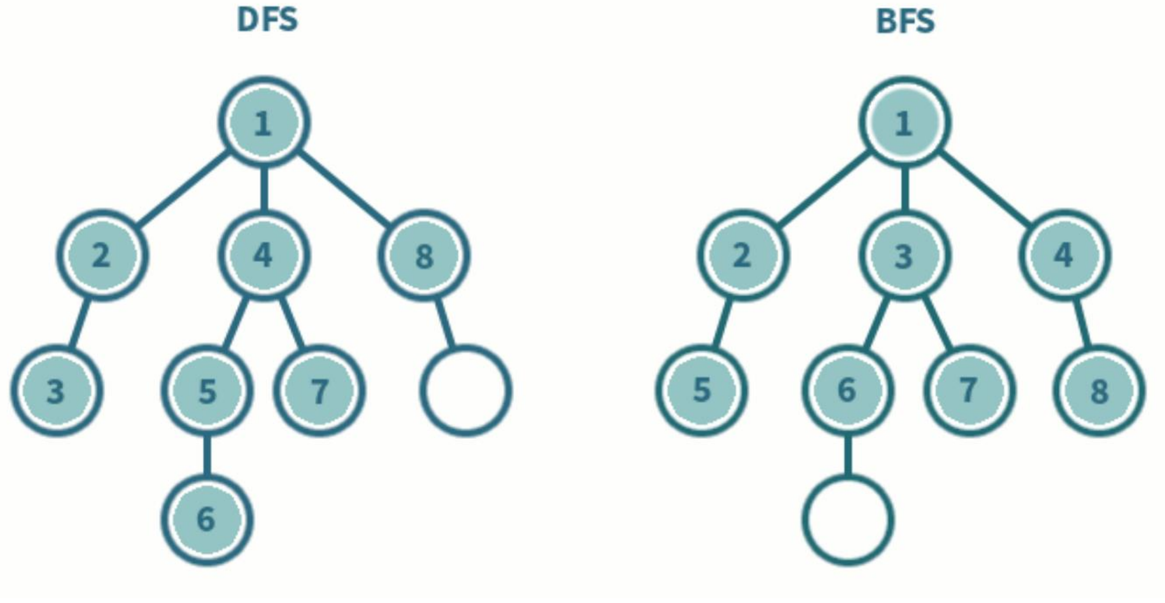
Github



Crawling algorithms 1

Traversal algorithms[1],[2]:

- (RC) Random Crawling - selecting random node from V_observed
- (RW) Random Walk - selecting random neighbour of previously crawled node
- (BFS) Breadth-first search
- (DFS) Depth-first search

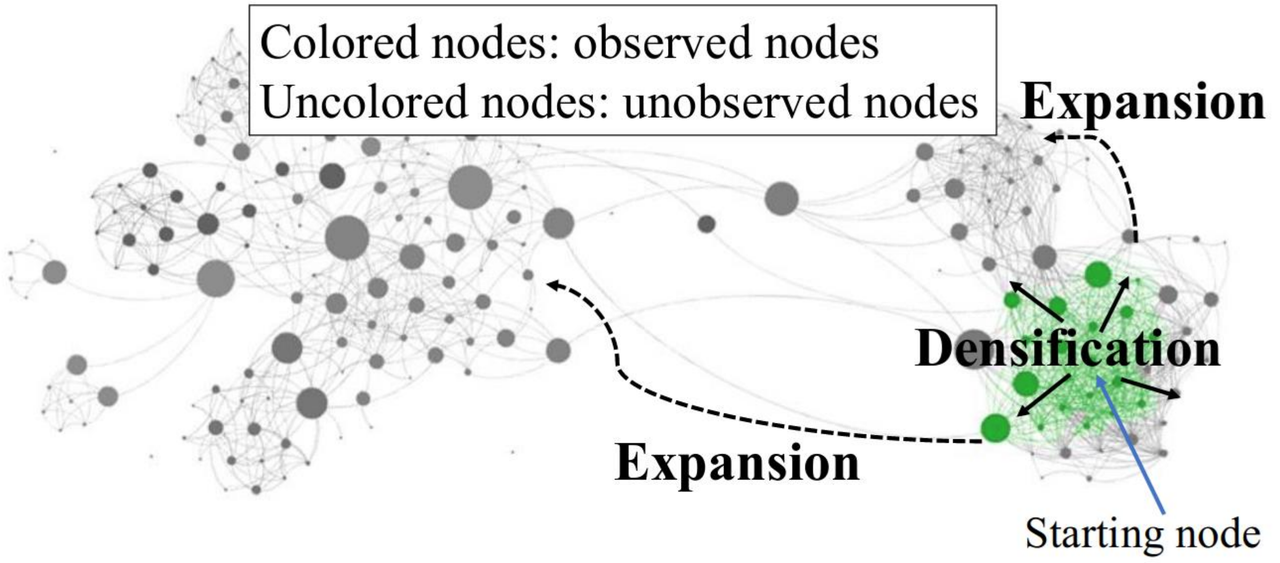


[1] S. Ye, J. Lang, and F. Wu, “Crawling online social graphs,” // 12th International Asia-Pacific Web Conference. IEEE, 2010, pp. 236–242.
[2] K. Areekijseree, R. Laishram, and S. Soundarajan, “Guidelines for online network crawling: A study of data collection approaches and network properties,” // Proceedings of the 10th ACM Conference on Web Science. ACM, 2018, pp. 57–66.

Crawling algorithms 2

Node-properties algorithms:

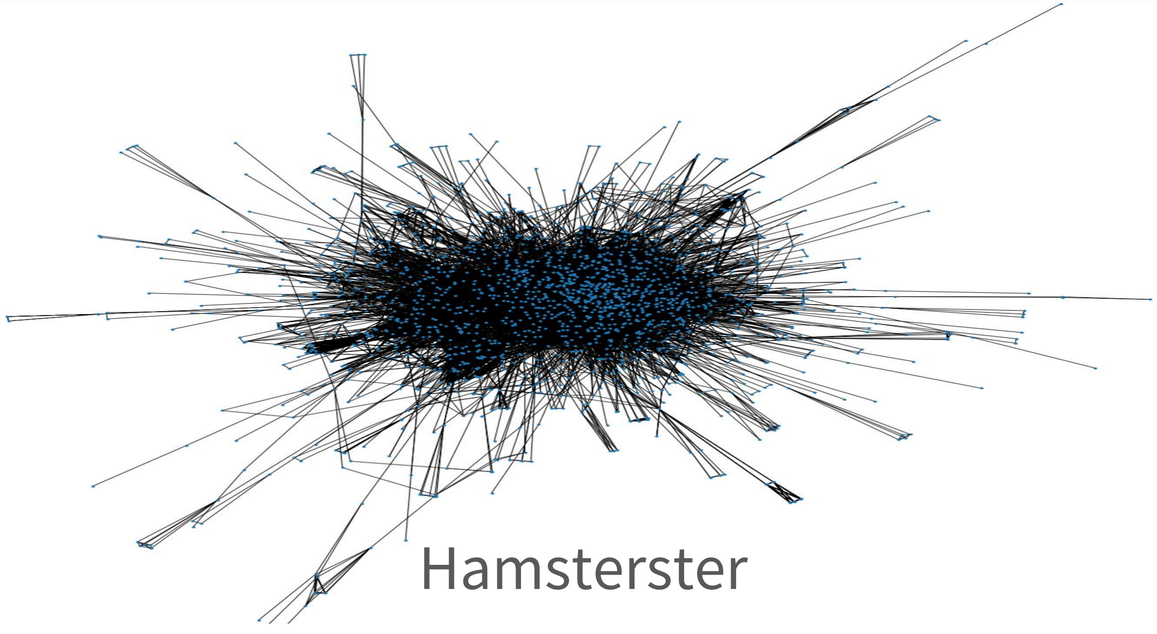
- (MOD[3]) Maximum Observed Degree - from observed nodes selects one with largest degree
- (DE[4]) Densification-Expansion - switching between RW and MOD analogues depending on calculated statistics

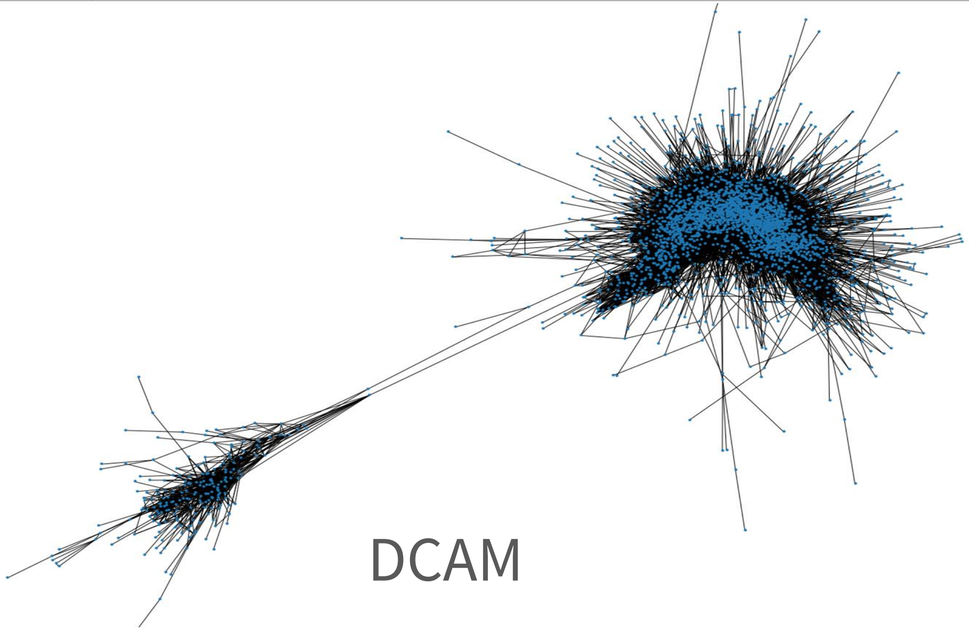


[3] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, “Pay few, influence most: Online myopic network covering,” // Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014, pp. 813–818.
[4] K. Areekijseree and S. Soundarajan, “De-crawler: A densification-expansion algorithm for online data collection,” // ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2018, pp. 164–169.

Datasets

Name*	Number of nodes	Number of edges	Avg. degree	Description
Hamsterster	2000	16097	16	friend graph in Hamsterster social network
DCAM**	2752	68741	50	community subgraph from VKontakte
Slashdot	51083	131175	5.1	friendship data of Facebook users in 2009
Facebook 2009	63392	816886	26	reply network of technology website SlashDot
Gihub	120865	439858	7.3	membership network of the GitHub
DBLP2010	226413	716460	6.3	co-authorship network



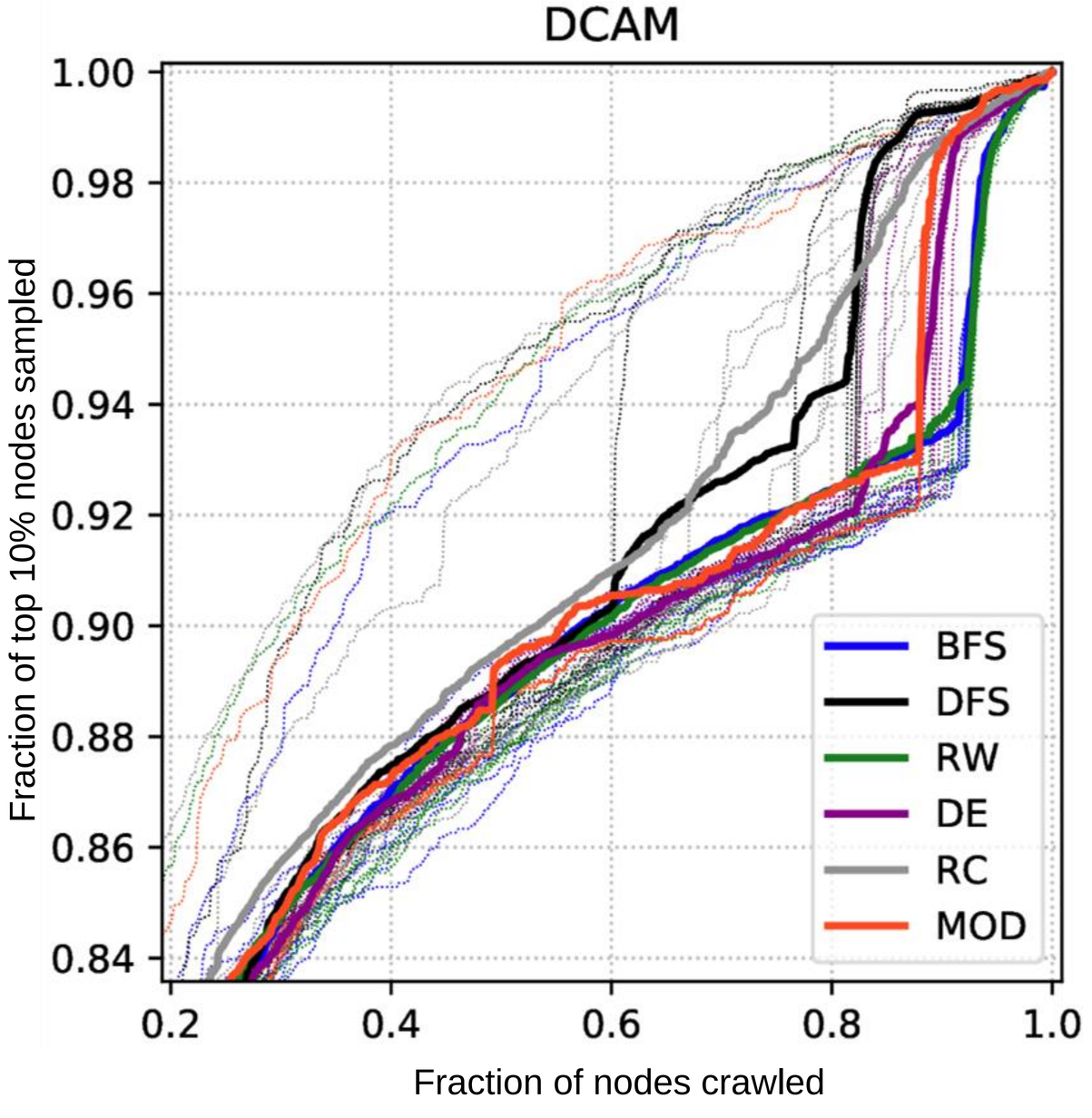


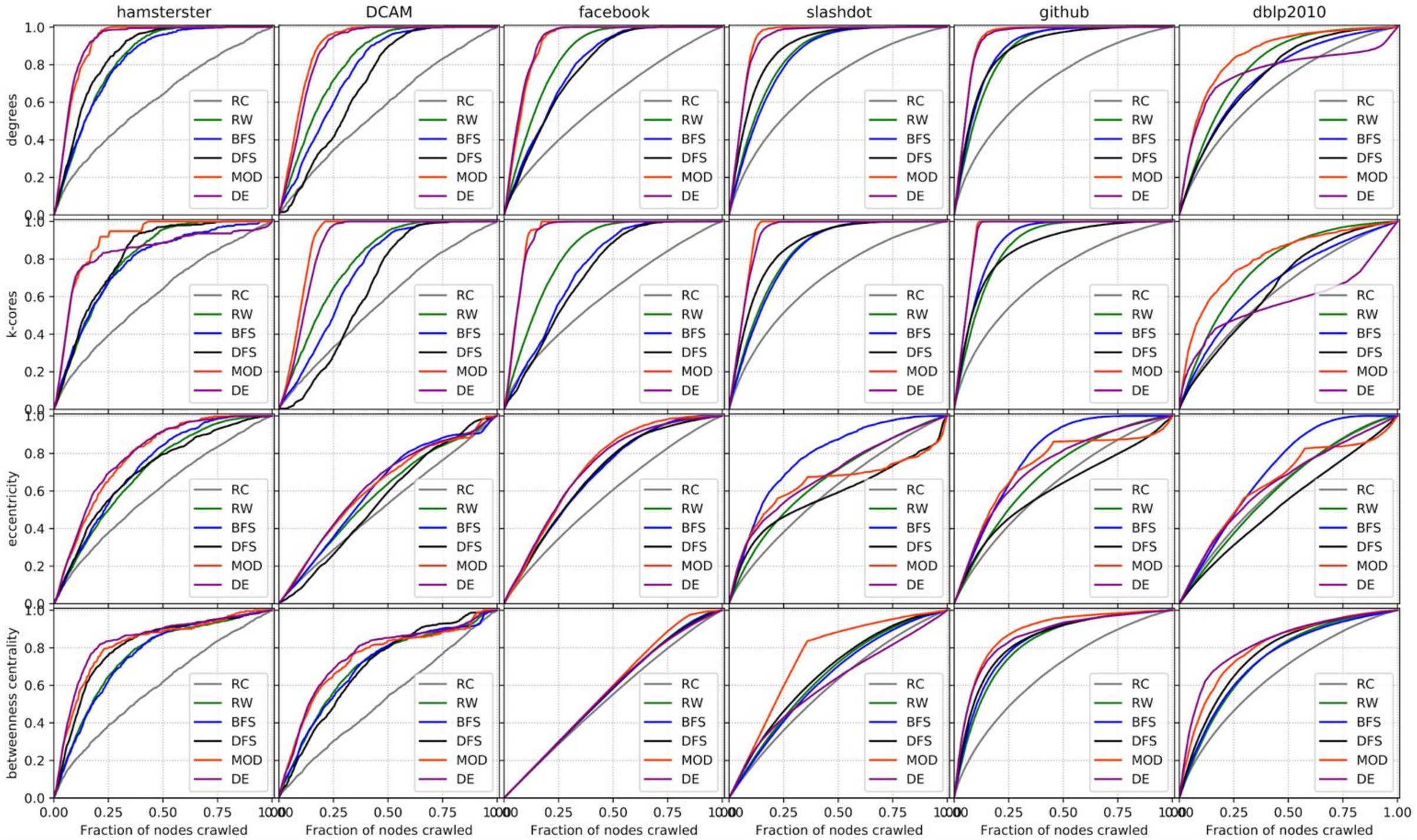
*All graphs were downloaded from http://networkrepository.com/ , ** except DCAM, which we crawled from https://vk.com/

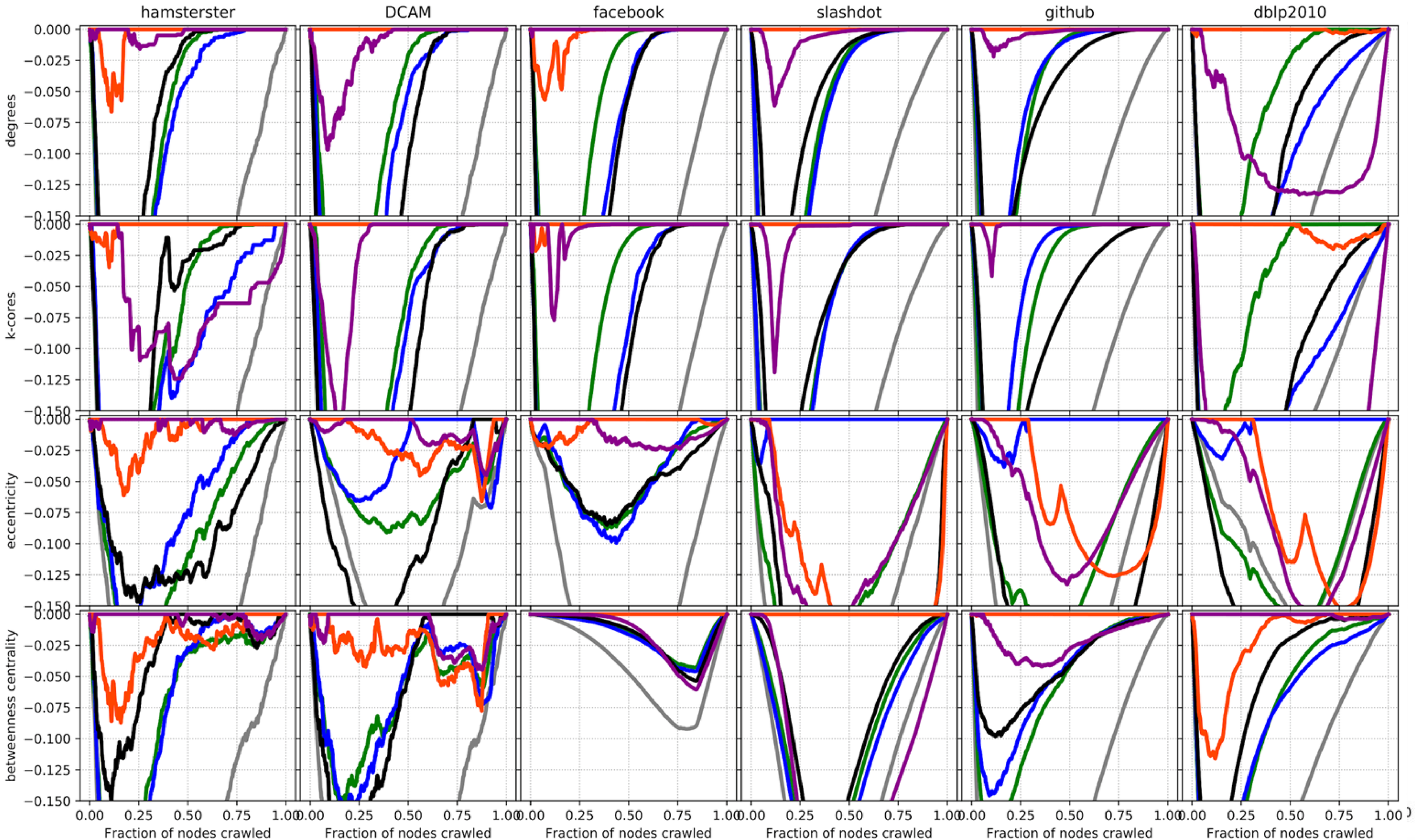
Methodology

Main steps:

1. For every graph we fix set of top nodes
2. Run algorithm concurrently from 8 starting nodes (seeds)
3. Building a chart, showing how # of founded nodes (y axis) in every set depends on # of queries to API (x axis)
4. For quality metric we take AUC of collected nodes

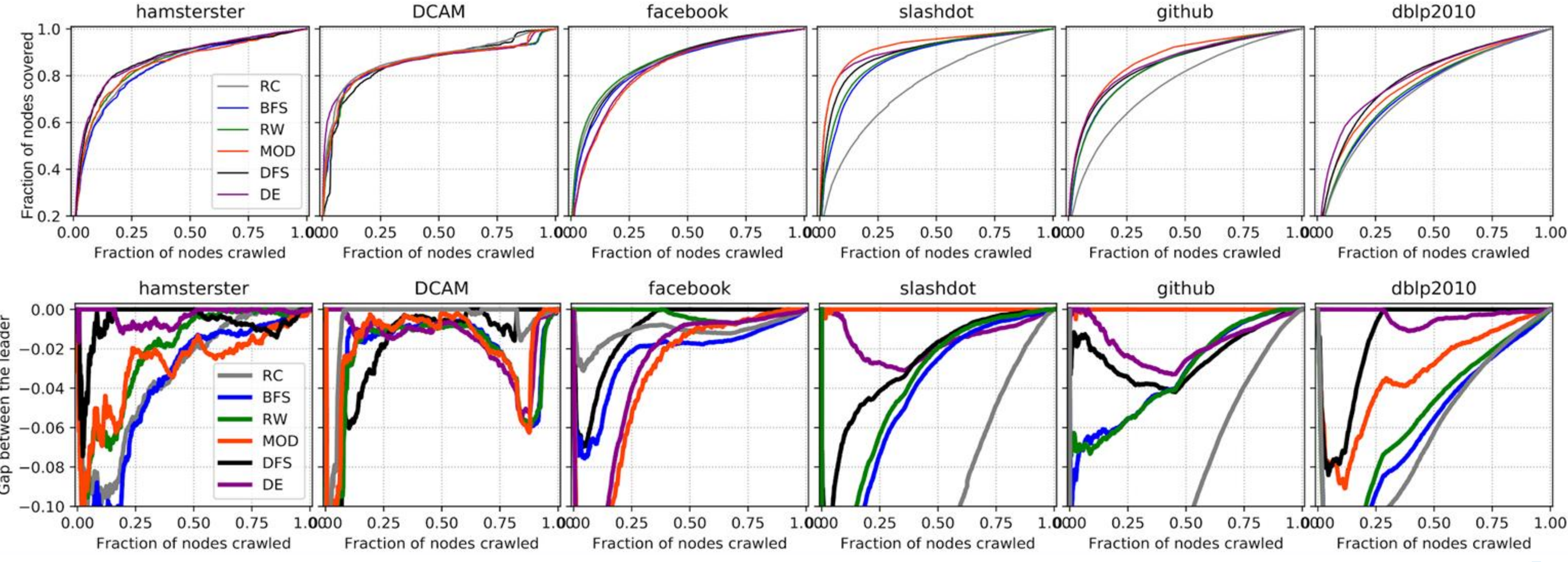



hamsterster
DCAM
facebook
slashdot
github
dblp2010
degrees
k-cores
eccentricity
betweenness centrality
RC
RW
BFS
DFS
MOD
DE
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.25
0.50
0.75
1.00
Fraction of nodes crawled


hamsterster
DCAM
facebook
slashdot
github
dblp2010
degrees
k-cores
eccentricity
betweenness centrality
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
0.00
0.25
0.50
0.75
1.00
Fraction of nodes crawled

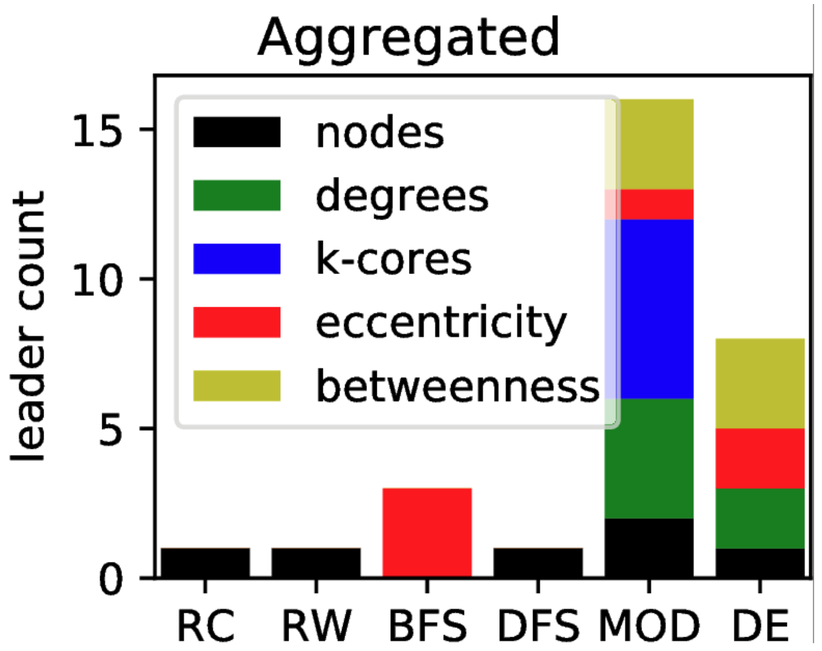
Repeating all-graph crawling

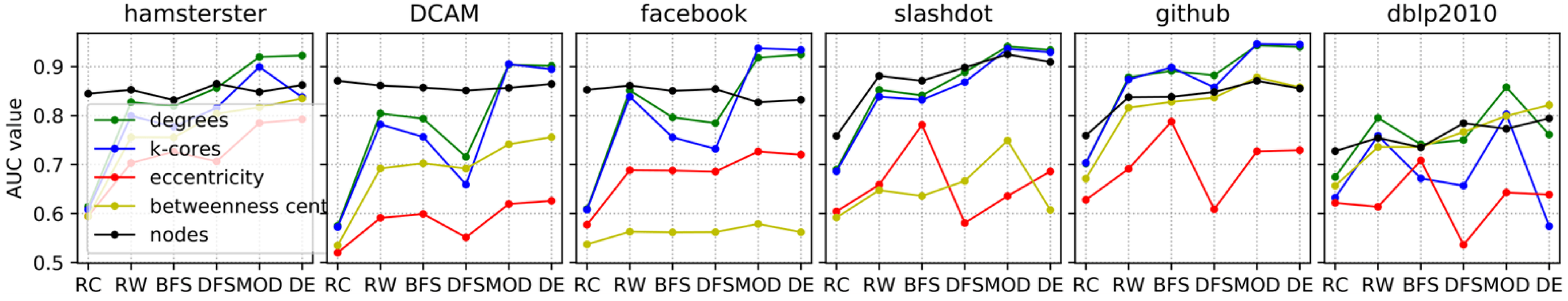
Also we repeated experiment with crawling whole graph and draw lag graph



Comparison

- MOD is best in most cases (proved[3])
- … even better than DE almost everywhere (disproved[4])
- Except several cases:
  - BFS for min-eccentric,
  - DE is good in finding degrees
  - all methods are good enough in all-graph coverage





[3] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, “Pay few, influence most: Online myopic network covering,” // Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2014, pp. 813–818.
[4] K. Areekijseree and S. Soundarajan, “De-crawler: A densification-expansion algorithm for online data collection,” // ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2018, pp. 164–169.

Results

☑ Analyze existing crawling methods and implement them in framework:

DFS, BFS, RW, RC, MOD, DE [1], [2]

☑ Select different networks for crawling (number of nodes):

Hamsterster^{2k}, DCAM^{3k}, Facebook2009^{63k}, Slashdot^{51k}, Github^{121k}, DBLP2010^{226k}

☑ Choose how to measure “influence” of nodes:

degrees, k-coreness, eccentricity, betweenness centrality

☑ Select a metric for comparison:

used AUCC (Area Under Crawling Curve)

☑ Handle the experiment and compare

☑ Repeat and prove (or disprove) experiments with crawling all nodes from graph