# Centralities: Capturing the Fuzzy Notion of Importance in Social Graphs

Erwan Le Merrer
INRIA Rennes, Bretagne Atlantique, France
elemerre@irisa.fr

Gilles Trédan
INRIA Rennes, Bretagne Atlantique, France
gilles.tredan@irisa.fr

## ABSTRACT

The increase of interest in the analysis of contemporary social networks, for both academic and economic reasons, has highlighted the inherent difficulties in handling large and complex structures. Among the tools provided by researchers for network analysis, the centrality notion, capturing the importance of individuals in a graph, is of particular interest. Despite many definitions and implementations of centrality, no clear advantage is given to a particular paradigm for the study of social network characteristics. In this paper we review, compare and highlight the strengths of different definitions of centralities in contemporary social networks.

## 1. INTRODUCTION

Thanks to abstractions provided by graph theory, today's interpersonal interactions can be captured and analyzed [28]. In all kinds of non-regular networks, nodes have a different *importance* due to a particular position in the structure. In a social network for example, this is demonstrated by the fact that some people or *actors* are far more important than others; those may know a significant part of the other network participants, and then play a hub role. They may also be in contact with people that are part of different communities, then link clusters and provide connectivity to the structure. They are crucial for the social graph characteristics, as their removal significantly stretches all distances between nodes in the structure (see *e.g.* [2]), or may even disconnect the graph in some critical cases [26].

Since the 1950s, researchers have developed the concept of individual importance in graphs, now known as the *centrality* notion [28]. In last few years, following the omnipresence of interaction graphs, centrality has seen a resurgence in interest. It is now used in wide research fields as *e.g.* leader detection in terrorists networks [26], placement of staff in military headquarters [4], or for the construction of overlay networks [29]. Centrality of a particular actor in a network is given by an algorithm; many methods have been introduced for this purpose, among them: *degree* centrality, *closeness* centrality [13], *eccentricity* [16], *eigenvector* centrality [6], random walk betweenness [23], *delta* [20] and *second order* centrality [19].

All those methods output proper results when applied to graphs. This is due to the fact that they all have their own definition of centrality; some give high importance to actors placed on network shortest-paths, whereas others weight based on proximity to other participants or number of neighbors. As no consensus exists on the definition of what properties a centrality algorithm should exhibit (except from simply assessing important actors), it turns out that studies using centralities use several centrality methods in parallel to compare their results [9, 23, 26]. The main reason for such a disparity is that there is no given data-set that the community interested in centrality notion has agreed upon, in order to provide a benchmark solution to compare centralities against (as done *e.g.* in the database community with well known data-sets). This is mainly because interaction graphs show many complex characteristics [2], and because centrality methods capture them in a different fashion.

In this paper, we consider the diversity of existing centrality algorithms and the complexity of interaction graphs. We review and give insight into their main strengths/weaknesses, when applied to social networks.

## 2. FORMS OF CENTRALITIES

We briefly define in this section the most used and well known centralities, as well as the recently introduced *second order* centrality.

Networks are representable in a graph form; let $G =< V, E >$ be an *undirected* graph (relation between actors are bidirectional) composed of vertices or *nodes* $V$ (here


*SNS'09,* March 31, 2009, Nuremberg, Germany.

the actors), and edges $E$ representing interactions between nodes. $G$ is composed of $n$ nodes ($n = |V|$) and $m$ edges ($m = |E|$). The *degree* of a node $i$, $d(i)$, is the number of edges adjacent to $i$; this neighborhood is noted $\Gamma(i)$. Let $d(i, j)$ denote the distance (shortest-path) between nodes $i$ and $j$ in the current graph. We consider *connected* graphs, which are graphs such that a *path* (*i.e.* a succession of edges) exists between any two nodes of $G$.

## 2.1 Degree Centrality

Starting with the simplest form of centrality, degree centrality assesses the importance of a node according to its degree in the interaction graph. We note $C_d(i) = d(i)$, the degree centrality of a node $i$. Albert et al. [2] show that in social interaction graphs of movie actor collaboration, science collaboration, phone calls or graphs of sexual contacts, the degree distribution follows a power-law. This indicates that some nodes are far more important than others (highly connected ones). Computing this centrality is often straightforward.

## 2.2 Closeness Centrality

Here important nodes are nodes close to all others in the graph. Practically, this is computable for a node $i$ by averaging the distance between $i$ and all other nodes $v$ in $G$; we note $C_c(i) = \frac{1}{\Sigma_{j \in V} d(i,j)}$.

## 2.3 Eccentricity

Eccentricity [16] now takes the notion of maximal distance between pairs of nodes, to compute their importance: $C_e(i) = \frac{1}{max_{j \in V} d(i,j)}$ for a node $i$. The intuition is that a node is central if its maximum distance to another node is close to the radius of the graph.

## 2.4 Eigenvector Centrality

Another method, proposed by Bonacich [6], is to consider the importance of neighbors of a node; in other words, an important node has important neighbors in the graph topology. Considering a node $i$, we then have $C_\lambda(i) = \Sigma_{j \in \Gamma(i)} C_\lambda(j)$. Google's *pagerank* algorithm is currently using a variant of eigenvector centrality [25].

## 2.5 Betweenness Centrality

Freeman [12] defined *betweenness* centrality by the ratio of the number of shortest-paths that a node is part of, over all graph shortest-paths. Here important nodes lie on a significant part of graph's shortest-paths. For a node $i$, we write $C_b(i) = \Sigma_{j \neq k \neq i} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$, with $\sigma_{jk}(i)$ being the number of shortest-paths from node $j$ to node $k$ that passes on node $i$.

Brandes [7] lowers the complexity of computation (from time complexity $\Theta(n^3)$ to $O(nm)$) by providing an approximation algorithm. Experimental studies [8, 14] consider practical complexity reduction for large graphs.

## 2.6 Random Walk Betweenness

Starting from the fact that important nodes do not mandatorily rely on shortest-paths (non-optimal nodes are also important, *e.g.*, for redundancy or network resilience), Newman proposed the use of *random walks* [23], in order to measure importance left out by shortest-path methods. The algorithm consists in launching a random walk from each node $j$ to every other node $k$. The random walk betweenness of a node $i$ is equal to the number of times that a random walk starting at $j$ and ending at $k$ passes through $i$ along the way, averaged by all possible $(j, k)$ pairs.

## 2.7 Second Order Centrality

Kermarrec et al. [19] builds a method based on Newman's paradigm, thus using a random process emulating information flow on the graph. The algorithm has been designed to be distributed among nodes, so that no global graph knowledge is needed. It is based on the regularity of visits of a random walk on a particular node, to assess that node's importance. One single random walk is running on the graph; at each visit on a node $i$, $i$ records the return time of this walk (number of steps since last visit to i); formally, if $\Xi_i(k)$ is the $k^{th}$ return time at node $i$, after $N$ visits, a node $i$ computes

$$C_{\sigma_i}(N) = \sqrt{\frac{1}{N-1}\Sigma_{k=1}^{N}\Xi_i(k)^2 - [\frac{1}{N-1}\Sigma_{k=1}^{N}\Xi_i(k)]^2},$$

with $\Xi_i$ being a table recording all return times at node $i$. It turns out that an important node has a low standard deviation $C_\sigma$ of those return times, compared to other graph nodes.

# 3. APPLICABILITY OF CENTRALITIES

This section highlights and discuss key properties of centralities, when applied to social networks.

## 3.1 Importance Ordering

Users expect a centrality algorithm to provide values that *correctly* reflect the order of nodes according to their importance. Unfortunately, centrality is a very abstract notion, and as we have shown in previous section, definitions differ in practice. Correlations can be observed in many cases, particularly for social networks (see *e.g.* [17] for overlap between shortest-path betweenness and degree centrality, and [23] for random walk betweenness and shortest-path betweenness). This implies that when using a particular centrality method over an arbitrary graph (graph characteristics are unknown), the results are not necessarily the absolute correct ones *w.r.t.* other methods.

In the context of social networks, particularly critical nodes are often sought out for graph disruption [3], as in the case of action planning on opponent structure, as hacking [11] or attack on terrorist networks [26]. In

| Name | $2^{nd}$ Order | Degree | Betweeness | Eigenvector | R.W. Betweeness | Closeness |
|---|---|---|---|---|---|---|
| Time | $O(n^3)$ | $O(n^2)$ | A: $O(nm)$ [7] | $O(n^3)$ | $O((m+n)n^2)$ | A: $O(\frac{\log n}{\epsilon^2}(n \log n + m))$ [10] |
| Distrib. | Done | Direct | Only bridges [22] | Done | Needs full graph | Undone |

**Figure 1: Table of costs (matrix based; "A": approximation algorithm) and distributability**

this case study, the network is disconnected (at best), or the average distance between nodes is significantly increased (at least); the goal is then to affect at maximum the topological properties of the graph with a minimal number of removals. Betweenness is here a good candidate.

Another application is for crawling robots to find good start locations in the web to cover it within just few hops [11] (used for *e.g.* spam). Here the closeness and eigenvector centralities might be used.

Consider the famous Milgram's "six degrees of separation" experiment [27], where six hops on average are necessary to connect any two people in the USA. Those six hops represent here the shortest-path between the pair of chosen nodes; it is fair to give some relative importance to nodes close to that shortest-path, to be able to handle human failures in this chain, with a limited stretch compared to that optimal path. Non-optimal path are taken into account with random walk between and second order centrality.

Those three examples ask for different solutions, as their use is application-dependant; in this case only experience allows to cleverly select the most relevant centrality for a target graph. Thus, insights have to be taken into account, instead of relying on a hypothetical absolute sort of nodes' importance.

## 3.2 Cost of Algorithms

Current social networks are very large, with potentially millions of users (Facebook and MySpace respectively count 132 and 117 million users , source: comScore, Jun-2008). Naive algorithms would be too slow to operate on graphs of this size; algorithmic costs cannot be neglected when choosing and implementing algorithms for social networks.

A lot of centrality implementations require time for completion scaling like $O(n^3)$ (*e.g.*, betweenness centrality [12], random walk betweenness and second order centrality); if operations are time consuming, this is cumbersome even for supercomputers. A first solution is then to consider approximation algorithms; unfortunately, current generic methods are far from logarithmic or even linear complexity (see *e.g.* [7, 14, 8, 5]). Eppstein et al. [10] provide an approximation of closeness centrality in time $O(m)$ for graph exhibiting a *small world phenomenon*. Finally, Okamoto et al. [24] only compute the top $k$ nodes with highest closeness centrality in $O((k + n^{\frac{2}{3}}.\log^{\frac{1}{3}} n)(n \log n + m))$ time, under certain conditions. If a drop of complexity cannot be found from algorithms themselves, another solution may come from the data-set. Extrapolation, based on algorithm application on a reduced size graph, is achievable through an accurate and representative sampling of the base graph [21]. Figure 1 compares algorithm costs when computed from the adjacency matrix of the given graph, to provide a fair comparison. Note anyway that some costs can be drastically reduced by using clever heuristics while collecting the graph's data (*e.g.*, degrees of nodes can directly be recorded while crawling a network).

From another angle, targeting graphs of millions of nodes means that physical memory on computers is a killer for algorithms that need to build and make operations on graphs' adjacency matrix. This is related to space complexity; it is generally $O(n^2)$, or $O(n+m)$ for approximation algorithms as Brandes [7]. Abiteboul et al. [1] propose an eigenvector centrality method for page ranking, that does not need to store the matrix. Such a storage issue calls for distributed solutions, as pointed in next subsection.

## 3.3 Centralized VS Distributed

Following the natural development of computer science, centralities were mostly thought of as a centralized computational model (where a graph is captured and analysed offline). Some applications target a distributed execution of the centrality measurement: *(i)* when the network cannot be accessed on a transparent manner, *(ii)* when computation costs on the central server appears prohibitive (for CPU or storage), or *(iii)* when the evolving graph must be considered (online application). Degree centrality is directly distributable, as each graph node has to only be aware of its direct neighborhood $\Gamma$. In such a paradigm, memory use on a central point is replaced by the gathering of resources from multiple participating entities. Nanda et al. [22] propose a decentralized algorithm to detect nodes that bridge highly connected regions in a topology. Finally the second order centrality was designed to be distributed [19], and also comes along with a theoretical analysis for a centralized matrix form. Some historical centralities have no distributed version currently designed, probably because this was not a concern at the time; closeness, eccentricity and betweenness are implementable by computing all-pairs shortest-path distributedly (in optimal time $O(n)$ [18]), and then waiting for nodes to communicate results. Random walk betweenness requires each node to send a random walk

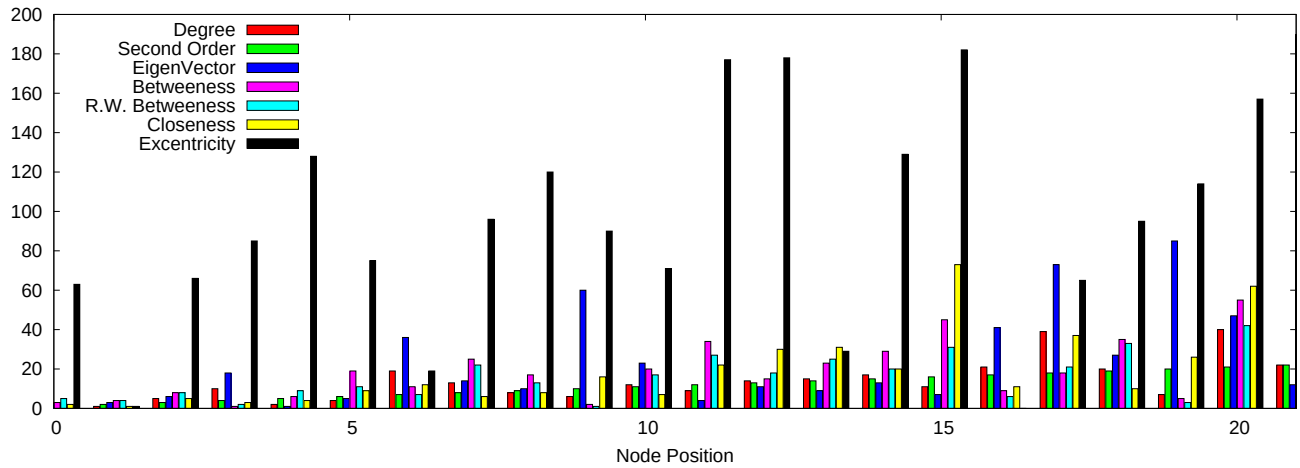


**Figure 2: Histogram of centralities: nodes sorted by ascending score of second order centrality**

to each other node, thus implying a full graph knowledge; this is simply achievable by network flooding for propagation of each node ID. In both trivial implementations, nodes should be able to store $O(n)$ IDs and/or informations concerning other nodes; a second concern is obviously network jamming created by initial knowledge extension. Ideally, a particular centrality then has a distributed algorithm to compute it, at a reasonable cost *w.r.t.* data-set size.

### 3.4 Online Reactivity

While some studies use static graphs for analysis after the graph snapshot for a particular event (*e.g.*, past disease spreading, or map of urban streets), some others may be interested in the evolution of the graph structure (ad-hoc networks, active social interactions). Some algorithm may be particularly fast for computation (core algorithm of Abiteboul et al. [1] for page ranking), while some other could be sensitive to dynamics and size (second order centrality needs to visit $O(n^3)$ nodes in its online version). Another important aspect is the reactivity of the centrality to small topological changes. For example, topological change affects only the degree centrality of involved nodes, whereas betweenness centrality has to be recomputed for all graph nodes. This calls for centrality implementations that could be responsive to graph changes (node removal or loss/addition of edges), *w.r.t.* the time it needs to produce a result.

## 4. TESTING A MUSICIAN NETWORK

In this section, we present experiments we ran to compare centralities and illustrate the differences of algorithms' outputs. In order to compare them, we implemented algorithms of centralities presented in Section 2. The experimental network is a 191 nodes network modeling the largest connected component of jazz players collaborations [15]. Figure 3 plots the degree distribution of the graph, showing that 10% of the graph nodes have more than 40 neighbors; such imbalance is representative of common social graphs [2].

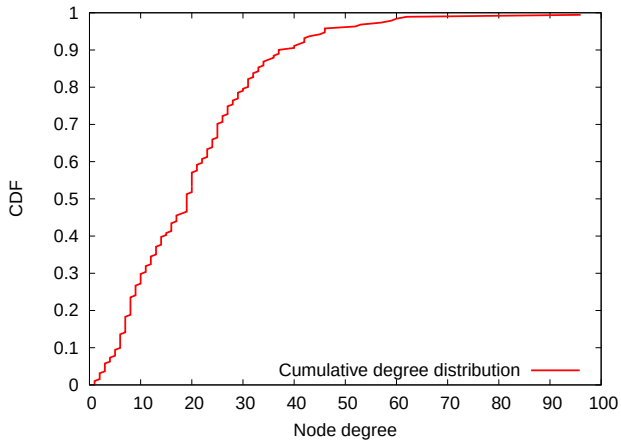


**Figure 3: CDF of degrees of the musician graph**

### 4.1 On Centrality Agreement

Figure 2 plots the centrality scores for the 40 more important nodes according to each centrality, sorted by second order centrality. For example a yellow bar ($x = 15, y = 70$) means that the node that has the $15^{th}$ highest second order centrality has the $70^{th}$ highest closeness centrality. Equivalent centrality output thus should plot results in a triangle shape ($x \equiv y$). The fuzziness of the figure clearly illustrates that some centralities are not correlated on this social graph example. More specifically, eccentricity is often very different than the second order centrality, while closeness and degree centralities achieve closer correlation. A closer look at the top-10 ranking nodes shows that, apart from eccentricity and - to a lesser extent - eigenvector, centralities agree on important rankings. This confirms that centralities produce different results in practical experiments; they do not capture the same graph char-

acteristics, which are often uncorrelated.

Eccentricity outputs very particular results; this centrality misses network core nodes (according to all other centralities), being attracted by long - and arguably less important - chains of nodes at the "edges" of the graph (recall that it considers max distance to any node).

## 4.2 Impact of Node Removal

One way of assessing the absolute importance of nodes given by centralities *w.r.t.* the topology is by removing highest ranked nodes, and observing the resulting graph. For a given centrality, we sequentially removed nodes starting from the most important remaining one, and then computed on the resulting graph (i) the relative size of the biggest connected component (Figure 4), and (ii) the average path length between all nodes belonging to this component (Figure 5). For example, considering ($x = 60$) for the degree centrality, we learn that the original graph minus the 60 highest degree nodes still connects 90% of the remaining 131 nodes, and that the average distance between two nodes belonging that connected part is around 3.8 hops.

As awaited [28], betweenness measures succeed to give importance to critical nodes for graph connectivity, by providing a very similar effect on resulting structure. A drop is observed for both centralities around 45 nodes removed (Figure 4); it represents the last removal before a first relatively large part of the graph is disconnected (here around 25% of nodes). This expresses their efficiency for identifying critical nodes for structural disruption. An advantage goes to random walk betweenness, which does not only consider optimal paths; the removal of the nodes it identifies damages alternative but yet centrality-important paths. At the same time, average route length drops, as main component size is reduced due to partition.

Second order centrality does not quickly disconnect large graph parts, but considerably stretches route lengths in the main component. It is somehow related to the random walk betweenness efficiency for non-optimal paths, but by being less critical, it smoothly targets relatively important nodes and then outputs a hardly navigable structure. Note that it ends up with the smallest component (around 15 nodes).

We can observe that eccentricity, degree, eigenvector and closeness do not significantly affect graph connectivity; meanwhile, node removal have different effects on route lengths. Degree centrality indeed stretches paths more importantly; it succeeds in sorting important nodes for route lengths, as this simulation actually removes topology hubs that play shortcut roles. Eccentricity behaves poorly in both scenarios, confirming its non-correlation with other measures, and its very particular applicability.

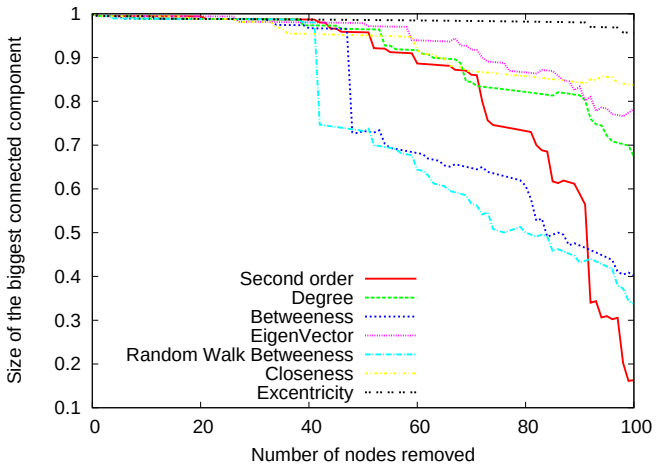


**Figure 4: Impact of removal on component size**

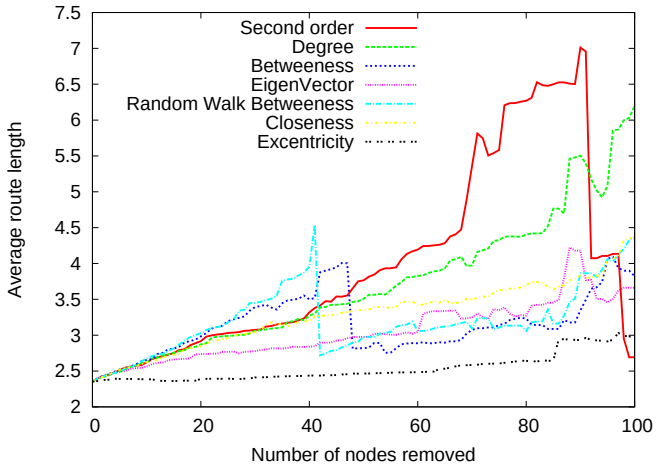


**Figure 5: Impact of removal on route length**

## 5. CONCLUSION

We reviewed in this paper the diverse notion of centrality. All proposed methods have their own concerns and strengths, showing again that centrality is a versatile tool to capture particular graph characteristics. Considering attributes of current social networks (particularly size and dynamicity), research towards applicable and efficient centrality implementations is yet of the utmost importance.

## 6. REFERENCES

[1] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 280–290, New York, NY, USA, 2003.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[3] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex

networks. *Nature*, 406:378, 2000.

[4] D. Anthony. Social network analysis in military headquarters using cavalier. In *5th International Command, Control Research and Technology Symposium*.

[5] David Bader, Shiva Kintali, Kamesh Madduri, and Milena Mihail. Approximating betweenness centrality. pages 124–137. 2007.

[6] P Bonacich. Factoring and weighting approaches to status scores and clique identification. *J. Math. Sociol. 2*, pages 113–120, 1972.

[7] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[8] Ulrik Brandes and Christian Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, pages 2303–2318, 2007.

[9] Tim Dwyer, Seok-Hee Hong, Dirk Koschützki, Falk Schreiber, and Kai Xu. Visual analysis of network centralities. In *APVis '06: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation*, pages 189–197, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[10] David Eppstein and Joseph Wang. Fast approximation of centrality. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 228–229, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[11] Dániel Fogaras. Where to start browsing the web. In *IICS: nnovative internet community systems*, pages 65–79. Springer-Verlag, 2003.

[12] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, March 1977.

[13] Linton C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1(3):215–239, 1979.

[14] Robert Geisberger, Peter Sanders, and Dominik Schultes. Better approximation of betweenness centrality. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2008.

[15] Pablo Gleiser and Leon Danon. Community structure in jazz. *Advances in Complex Systems*, 6:565, 2003.

[16] Per Hage and Frank Harary. Eccentricity and centrality in networks. *Social Networks*, 17(1):57 – 63, 1995.

[17] K. i. Goh, E. Oh, B. Kahng, and D. Kim. Betweenness centrality correlation in social networks. *Phys. Rev. E*, 67:01710–1, 2003.

[18] Saroja Kanchi and David Vineyard. An optimal distributed algorithm for all-pairs shortest-path. *Information theories and applications*, 11:141–146, 2004.

[19] A.-M. Kermarrec, E. Le Merrer, B. Sericola, and G. Trédan. Rr-6809 inria - second order centrality:distributed assessment of nodes criticity in complex networks, 2009.

[20] V. Latora and M. Marchiori. A measure of centrality based on network efficiency. *New Journal of Physics*, 9(188), 2007.

[21] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, New York, NY, USA, 2006. ACM.

[22] Soumendra Nanda and David Kotz. Localized bridging centrality for distributed network analysis. In *Proceedings of the 17th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6, August 2008.

[23] Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, January 2005.

[24] Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. Ranking of closeness centrality for large-scale social networks. In *FAW '08: Proceedings of the 2nd annual international workshop on Frontiers in Algorithmics*, pages 186–195, Berlin, Heidelberg, 2008. Springer-Verlag.

[25] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[26] Muhammad Akram Shaikh, Jiaxin Wang, Zehong Yang, and Yixu Song. Graph structural mining in terrorist networks. In *ADMA '07: Proceedings of the 3rd international conference on Advanced Data Mining and Applications*, pages 570–577, Berlin, Heidelberg, 2007. Springer-Verlag.

[27] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[28] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.

[29] Eiko Yoneki, Pan Hui, ShuYan Chan, and Jon Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 225–234, New York, NY, USA, 2007. ACM.