Soc. Netw. Anal. Min. (2013) 3:899–914
DOI 10.1007/s13278-012-0076-6

ORIGINAL ARTICLE

Identifying high betweenness centrality nodes in large social networks

Nicolas Kourtellis · Tharaka Alahakoon · Ramanuja Simha · Adriana Iamnitchi · Rahul Tripathi

Received: 20 January 2012 / Revised: 21 May 2012 / Accepted: 30 May 2012 / Published online: 5 July 2012

Abstract This paper proposes an alternative way to identify nodes with high betweenness centrality. It introduces a new metric, κ-path centrality, and a randomized algorithm for estimating it, and shows empirically that nodes with high κ-path centrality have high node betweenness centrality. The randomized algorithm runs in time $O(\kappa^3 n^{2-2\alpha} \log n)$ and outputs, for each vertex v, an estimate of its κ-path centrality up to additive error of $\pm n^{1/2+\alpha}$ with probability $1 - 1/n^2$. Experimental evaluations on real and synthetic social networks show improved accuracy in detecting high betweenness centrality nodes and significantly reduced execution time when compared with existing randomized algorithms.

Keywords Betweenness centrality · Social network analysis · Algorithms · Experimental evaluation

N. Kourtellis (✉) · A. Iamnitchi · R. Tripathi
Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA
e-mail: nkourtel@mail.usf.edu

A. Iamnitchi
e-mail: anda@cse.usf.edu

R. Tripathi
e-mail: tripathi@cse.usf.edu

T. Alahakoon
2056 Pinnacle Pointe Drive, Norcross, GA 30071, USA
e-mail: alahakoo@mail.usf.edu

R. Simha
Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA
e-mail: rsimha@mail.usf.edu

1 Introduction

Social network analysis tools have been used in various fields such as physics, biology, genomics, anthropology, economics, organizational studies, psychology, and IT. The recent phenomenal growth of online social networks exacerbates the need for such tools that are scalable for applications in military, government, and for commercial purposes, to name only a few. Some of the relevant network metrics are local, such as degree centrality, while others capture global structural properties of the graph, such as the *betweenness centrality*. This important global graph metric is a centrality index that quantifies the importance of a node or an edge as a function of the number of shortest paths that traverse it.

Node betweenness centrality is relevant to problems such as identifying important nodes that control flows of information between separate parts of the network and identifying causal nodes to influence other entities behavior, such as genes in genomics or customers in marketing studies. Betweenness centrality has been used to: analyze social networks (Kahng et al. 2003, Liljeros et al. 2001, Ortiz et al. 2004, Said et al. 2008) and protein networks (Jeong et al. 2001); identify significant nodes in wireless ad hoc networks (Maglaras and Katsaros 2011); study the importance and activity of nodes in mobile phone call networks (Catanese et al. 2012) and interaction patterns of players on massively multiplayer online games (Ang 2011); study online expertise sharing communities such as physicians (Hua and Haughton 2012); identify and analyze linking behavior of key bloggers in dynamic networks of blog posts (Macskassy 2011); and measure network traffic in communication networks (Singh and Gupte 2005).

Node betweenness centrality, however, is computationally expensive. The best known algorithm for computing

exact betweenness centrality of all vertices is Brandes' algorithm (Brandes 2001), which takes time $O(nm)$ on unweighted graphs and $O(nm + n^2 \log n)$ on weighted graphs. Some randomized algorithms for estimating betweenness centrality have been proposed in the literature (Bader et al. 2007, Brandes and Pich 2007, Jacob et al. 2005), but the accuracy of these randomized algorithms decreases and the execution time increases considerably with the increase in the network size. Variants of betweenness centrality, such as flow betweenness (Freeman et al. 1991) and random-walk betweenness (Newman 2005), take computation time at least of the order nm. Thus, existing approaches for exactly computing or even estimating node betweenness centrality are infeasible for networks with millions of nodes and edges.

We introduce a new approach for identifying highly influential nodes based on their betweenness centrality score, according to the following observations. First, we observe that the exact value of the betweenness centrality is irrelevant for many applications: it is the relative "importance" of nodes (as measured by betweenness centrality) that matters. Second, we observe that for the vast majority of applications, it is sufficient to identify categories of nodes of similar importance: thus, identifying the top 1 % most important nodes is significantly more relevant than precisely ordering the nodes based on their relative betweenness centrality. Third, we observe that distant nodes in (social) networks are unlikely to influence each other (Borgatti and Everett 2006, Friedkin 1983). Finally, we use the observation that influence may not be restricted to shortest paths (Stephenson and Zelen 1989). Capturing these observations, we introduce a new distance-based centrality index called *κ-path centrality*, present a randomized algorithm for estimating it, provide a complexity and accuracy analysis of this algorithm, and show empirically that nodes with high κ-path centrality have high betweenness centrality.

The contributions of this paper are as follows. First, we introduce a new node centrality measure, κ-path centrality, which is intuitively more appropriate for very large social networks because it limits graph exploration to a useful neighborhood of κ social hops around each node. The supporting intuition is twofold: first, in social networks, distant nodes are unlikely to influence each other, and thus the (long) shortest path that connects them is irrelevant in practice. Second, shortest paths are not always the choice for information transmission, as information may travel on less optimal paths.

Second, we introduce and evaluate a randomized algorithm that estimates the κ-path centrality index for all nodes in a network of size n, up to an additive error of at most $n^{1/2+\alpha}$ with probability at least $1 - 1/n^2$ in time $O(\kappa^3 n^{2-2\alpha} \log n)$, where $\alpha \in [-1/2, 1/2]$ controls the trade-off between accuracy and computation time.

Third, we demonstrate empirically on a set of real and synthetic social networks that nodes with high κ-path centrality have high betweenness centrality. Moreover, we show that the running time of our randomized algorithm for estimating κ-path centrality is orders of magnitude lower than the runtime of the best known algorithms for computing exact or approximate betweenness centrality, while maintaining higher accuracy, especially in very large networks. This paper extends our previous work presented in Alahakoon et al. (2011) by comparing the κ-path measure with other betweenness variants found in the literature, by providing a complexity analysis of the proposed randomized algorithm and by including a more thorough empirical evaluation of the algorithm on eight new real networks.

In the remaining part of the paper, we briefly overview the main results in computing betweenness centrality in Sect. 2. We introduce the κ-path centrality index and present and analyze the complexity of the randomized algorithm for computing it in Sect. 3. Section 4 presents our experimental results, comparison with Brandes' algorithm, and two randomized algorithms for estimating betweenness centrality. We conclude in Sect. 5.

2 Node betweenness centrality

Node betweenness centrality is a global centrality index that quantifies how much a vertex controls the information flow between all pairs of vertices in a graph. In this section, we review the formal definition of node betweenness centrality and briefly overview algorithms used in the experimental evaluation that compute exact and approximate betweenness of all vertices in a graph.

2.1 Definition and notations

Let $G = (V, E)$ be any (directed or undirected) graph, described by the set of vertices V and set of edges E. The number of vertices (edges) in G is denoted by n (respectively, m). Let W be a non-negative weight function on the edges of G, where we assume without loss of generality that each edge e of G has $W(e) = 1$ if G is unweighted. We define the *length* of any path ρ in G as the sum of weights of edges in ρ. A *shortest path* from s to t in G is a path of minimum length, and we denote this length by $d_G(s, t)$. Let $P_s(t)$ denote the *set of predecessors* of a vertex t on shortest paths from s to t in G. Let σ_{st} denote the *number of shortest paths* from s to t in G and, for any $v \in V$, let $\sigma_{st}(v)$ denote the number of shortest paths from s to t in G that go through v. Note that $d_G(s, s) = 0$, $\sigma_{ss} = 1$, and $\sigma_{st}(v) = 0$

if $v \in \{s,t\}$ or if v does not lie on any shortest path from s to t.

The *betweenness centrality* index of a vertex v is the summation over all pairs of end vertices of the fractional count of shortest paths going through v.

Definition 1 (*Betweenness centrality* (Anthonisse 1971, Freeman 1977)) For every vertex $v \in V$ of a weighted graph $G(V, E)$, the betweenness centrality $\mathcal{C}_B(v)$ of v is defined by

$$\mathcal{C}_B(v) = \sum_{s \neq v} \sum_{t \neq v,s} \frac{\sigma_{st}(v)}{\sigma_{st}}. \quad (1)$$

2.2 Brandes' algorithm

Brandes' algorithm (Brande 2001) for computing betweenness centrality defines the notion of the *dependency score* of any source vertex s on another vertex v as $\delta_{s\star}(v) = \sum_{t \neq s,v} \frac{\sigma_{st}(v)}{\sigma_{st}}$. Notice that the betweenness centrality $\mathcal{C}_B(v)$ of any vertex v can be expressed in terms of dependency scores as $\mathcal{C}_B(v) = \sum_{s \neq v} \delta_{s\star}(v)$. The following recurrence relation on $\delta_{s\star}(v)$ is significant to Brandes' algorithm:

$$\delta_{s\star}(v) = \sum_{u: v \in P_s(u)} \frac{\sigma_{sv}}{\sigma_{su}} (1 + \delta_{s\star}(u)). \quad (2)$$

The algorithm takes as input a graph $G = (V, E)$ and an array W of edge weights and outputs the betweenness centrality $\mathcal{C}_B[v]$ of every $v \in V$. The running time of Brandes' algorithm on weighted graphs is $\mathcal{O}(nm + n^2 \log n)$ if the min-priority queue Q is implemented by a *Fibonacci heap*. Using BFS instead of Dijkstra's algorithm when the input graph is unweighted, the running time of Brandes' algorithm reduces to $\mathcal{O}(nm)$. The space complexity of Brandes' algorithm on both weighted and unweighted graphs is $O(m + n)$.

2.3 RA-Brandes algorithm

Adapting the technique of Eppstein and Wang (2004) for estimating the closeness centrality, Jacob et al. (2005) and, independently, Brandes and Pich (2007) proposed a randomized approximation algorithm for estimating the betweenness centrality of all vertices in any given graph. This algorithm, which we refer to as *Randomized-Approximate Brandes* or in short *RA-Brandes*, is different from Brandes' algorithm in only one main respect: Brandes' algorithm considers dependency scores $\delta_{s\star}(\cdot)$ of all n start vertices (also called pivots) s, whereas RA-Brandes considers dependency scores of only a multiset $\mathcal{S}$ of $\Theta((\log n)/\epsilon^2)$ pivots. The multiset $\mathcal{S}$ of pivots is selected by choosing vertices uniformly at random with replacement. The estimated betweenness centrality $\widehat{\mathcal{C}}_B[v]$ of any vertex v is then defined as the scaled average of these scores:

$$\widehat{\mathcal{C}}_B[v] = \frac{n}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \delta_{s\star}(v). \quad (3)$$

The running time of RA-Brandes on unweighted graphs is $O(\frac{\log n}{\epsilon^2}(m+n))$, and on weighted graphs is $O(\frac{\log n}{\epsilon^2}(m + n \log n))$ if the min-priority queue Q is implemented by a Fibonacci heap. Its space usage on both weighted and unweighted graphs is $O(m + n)$. The algorithm guarantees computing, for each vertex v, an approximation $\widehat{\mathcal{C}}_B[v]$ that is within $\mathcal{C}_B[v] \pm \epsilon n(n-1)$ with high probability $1 - 1/n^{\Omega(1)}$.

2.4 AS-Brandes algorithm

Bader et al. (2007) proposed a randomized algorithm for estimating the betweenness centrality of all vertices in any given graph. Their algorithm is based on the *adaptive sampling* technique of Lipton and Naughton (1989) used in an algorithm for estimating the size of the transitive closure of a directed graph. The adaptive sampling technique requires selecting a multiset of start vertices by sampling vertices adaptively in the sense that the number of vertices chosen varies with the information gained from each sample. To precisely bound the running time, this algorithm terminates when the number of samples reaches a predetermined cutoff T supplied to the algorithm. Because of its similarity to Brandes' algorithm and application of adaptive sampling technique, we refer to this algorithm as *Adaptive-Sampling Brandes* or in short *AS-Brandes*.

The algorithm AS-Brandes considers dependency scores of only a multiset $\mathcal{S}$ of at most T pivots. It estimates betweenness centrality of any vertex v by noting how fast the sum of dependency scores for v reach a threshold cn, where $c \geq 2$ is supplied to the algorithm. To this end, for each vertex v, the algorithm maintains a running sum $RS[v]$ of dependency scores $\delta_{s\star}(v)$ for pivots s and it records in a variable $k[v]$, the number of pivots used for v until $RS[v]$ becomes greater than cn; $k[v]$ is set to T if $RS[v]$ never exceeds cn. The estimated betweenness centrality $\widehat{\mathcal{C}}_B[v]$ of any vertex v is then defined as the scaled average of these scores over $k[v]$ samples:

$$\widehat{\mathcal{C}}_B[v] = n \frac{RS[v]}{k[v]}. \quad (4)$$

Since AS-Brandes considers only T pivots while Brandes' algorithm considers all n pivots, AS-Brandes should be roughly $\Omega(n/T)$ times faster than Brandes' algorithm. The space usage of AS-Brandes on both weighted and unweighted graphs is $O(m + n)$. The algorithm guarantees

that, for $0 < \epsilon < 0.5$, if the betweenness centrality $C_B[v]$ of a vertex v is at least n^2/t for some constant $t \geq 1$, then with probability at least $1 - 2\epsilon$, its estimated betweenness centrality $\widehat{C}_B[v]$ is within $(1 \pm 1/\epsilon) \cdot C_B[v]$ using ϵt pivots.

3 κ-path centrality

As introduced in Newman (2005), the random-walk betweenness centrality is based on the traversal of the network with absorbing random walks. Assume the traversal of a message (e.g., news or rumor) originating from some source s over a network and intending to finally reach some destination t in the network along a path, and assume that each node in the network has only its own local view (i.e., has information only of its outgoing neighbors). Thus, when the message is at a current node v, the node v forwards the message based on its local view to one of its outgoing neighbors chosen uniformly at random. The message continues to travel in this manner until it reaches the destination node t, and then stops.

The notion of κ-path centrality is based on a similar assumption regarding the random traversal of a message from a source s. However, we make two further assumptions in order to reduce the computation time without deviating much from the above random walk model. First, we consider message traversals along simple paths only, i.e., paths in which vertices do not repeat. As non-simple paths do not correspond to the intuitive notion of ideal message traversals in a social network, their consideration in the computation of centrality indices is a noisy factor. To discount non-simple paths, we assume that each intermediate node v on a partially traversed path forwards the message to a neighbor chosen randomly, with probability inversely proportional to edge weights, from the current set of unvisited neighbors; the message traversal is assumed to stop if all the outgoing neighbors of the current node v already appear in the path up to v. Although choosing a random neighbor in this manner at each step requires the premise that the message carries the history of the path traversed so far, this premise is needed to express the average contribution of any simple path in the overall information flow and to efficiently simulate such random simple paths. Second, we assume that the message traversals are only along paths of at most κ edges, where κ is a parameter dependent on the network. It has been found in many studies on social networks that message traversals typically take paths containing few edges (Friedkin 1983), and so this seems to be a reasonable assumption in the context of social networks. Based on these assumptions, we define κ-path centrality:

Definition 2 (κ-*path centrality*) For every vertex v of a graph $G = (V, E)$, the κ-path centrality $C_k(v)$ of v is defined as the sum, over all possible source nodes s, of the probability that a message originating from s goes through v, assuming that the message traversals are only along random simple paths of at most κ edges.

3.1 Formal analysis of κ-path centrality

Consider an arbitrary simple path $\rho_{s,\ell}$ with start vertex s and having $\ell \leq \kappa$ edges, where κ is the value of parameter κ in κ-path centrality. Let $s, u_1, u_2, \ldots, u_{\ell-1}, u_\ell$ denote the vertices in the order they appear in $\rho_{s,\ell}$ and $s = u_0$ for convenience. For every $0 \leq i \leq \ell$, let $(s, u_1, \ldots, u_{i-1}, u_i)$ denote $\rho_{s,i}$, the subpath from s to u_i, and let $\Pr[\rho_{s,i}]$ denote the probability that a message originating from s traversed through the path $\rho_{s,i}$. The probability $\Pr[\rho_{s,\ell}]$, as shown below, is equal to the product of individual probabilities associated with the random transitions of the message between successive nodes of $\rho_{s,\ell}$. The exact expression of $\Pr[\rho_{s,\ell}]$ depends on whether the graph is weighted or unweighted; so, we consider these two cases separately.

Consider the case of an unweighted, directed graph in which $\rho_{s,\ell}$ is a simple path from s to u_ℓ. For every $0 \leq i \leq \ell$, let $N(u_i)$ denote the set of outgoing neighbors of u_i. The expression for $\Pr[\rho_{s,\ell}]$ is given by the following recurrence relation:

$$\Pr[\rho_{s,i}] = \begin{cases} \Pr[\rho_{s,i-1}] \times \Pr[\text{edge}(u_{i-1}, u_i) \text{ is chosen given } \rho_{s,i-1}] & \text{if } i \geq 2 \\ \frac{1}{|N(s)|} & \text{if } i = 1 \end{cases} \quad (5)$$

Here, $\Pr[\text{edge } (u_{i-1}, u_i) \text{ is chosen given } \rho_{s,i-1}]$ denotes the conditional probability that the message is forwarded from u_{i-1} to u_i, given that the path traversed up to u_{i-1} is $\rho_{s,i-1}$. This probability is equal to $1/|N(u_{i-1}) - \{s, u_1, u_2, \ldots, u_{i-2}\}|$, since, by our assumption, each node u_i forwards the message to a node chosen uniformly at random from the unvisited neighbors of u_i. The above recurrence relation easily leads to the following solution:

$$\Pr[\rho_{s,\ell}] = \prod_{i=1}^{\ell} \frac{1}{|N(u_{i-1}) - \{s, u_1, u_2, \ldots, u_{i-2}\}|}. \quad (6)$$

Notice from the above expression that the larger the outdegree of a node, the smaller is the probability of the message being forwarded through a specific edge. This observation corresponds to the intuition that if the intermediate nodes of a path have a high outdegree, then it is less likely for a message from the source to take that path in its entirety.

Next, consider the case of a weighted, directed graph in which $\rho_{s,\ell}$ is a simple path from s to u_ℓ. In this case, each edge (u_{i-1}, u_i) in $\rho_{s,\ell}$ has a weight $W(u_{i-1}, u_i)$. Intuitively, the weight of the edge (u_{i-1}, u_i) quantifies how easily any information from u_{i-1} can pass to u_i: the smaller the weight of an edge, the more accessible is the endpoint of the edge. Thus, it is more likely for a message to be forwarded on to a lower weight edge than to be forwarded on to a higher weight edge from any node. This intuition suggests the following analog of Eq. (6) for the case of weighted graphs:

$$\Pr[\rho_{s,\ell}] = \prod_{i=1}^{\ell} \frac{1/W(u_{i-1}, u_i)}{\sum_{v \in N(u_{i-1}) - \{s, u_1, u_2, \ldots, u_{i-2}\}} 1/W(u_{i-1}, v)}. \quad (7)$$

Here, the conditional probability that the message is forwarded from u_{i-1} to u_i, given that the path traversed up to u_{i-1} is $\rho_{s,i-1}$, is given by the expression within the product symbol. In this expression, the numerator $1/W(u_{i-1}, u_i)$ corresponds to the intuition that the probability of the message traversing the edge (u_{i-1}, u_i) is inversely proportional to the weight of this edge and the denominator is only a normalization factor so that the probabilities sum to one.

With the above expression for $\Pr[\rho_{s,\ell}]$, we now formalize the notion of κ-path centrality. For any simple path $\rho_{s,\ell}$ originating from s and any $v \neq s$, let

$$\chi[v \in \rho_s] = \begin{cases} 1 & \text{if } v \text{ lies on } \rho_s, \text{ and} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Then, the probability that the message originating from s goes through any vertex v as per our assumptions is given by

$$\sum_{1 \leq \ell \leq \kappa} \sum_{\rho_{s,\ell}:|\rho_{s,\ell}|=\ell} \chi[v \in \rho_{s,\ell}] \cdot \Pr[\rho_{s,\ell}]. \quad (9)$$

The first summation is over the edge counts ℓ of any simple path and the second summation is over all simple paths $\rho_{s,\ell}$ whose edge count is exactly ℓ. In these summations, the contribution $\Pr[\rho_{s,\ell}]$ of any simple path $\rho_{s,\ell}$ is included if and only if v lies on $\rho_{s,\ell}$, as indicated by the expression $\chi[v \in \rho_{s,\ell}] \cdot \Pr[\rho_{s,\ell}]$. Thus, we get an alternative formulation of κ-path centrality.

Proposition 1 ($\boldsymbol{\kappa}$-path centrality) *For every vertex v of graph $G = (V, E)$, the* κ-path *centrality $C_k(v)$ of v is given by*

$$C_k(v) = \sum_{s \neq v} \sum_{1 \leq \ell \leq k} \sum_{\rho_{s,\ell}:|\rho_{s,\ell}|=\ell} \chi[v \in \rho_{s,\ell}] \cdot \Pr[\rho_{s,\ell}], \quad (10)$$

where $\Pr[\rho_{s,\ell}]$ is described by Eq. (6) if G is unweighted and by Eq. (7) if G is weighted.

3.2 Comparison with variants of betweenness

The notion of κ-path centrality contrasts with other variants of betweenness (e.g., κ-betweenness, random-walk betweenness and flow betweenness) in definitions, assumptions, as well as algorithmic complexity.

3.2.1 κ-betweenness or bounded-distance betweenness

Betweenness centrality considers contributions from all shortest paths irrespective of their length. Borgatti and Everett (2006) suggested the idea of limiting the length of shortest paths in the definition of betweenness centrality, as they argued that long paths were seldom used for propagation of influence in some networks. They defined κ-betweenness centrality as an index in which, for each vertex v, its centrality (similar to the case of betweenness) is the sum of dependency scores $\delta_{s\star}(v)$ of all start vertices s on v, but the dependency scores account for only those shortest paths that are of length at most k (as opposed to the case of betweenness in which contributions from all shortest paths are considered). Later, Brandes (2008) redefined this measure as bounded-distance betweenness centrality. For every vertex $v \in V$ of a graph $G = (V, E)$, the *k-betweenness centrality* (Borgatti 2006) $C_{B(k)}(v)$ of v is defined as $C_{B(k)}(v) = \sum_{s,t \in V: d_G(s,t) \leq k} \frac{\sigma_{st}(v)}{\sigma_{st}}$. The κ-betweenness centrality of all vertices of a graph can be computed using Brandes' algorithm where we stop the underlying single-source shortest path search when a vertex of distance k from the source is reached. In traversing the graph from every (source) vertex to all other vertices, the single-source shortest path search breaks on reaching the first vertex that is at distance at least k from the source. In the worst case, if the shortest path distances from every vertex to all other vertices are no more than k, then the algorithmic complexity will be identical to Brandes' algorithm.

3.2.2 Random-walk betweenness

Introduced by Newman (2005), it assumes that message transmission between any two individuals s and t in a social network follows a random path. It models the path the message takes as an absorbing random walk from s to t. The net flow of this random walk on an edge $\{x, y\}$ is defined as the absolute difference between the probability that the walk goes from x to y and the probability that it goes from y to x. The net flow of the random walk through vertex x is defined as one-half of the sum of the net flows on the edges incident to x. The net flow (along an edge or a vertex) is defined in this way so as to discount the

possibility that a random walk repeats a vertex or an edge multiple times. The random-walk betweenness of a vertex v is the expected net flow of a random walk from source s to destination t through v, where the expectation is over all possible pairs (s, t). The best known algorithm for exactly computing random-walk betweenness of all vertices takes time $O(I(n-1)+mn \log n)$, where $I(n) = O(n^3)$ is the time for computing the inverse of an $n \times n$-matrix (Brandes 2005).

3.2.3 Flow betweenness

Introduced by Freeman et al. (1991), it models any directed network as a flow network where edges represent pipes that can carry up to unit amount of flow. The model assumes a flow to be generated at a source node s, transmitted across edges, and absorbed at a sink node t. The value of the flow is defined as the total amount of flow generated at s, and the amount of flow through any vertex x is the total amount of flow leaving x. This notion requires determining the quantity of the flow through a particular vertex v assuming that the flow transmitting from s to t has the maximum possible value. (In case this quantity is not unique because more than one solutions exist for the st-maximum flow problem, then we seek for the maximum flow through v over all possible solutions.) The flow betweenness of a vertex v is defined as the average of this quantity over all possible source–sink pairs (s, t). The flow betweenness of all vertices can be exactly computed in time $O(m^2 n)$ as reported in (Newman 2005).

3.3 Estimating κ-path centrality with a randomized approximation algorithm

We present a randomized approximation algorithm for estimating the κ-path centrality of all vertices in any graph. The algorithm takes as input a graph $G = (V, E)$, a non-negative weight function W on the edges of G, and parameters $\alpha \in [-1/2, 1/2]$ and integer $\kappa = f(m, n)$, and runs in time $O(\kappa^3 n^{2-2\alpha} \ln n)$. For each vertex v, it outputs an estimate of $\mathcal{C}_\kappa(v)$ up to an additive error of $\pm n^{1/2+\alpha}$ with probability at least $1 - 1/n^2$. We refer to this algorithm as *Randomized approximate κpath* or in short RA-κpath.

The algorithm, shown in Algorithm 1, performs $T = 2\kappa^2 n^{1-2\alpha} \ln n$ iterations (the expression for T comes from the analysis of the algorithm, shown next). In each iteration, a start vertex $s \in V$ and a walk length $\ell \in [1, \kappa]$ are chosen uniformly at random. In every iteration, a random walk consisting of ℓ edges from s is performed, which essentially simulates a message traversal from s in G using the assumption made in Definition 2. The number of times any vertex v is visited over all the random walks is recorded in a variable $count[v]$. The estimated κ-path centrality $\widehat{\mathcal{C}}_\kappa[v]$ of any vertex v is then defined as the scaled average of the times v is visited over T walks: $\widehat{\mathcal{C}}_\kappa[v] = \kappa n \cdot \frac{\text{count}[v]}{T}$.

Theorem 1 *The algorithm* RA-κpath *runs in time* $O(\kappa^3 n^{2-2\alpha} \log n)$ *and outputs, for each vertex* v, *an estimate* $\widehat{\mathcal{C}}_\kappa[v]$ *of* $\mathcal{C}_\kappa[v]$ *up to an additive error of* $\pm n^{1/2+\alpha}$ *with probability* $1 - 1/n^2$.

Proof Fix an arbitrary vertex $v \in V$, real $\alpha \in [-1/2, 1/2]$, and integer $\kappa \geq 1$. We define random variables X_i, for $1 \leq i \leq T$, corresponding to the T iterations as follows

$$X_i = \begin{cases} 1 & \text{if the } i\text{th random simple path goes through } v, \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that when the algorithm terminates, $\text{count}[v] = \sum_{i=1}^{T} X_i$. Let us now evaluate the expected value $E[X_i]$ of X_i, for any $1 \leq i \leq T$. Since X_i is an indicator random variable, we have $E[X_i] = \Pr[X_i = 1]$, and, by the definition of X_i, $\Pr[X_i = 1]$ equals the probability that the i'th random simple path goes through v. The algorithm chooses a random start vertex s and a random edge count $\ell \in [1, \kappa]$, where both are distributed uniformly over their respective sample sets. Thus, for any vertex s and edge count $\ell \in [1, \kappa]$, s is chosen as a start vertex and ℓ is chosen as a edge count with probability $1/\kappa n$. Once s and ℓ are fixed, then a path $\rho_{s,\ell}$ of ℓ edge counts originating from s is traversed with probability $\Pr[\rho_{s,\ell}]$, described by Eq. (6) if G is unweighted and by Eq. (7) if G is weighted. It follows that

$$\begin{aligned} E[X_i] &= \frac{1}{\kappa n} \sum_{s \neq v} \sum_{1 \leq \ell \leq \kappa} \sum_{\rho_{s,\ell} : |\rho_{s,\ell}| = \ell} \chi[v \in \rho_{s,\ell}] \cdot \Pr[\rho_{s,\ell}], \\ &= \frac{1}{\kappa n} \mathcal{C}_\kappa[v] \quad \text{(by Proposition 1).} \end{aligned} \tag{11}$$

Let us define random variables Y_i, for $1 \leq i \leq T$, as $Y_i = \kappa\, n X_i$. Note that Y_is are independent random variables and each Y_i takes value of either 0 or $\kappa\, n$. Also, note that the estimate of $\mathcal{C}_\kappa[v]$ returned by RA-κpath algorithm is $\widehat{\mathcal{C}}_\kappa[v] = \kappa n \frac{\text{count}[v]}{T} = \frac{\sum_{i=1}^{T} Y_i}{T}$. Thus, by linearity of expectation,

$$\begin{aligned} E\left[\frac{\sum_{i=1}^{T} Y_i}{T}\right] &= \frac{\kappa n}{T} E\left[\sum_{i=1}^{T} X_i\right] \\ &= \kappa n \cdot E[X_i] \\ &= \mathcal{C}_\kappa(v) \quad \text{(by Eq. 11).} \end{aligned}$$

```
Input: Graph G = (V, E), Array W of edge weights,
       α ∈ [−1/2, 1/2] and integer κ
Output: Array Ĉ_κ of κ-path centrality estimates
begin
    foreach v ∈ V do
        count[v] ← 0; Explored[v] ← false;
    end
    /* S is a stack and n = |V| */
    T ← 2κ²n^{1−2α} ln n; S ← ∅;
    for i ← 1 to T do
        /* simulate a message traversal from s over ℓ edges */
        s ← a vertex chosen uniformly at random from V;
        ℓ ← an integer chosen uniformly at random from [1, κ];
        Explored[s] ← true; push s to S; j ← 1;
        while (j ≤ ℓ and ∃(s, u) ∈ E s.t. !Explored[u]) do
            v ← a vertex chosen randomly from {u | (s, u) ∈ E
                and !Explored[u]} with probability
                proportional to 1/W(s, v);
            Explored[v] ← true; push v to S;
            count[v] ← count[v] + 1;
            s ← v; j ← j + 1;
        end
        /* reinitialize Explored[v] to false */
        while S is nonempty do
            pop v ← S; Explored[v] ← false;
            /* if message traversal stops in less than ℓ edges,
            reset count values to the old ones */
            if (j ≤ ℓ) count[v] ← count[v] − 1
        end
    end
    foreach v ∈ V do
        Ĉ_κ[v] ← κn · count[v]/T;
    end
    return Ĉ_κ;
end
```

Algorithm 1 Randomized approximation algorithm for estimating the κ-path centrality

Application of Hoeffding bound[1] gives

$$\Pr\left[\left|\frac{\sum_{i=1}^{T} Y_i}{T} - \mathcal{C}_\kappa(v)\right| \geq \xi\right] \leq 2e^{-2T^2\xi^2/(T\kappa^2 n^2)} = 2e^{-2T\xi^2/(\kappa^2 n^2)}.$$

Keeping the error margin ξ to $n^{1/2+\alpha}$ results in

$$\Pr[|\widehat{\mathcal{C}}_\kappa[v] - \mathcal{C}_\kappa(v)| \geq \xi] \leq 2e^{-2T/(\kappa^2 n^{1-2\alpha})}. \quad (12)$$

This probability can be made at most $1/n^3$ if $T \geq 2\kappa^2 n^{1-2\alpha} \ln n$. Thus, setting T to $2\kappa^2 n^{1-2\alpha} \ln n$ yields, for every vertex v, an estimate $\widehat{\mathcal{C}}_\kappa[v]$ of $\mathcal{C}_\kappa[v]$ up to an additive error of $\pm n^{1/2+\alpha}$ with probability at least $1 - 1/n^2$. In each of the T iterations, the time spent is $O(\kappa n)$. Therefore, the running time of the algorithm is $O(T\kappa n) = O(\kappa^3 n^{2-2\alpha} \ln n)$. □

4 Experimental evaluation

In order to assess the performance of the algorithm RA-κpath, we compare in Sect. 4.2 its accuracy and running time with that of Brandes' algorithm and in Sect. 4.3 with that of the two betweenness centrality approximation algorithms (RA-Brandes and AS-Brandes). We performed experiments on both real and synthetic social networks. The real networks selected cover a wide range of application domains and scales (file sharing, citation, co-authorship, collaboration, email communication and social), and are presented in Table 1. In order to test the performance of RA-κpath on social graphs that maintain consistent social properties with increase in their size, we created ten independent sets of networks with varying sizes (1, 10, 50 and 100K nodes) using a synthetic social network generator based on the model in (Sala et al. 2010). All experiments were done on a cluster of identical machines with dual core AMD Opteron processors at 2.2 GHz and 4GB RAM.

4.1 Performance metrics

For evaluating the accuracy of κ-path centrality in estimating the relative importance of a node as per the betweenness centrality index, we chose two accuracy metrics. The first metric, called RA-κpath correlation, is the correlation between the approximate κ-path centrality values computed by RA-κpath and the exact betweenness centrality values computed by Brandes' algorithm, for all users in the graph. We applied the same approach to measure the accuracy of the other two approximation algorithms, RA-Brandes and AS-Brandes. We refer to these metrics as RA-Brandes correlation and AS-Brandes correlation, respectively.

The second accuracy metric captures the ability to identify the *top-N%* high betweenness centrality nodes. For this, we measured the percentage of the overlap between the *top-N%* nodes as returned by a particular approximation algorithm (RA-κpath, RA-Brandes, and AS-Brandes) and the *top-N%* nodes as identified by Brandes' algorithm. We refer to these metrics as top *N%* RA-κpath, top *N%* RA-Brandes, and top *N%* AS-Brandes, respectively.

[1] The Hoeffding bound (Hoeffding 1963), a classical result in probability theory, states the following: Let $X_1, X_2, \ldots, X_T$ be independent random variables, such that each X_i ranges over the real interval $[a_i, b_i]$, and let $\mu = E[\sum_{i=1}^{T} X_i/T]$ denote the expected value of the average of these variables. Then, for every $\xi > 0$, $\Pr[|\frac{\sum_{i=1}^{T} X_i}{T} - \mu| \geq \xi] \leq 2e^{-2T^2\xi^2/\sum_{i=1}^{T}(b_i - a_i)^2}$.

Table 1 Summary information of the real networks used

Real networks	Nodes	Edges	d/u, w/uw	References	Network type
Kazaa	2424	13354	u, w	Iamnitchi et al. (2004)	File sharing
Kazaa (U)	2424	13354	u, uw	Iamnitchi et al. (2004)	File sharing
SciMet	2729	10416	u, uw	Batagelj (2006)	Citation
Kohonen	3772	112731	u, uw	Batagelj (2006)	Citation
Geom	6158	11898	u, w	Batagelj (2006)	Co-authorship
Geom (U)	6158	11898	u, uw	Batagelj (2006)	Co-authorship
CA-AstroPh	18772	396160	u, uw	Leskove (2011)	Collaboration
CA-CondMat	23133	186936	u, uw	Leskove (2011)	Co-authorship
Cit-HepPh	34546	421578	d, uw	Leskove (2011)	Citation
Email-Enron	36692	367662	u, uw	Leskove (2011)	Email communication
Cond-Mat-2005	40421	175693	u, w	Newma (2001)	Co-authorship
Cond-Mat-2005 (U)	40421	175693	u, uw	Newma (2001)	Co-authorship
P2P-Gnutella31	62586	147892	d, uw	Ripeanu et al. (2002)	File sharing
Soc-Epinions1	75879	508837	d, uw	Leskove (2011)	Social
Soc-Slashdot0902	82168	948464	d, uw	Leskove (2011)	Social

d/u Directed/undirected, *w/uw* weighted/unweighted

For evaluating the runtime performance, we determined the ratio of the execution time of each of the three approximation algorithms over our implementation of Brandes' algorithm. We refer to this performance metric as speedup, and thus we compare RA-κpath speedup, RA-Brandes speedup, and AS-Brandes speedup.

4.2 Comparison with Brandes' algorithm

We computed the correlation and speedup of RA-κpath with respect to Brandes' algorithm for the real and synthetic social networks for κ varying from 2 to 20 in increments of 2 and α varying from 0 to 0.5 in increments of 0.1. In Figs. 1, 2, 3, we present the correlation and speedup of the real networks with (i) sizes below 10K nodes (Fig. 1a, b), (ii) sizes between 10 and 50K nodes (Fig. 2a, b) and (iii) sizes between 50 and 100K nodes (Fig. 3a, b). We present in Fig. 4a, b the correlation and speedup of all synthetic networks used with sizes between 1 and 100K nodes. These values are averages of ten executions on the ten independently generated networks for each size (thus, $10 \times 10 = 100$ runs for each network size).

We found that, as α decreases, (1) the correlation of RA-κpath with respect to Brandes' algorithm increases, and (2) the speedup of RA-κpath with respect to Brandes' algorithm decreases. The best correlation results are found for $\alpha = 0$ and $\kappa = 20$. Nevertheless, for these values of α and κ, the runtime speedup of RA-κpath in comparison to Brandes' algorithm suffers the most. Furthermore, the improvement of the correlation of RA-κpath across different values for κ, given a constant value of α, shows that the length of the path allowed to take in RA-κpath is extremely important to achieve better results. The correlation performance follows a similar pattern across all network sizes and types.

In particular, we observed that for small real networks such as the first six networks (<10K nodes), the maximum correlation of RA-κpath with Brandes' algorithm is $\sim$0.75 to $\sim$0.95 and the RA-κpath runtime is in the order of $\sim$30 to $\sim$50 times faster than Brandes' algorithm. For larger real networks, the maximum correlation is somewhat lower ($\sim$0.70 to $\sim$0.90). However, the runtime of RA-κpath is about $\sim 10^2$ to $\sim 10^4$ times faster than Brandes' algorithm. The speedup of the runtime of RA-κpath exhibits further improvements on the synthetic social networks, and especially for the networks of larger size.

Overall, the maximum correlation achieved is in the range of $\sim$0.70 to $\sim$0.95 and the maximum speedup achieved is in the range of $\sim 10^2$ to $\sim 10^6$, depending on the values of α, κ, and the size of the network (real or synthetic). A general observation from these results is that we can achieve a near optimal performance of RA-κpath in both correlation and speedup performance metrics when, for a network of n vertices and m edges, α is set to 0.2 and κ is set to $\ln(n+m)$. We used these values of α and κ in the following experiments that compare the performance of RA-κpath with RA-Brandes and AS-Brandes.

4.3 Comparison of RA-κpath with RA-Brandes and AS-Brandes

Figures 5 and 6 show the correlation and speedup results of the three algorithms (RA-κpath, RA-Brandes, and AS-Brandes) with respect to Brandes' algorithm on real

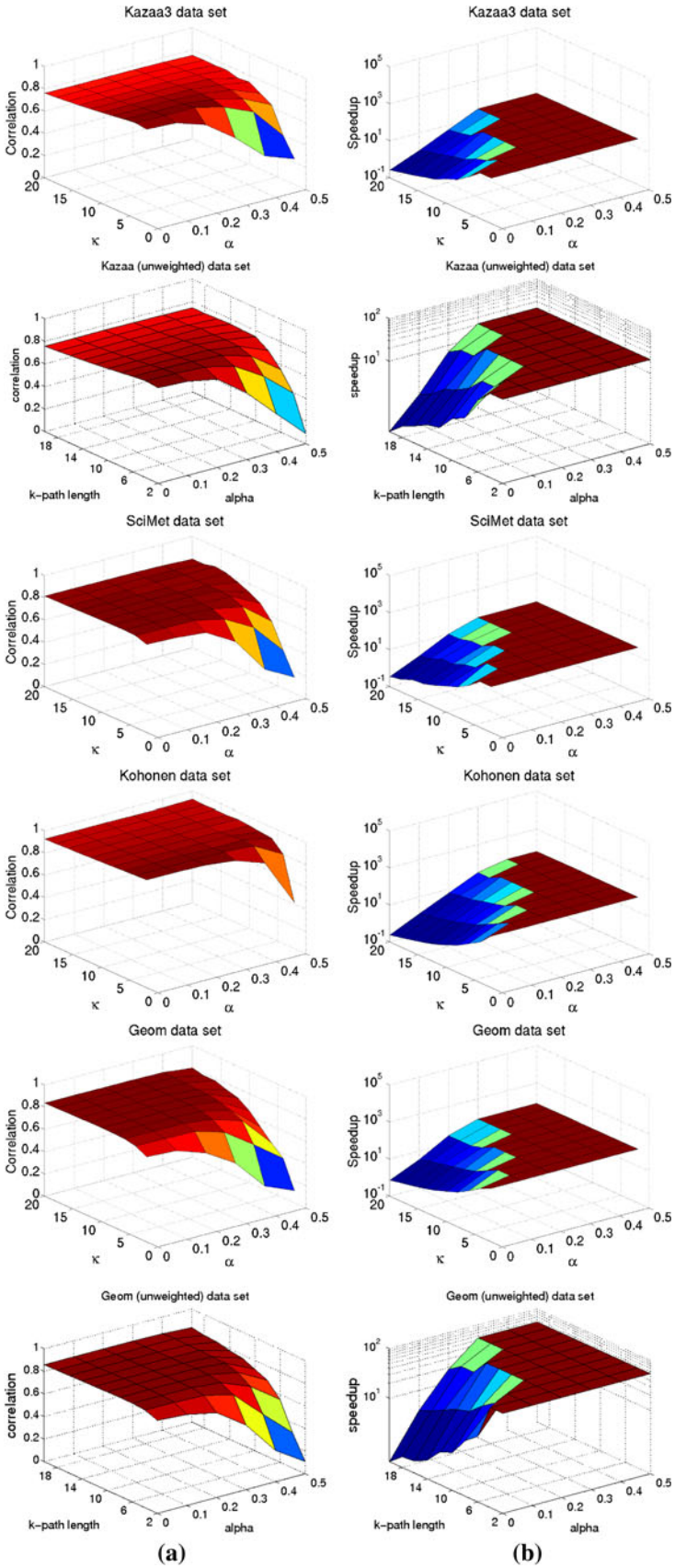

Fig. 1 RA-κpath correlation (**a**) and speedup (**b**) for the real networks with size below 10K nodes

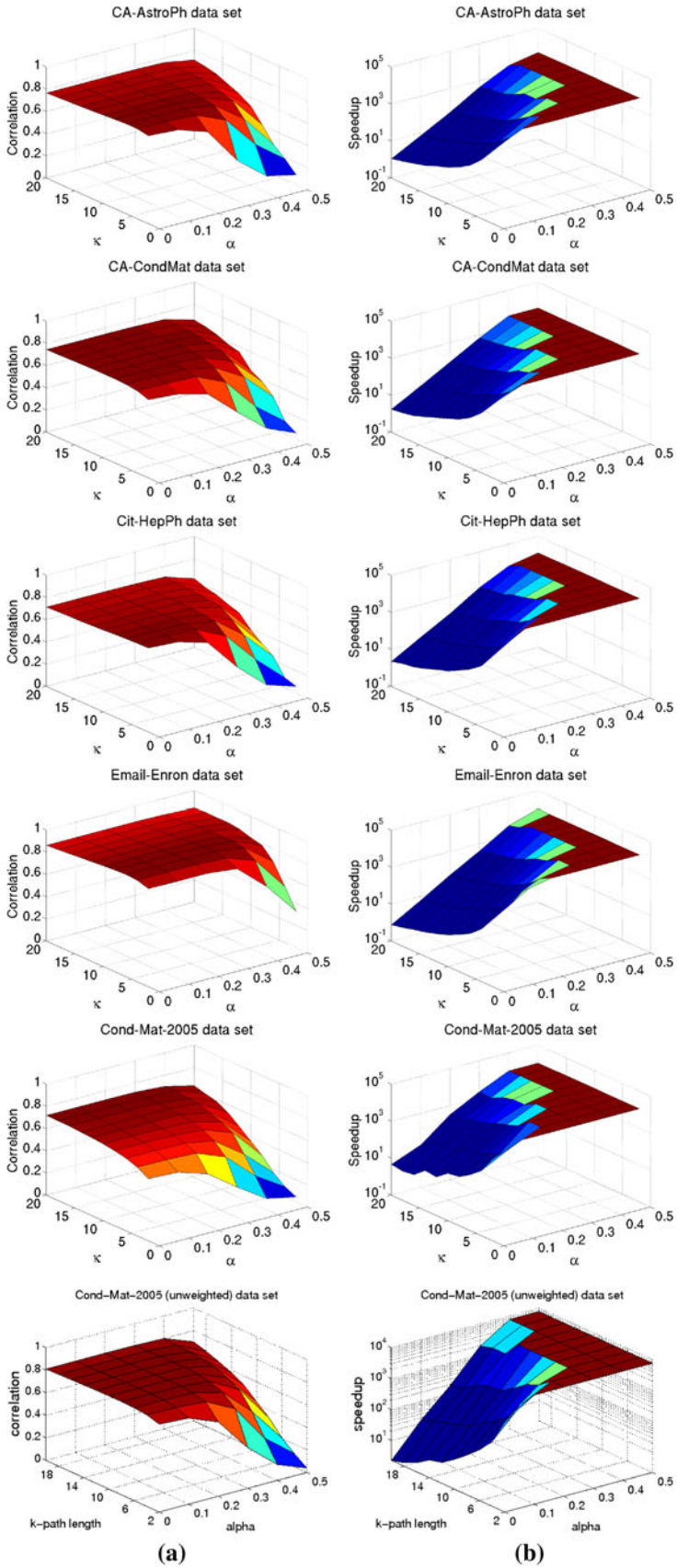

Fig. 2 RA-κpath correlation (**a**) and speedup (**b**) for the real networks with size between 10 and 50K nodes

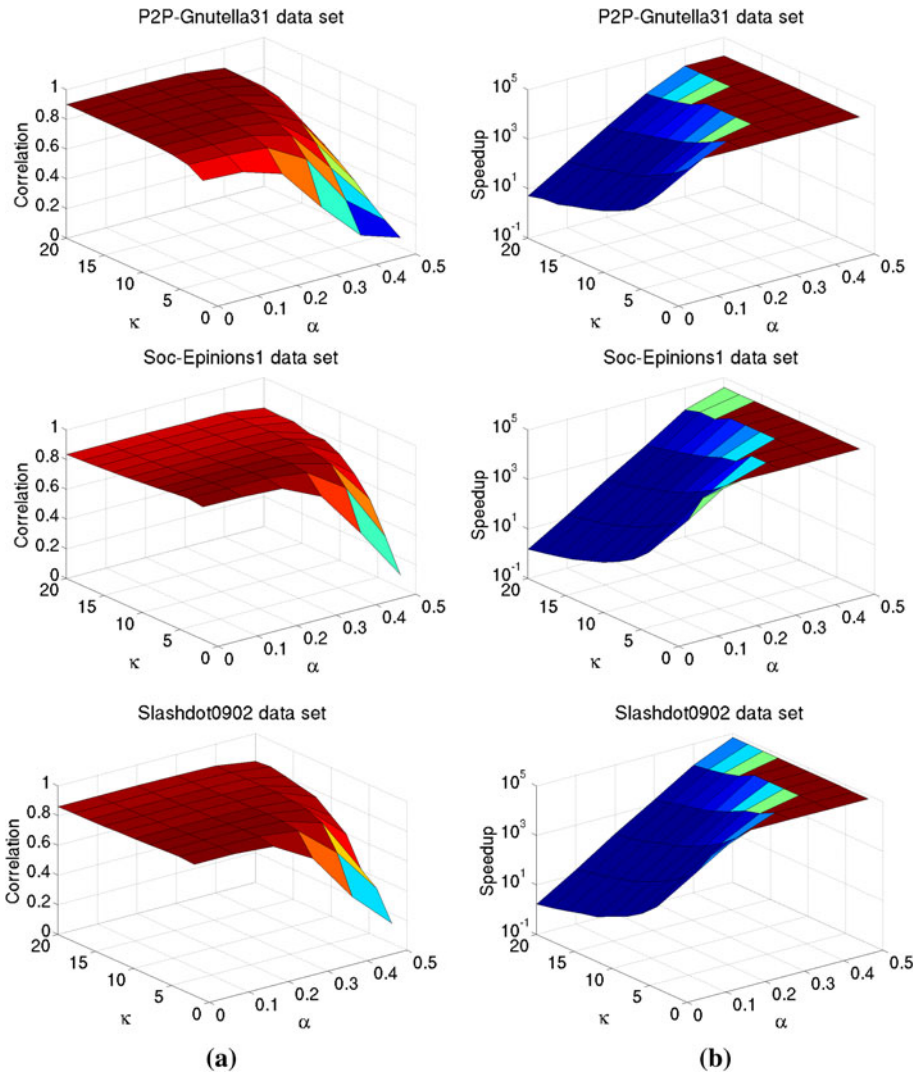

Fig. 3 RA-κpath correlation (**a**) and speedup (**b**) for the real networks with size between 50 and 100K nodes

networks. These results were obtained for $\epsilon = 0.5$ for RA-Brandes, and $s = 20$ and $c = 5$ for AS-Brandes. This choice of parameters for AS-Brandes was also used in Bader et al. (2007). The results demonstrate the superiority of RA-κpath over the other two algorithms in both performance metrics for most of the real networks examined.

However, we believe that the choice of parameter values $\epsilon = 0.5$ and $s = 20$ is not suitable for the sizes of the networks we examined: For example, in Bader et al. (2007) where these values for parameters s and c are used in AS-Brandes, the largest networks evaluated have less than 10K nodes and less than 50K edges. For this reason, we decided to match the speedups of the three algorithms in order to infer less biased parameter values for AS-Brandes and RA-Brandes. We thus performed several experiments with various values of ϵ (for RA-Brandes) and s (for AS-Brandes), and settled on the following heuristic that helped us to closely match the speedups of the three algorithms with respect to Brandes' algorithm:

- $\epsilon = 2 \times ((\text{RA-}\kappa\text{path speedup}) \times \ln(n)/n)^{1/2}$ and
- $s = 2 \times (\text{RA-}\kappa\text{path speedup})$.

The intuition for this choice of ϵ is as follows: RA-Brandes considers dependency scores of $\Theta((\ln n)/\epsilon^2)$ pivots, while Brandes' algorithm considers these scores of all n pivots, and so RA-Brandes speedup can be estimated to $\Theta(n\epsilon^2/\ln n)$; setting this estimate to RA-κpath speedup yields the above expression for ϵ. The intuition for the choice of s follows a similar reasoning. Figures 7 and 8 demonstrate this process for the real and synthetic networks, respectively. For the synthetic social graphs, the values presented are averaged over ten executions on the ten independently generated networks for each size (thus, $10 \times 10 = 100$ runs for each network size).

Figures 9 and 10 show that the correlations of RA-κpath, RA-Brandes, and AS-Brandes vary widely when their speedups are matched. If we compare the results in

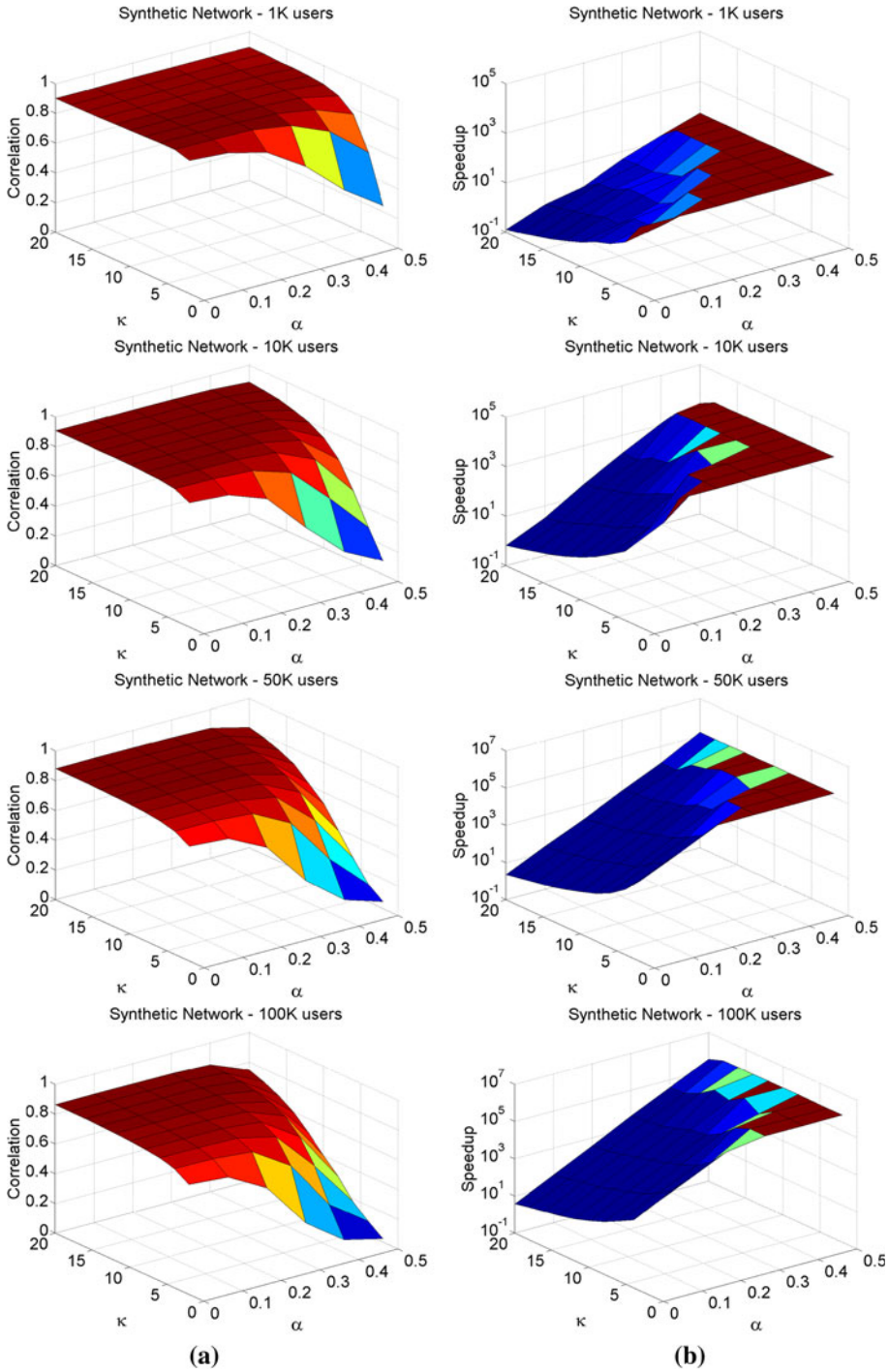

Fig. 4 RA-κpath correlation (**a**) and speedup (**b**) for all the synthetic networks (size between 1 and 100K nodes)

Fig. 9 with the previous correlation performance results shown in Fig. 6, we also notice that the correlations of RA-Brandes and AS-Brandes are enhanced during the speedup-matching process.

Overall, these real networks exhibit a wide range of correlation performance because they acquire different graph properties due to their diverse domains. In most cases (except for the *Kohonen* and *Soc-Epinions*1 networks), RA-κpath outperforms the other two algorithms by a correlation difference of 0.1–0.6, depending on the network type and size. The synthetic networks, on the other hand, are embedded with generic graph properties of real social networks such as power-law degree distribution and high average clustering coefficient. These networks maintain the particular graph properties while increasing the graph size and demonstrate that RA-κpath is consistently better than the other two algorithms on larger networks.

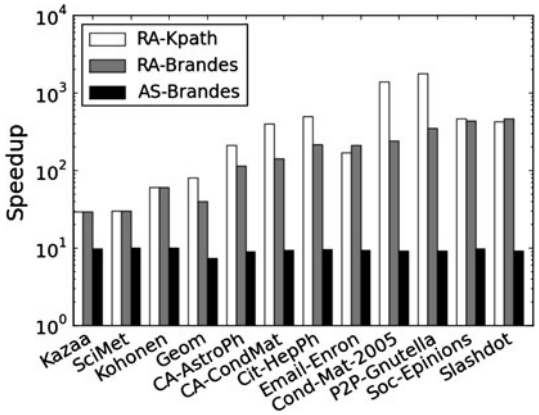

Fig. 5 Speedups of RA-κpath, RA-Brandes, and AS-Brandes with respect to Brandes' algorithm for real networks. The parameters used are $\alpha = 0.2, \kappa = \ln(n+m), \epsilon = 0.5, s = 20$, and $c = 5$

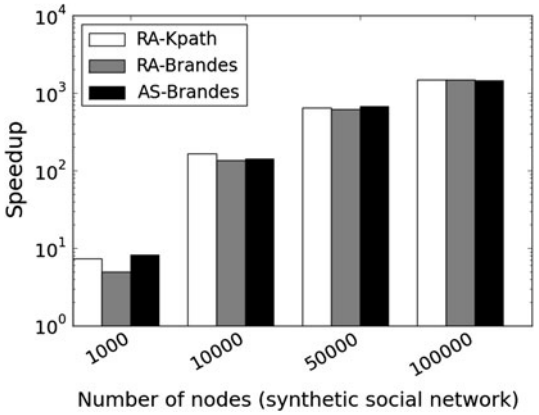

Fig. 8 The speedups of the three algorithms on the synthetic networks were matched to set values of their parameters for the correlation experiments

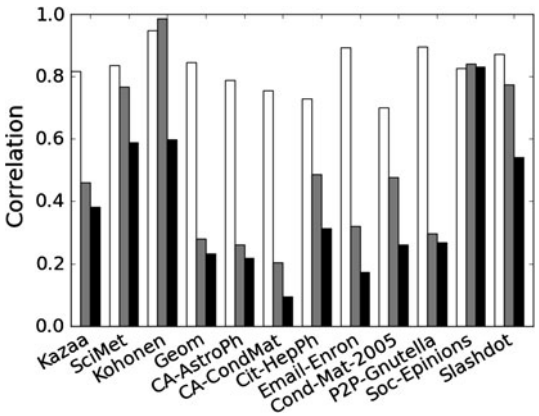

Fig. 6 Correlations of RA-κpath, RA-Brandes, and AS-Brandes with respect to Brandes' algorithm for real networks. The parameters used are $\alpha = 0.2, \kappa = \ln(n+m), \epsilon = 0.5, s = 20$, and $c = 5$

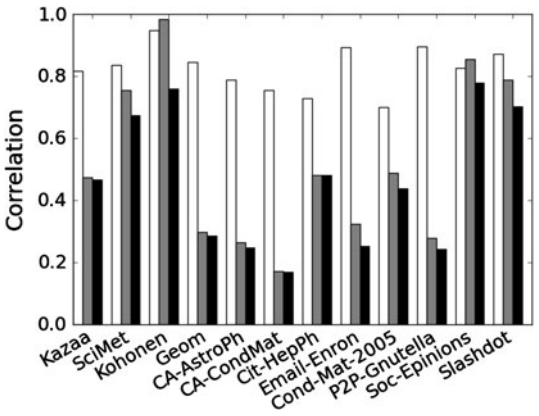

Fig. 9 Correlations of RA-κpath, RA-Brandes, and AS-Brandes with respect to Brandes' algorithm for the real networks. The speedups of the three algorithms were first matched to set values of their parameters and then the algorithms were run with these values to compute their correlation scores

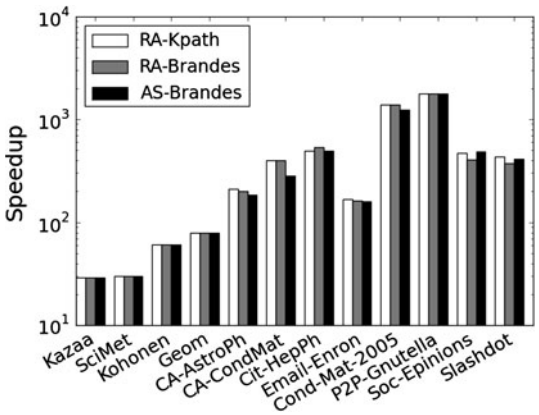

Fig. 7 The speedups of the three algorithms on the real networks were matched to set values of their parameters for the correlation experiments

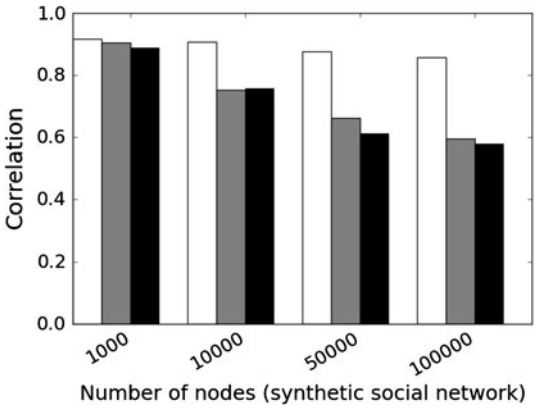

Fig. 10 Correlations of RA-κpath, RA-Brandes, and AS-Brandes with respect to Brandes' algorithm for the synthetic networks. The speedups of the three algorithms were first matched to set values of their parameters and then the algorithms were run with these values to compute their correlation scores

The better performance of RA-κpath shown in these results, even after we matched its speedup with the other algorithms, demonstrates that our proposed algorithm can be used to calculate more accurately relative ranks of betweenness scores for the nodes in a network and could be ideal for experiments on large networks where reliable results are needed fast.

Table 2 shows top N% RA-κpath (RA-K), top N% RA-Brandes (RA-B), and top N% AS-Brandes (AS-B), for the real and synthetic social networks and for $N = 1$, 5, and 10. The results shown were obtained after the algorithms were matched in speedup, as mentioned earlier. Overall, RA-κpath outperforms the other two algorithms by a significant difference of $\sim$10 to $\sim$40 %, in identifying the *top*-1 % high betweenness centrality nodes, in all the sizes and types of networks. This result stresses the effectiveness of RA-κpath in identifying the nodes in a social network which rank the highest in betweenness, and doing so in a fast and efficient way.

When we examine the *top*-5 % and *top*-10 % of nodes, we increase accordingly the subset of nodes considered for the calculation of the high betweenness node overlap. Intuitively, this means that any of these algorithms should perform better, as more nodes are included in the subset, thus increasing the probability of finding more *top* central nodes. This intuition is verified for the RA-Brandes and AS-Brandes algorithms. However, this is not the case for RA-κpath, for which we notice an overall decrease in the performance. This performance deterioration could be due to the arbitrary ordering of low κ-path centrality nodes arising from closeness in their values, allowing them to enter the set of *top* central nodes. In the future, we plan to further examine this ordering and find ways to improve the relative ranking of nodes, thus enhancing the performance of the RA-κpath algorithm.

5 Summary and discussions

In this paper, we introduced a new graph centrality index called κ-path centrality and presented a randomized algorithm RA-κpath for estimating its value for all vertices. Our experimental evaluation demonstrates that this centrality metric can be used to estimate in a scalable way the relative importance of nodes as per the betweenness centrality index: the correlation between the exact and approximate centrality indices is between 0.70 and 0.95 for all network sizes for a speedup gain of up to six orders of magnitude for networks with more than 10K nodes.

Our experiments also show that RA-κpath is very effective and fast in identifying the *top*-1 % or the *top*-5 % nodes in the exact betweenness score, outperforming previously known approximate betweenness centrality algorithms AS-Brandes (Bader et al. 2007) and RA-Brandes (Brandes 2007). The near optimal performance of RA-κpath in both correlation and speedup performance metrics can be achieved when its parameters are set to $\alpha = 0.2$ and

Table 2 Percentage overlap of the *top-N*% nodes computed by the three algorithms with respect to the exact betweenness centrality values

Network	Size (K)	1 %			5 %			10 %		
		RA-K	RA-B	AS-B	RA-K	RA-B	AS-B	RA-K	RA-B	AS-B
Kazaa	2.4	**79.2**	58.3	58.3	**72.7**	64.5	66.9	72.3	79.3	**79.8**
SciMet	2.7	**85.2**	48.1	44.4	**77.9**	66.2	64.0	**76.5**	70.2	69.1
Kohonen	3.7	**75.7**	45.9	64.9	64.4	67.6	**69.1**	60.2	**76.7**	74.0
Geom	6.1	**68.9**	55.7	59.0	71.0	**84.0**	83.4	72.0	**90.4**	89.9
CA-AstroPh	18.7	**63.1**	42.2	39.6	**68.8**	68.1	68.1	74.9	77.8	**78.7**
CA-CondMat	23.1	**74.5**	48.1	48.9	**76.6**	73.2	72.4	76.9	**81.8**	**81.8**
Cit-HepPh	34.5	**71.3**	53.9	47.8	**66.1**	61.2	61.4	66.3	68.9	**69.7**
Email-Enron	36.7	75.1	**79.0**	76.8	63.8	88.5	**89.1**	65.6	**92.7**	**92.7**
Cond-Mat-2005	40.4	66.1	**68.6**	61.4	68.2	**86.5**	85.7	70.4	89.4	**89.5**
P2P-Gnutella31	62.5	**78.2**	31.0	26.6	**78.3**	50.7	50.1	**77.4**	66.2	65.0
Soc-Epinions1	75.9	**80.6**	70.2	71.0	75.0	**90.2**	90.0	72.7	94.8	**95.0**
Soc-Slashdot0902	82.2	**85.9**	67.4	67.7	85.2	**88.8**	88.3	78.4	**92.1**	92.0
Synth-1 K	1	**83.0**	70.0	65.0	**82.4**	70.6	69.6	**77.3**	70.1	69.7
Synth-10 K	10	**88.3**	58.0	58.4	**82.4**	67.8	67.8	**78.7**	78.5	78.5
Synth-50 K	50	**86.6**	61.6	60.8	**81.7**	76.5	77.0	77.5	83.5	**83.8**
Synth-100 K	100	**87.5**	61.0	60.4	**81.4**	79.7	79.8	77.1	84.4	**84.6**

The speedups of the three algorithms were first matched to set their parameters and then executed to compute the *top-N*% overlap

Values in bold denote the highest in the respective (*N*-value) category

$\kappa = \ln(n + m)$, for a network of n number of nodes and m number of edges.

Through our experiments, we have shown that κ-path centrality can be used as an alternative to node betweenness centrality, since (a) κ-path centrality closely models the spread of information in a network and allows to quantify the influence of any node in the network and (b) the speedup performance of RA-κpath for estimating κ-path centrality surpasses those achieved by existing methods of computing exact or approximate betweenness centrality values.

In fact, a parallelized version of our proposed randomized RA-κpath algorithm has been successfully used in a study of the Steam Community (Blackburn et al. 2012), a large-scale gaming social network with over 12 million players and 88.5 million social edges. Our randomized algorithm was used to approximate the betweenness centrality of players and help identify top central players in the gaming social network.

There are various practical applications for identifying the top betweenness centrality nodes in large networks. For example, in unstructured peer-to-peer overlays, the high betweenness peers have a significant impact since they relay much of the traffic (Kourtellis 2011). If under-provisioned, they can damage the overall system performance. If malicious, they can snoop on or divert significant communication. Alternatively, they are great monitoring locations for examining the network communication for traffic characterization studies.

Therefore, identifying the top betweenness centrality nodes can have impact on the network performance (through resource provisioning), security (by restricting the monitoring of potential malicious activity to a small group of candidates), and traffic characterization. Deterministically identifying high betweenness nodes in such a network is infeasible not only because of the large scale (typically hundreds of thousands or millions of nodes), but also because of their dynamic nature caused by high node churn.

Another example of the applicability of our approach is efficient data placement and diffusion. For example, data can be placed on a few high betweenness centrality nodes in a large communication network, such as the Web graph, where informed data placement may lead to faster access to event announcements, or a mobile phone network, where data can be security patches that can be efficiently propagated from a few targeted central individuals to the rest of the population.

Acknowledgments This research was partially supported by the National Science Foundation under Grants No. CNS-0831785 and CNS-0952420. The authors would also like to acknowledge the use of the computing services provided by Research Computing, University of South Florida.

References

Alahakoon T, Tripathi R, Kourtellis N, Simha R, Iamnitchi A (2011) K-path centrality: a new centrality measure in social networks. In: 4th ACM EuroSys workshop on social network systems

Ang CS (2011) Interaction networks and patterns of guild community in massively multiplayer online games. Soc Netw Anal Min 1(4):341–353

Anthonisse J (1971) The rush in a directed graph. Technical Report BN9/71. Stichting Mathematisch Centrum, Amsterdam

Bader D, Kintali S, Madduri K, Mihail M (2007) Approximating betweenness centrality. In: 5th Workshop on algorithms and models for the web-graph, pp 124–137

Batagelj V, Mrvar A (2006) Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/

Blackburn J, Simha R, Kourtellis N, Zuo X, Ripeanu M, Skvoretz J, Iamnitchi A (2012) Branded with a scarlet C: cheaters in a gaming social network. In: 21st International conference on world wide web, Lyon, France

Borgatti S, Everett M (2006) A graph-theoretic perspective on centrality. Soc Netw 28(4):466–484

Brandes U (2001) A faster algorithm for betweenness centrality. J Math Sociol 25(2):163–177

Brandes U (2008) On variants of shortest-path betweenness centrality and their generic computation. Soc Netw 30(2):136–145

Brandes U, Fleischer D (2005) Centrality measures based on current flow. In: Proceedings of the 22nd annual symposium on theoretical aspects of computer science, Lecture notes in computer science, vol 3404. Springer, pp 533–544

Brandes U, Pich C (2007) Centrality estimation in large networks. Int J Bifurc Chaos 17(7):2303–2318 (Special Issue on Complex Networks Structure and Dynamics)

Catanese S, Ferrara E, Fiumara G (2012) Forensic analysis of phone call networks. Soc Netw Anal Min. doi:10.1007/s13278-012-0060-1

Eppstein D, Wang J (2004) Fast approximation of centrality. J Graph Algorithms Appl 8(1):39–45

Freeman L (1977) A set of measures of centrality based on betweenness. Sociometry 40(1):35–41

Freeman C, Borgatti S, White D (1991) Centrality in valued graphs: A measure of betweenness based on network flow. Soc Netw 13(2):141–154

Friedkin N (1983) Horizons of observability and limits of informal control in organizations. Soc Forces, 62(1):57–77

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58:13–30

Hua G, Haughton D (2012) A network analysis of an online expertise sharing community. Soc Netw Anal Min. doi:10.1007/s13278-012-0047-y

Iamnitchi A, Ripeanu M, Foster I (2004) Small-world file-sharing communities. In: 23rd Conf. of the IEEE Communications Society (InfoCom), pp 952–963

Jacob R, Koschützki D, Lehmann K, Peeters L, Podehl D (2005) Algorithms for centrality indices. In Network Analysis, volume 3418 of LNCS, Springer, pp 62–82

Jeong H, Mason S, Barabási A, Oltvai Z (2001) Lethality and centrality in protein networks. Nature 411:41

Kahng G, Oh E, Kahng B, Kim D (2003) Betweenness centrality correlation in social networks. Phys Rev E 67:01710–01711

Kourtellis N, Iamnitchi A (2011) Inferring peer centrality in socially-informed peer-to-peer systems. In: 11th IEEE International conference on Peer-to-Peer computing, Kyoto, Japan

Leskovec J (2011) Stanford large network dataset collection. http://snap.stanford.edu/data/

Liljeros F, Edling C, Amaral L, Stanley H, Aberg Y (2001) The web of human sexual contacts. Nature 411:907

Lipton R, Naughton J (1989) Estimating the size of generalized transitive closures. In: 15th International conference on very large databases. Morgan Kaufmann, San Francisco, pp 165–171

Macskassy S (2011) Contextual linking behavior of bloggers: leveraging text mining to enable topic-based analysis. Soc Netw Anal Min 1(4):355–375

Maglaras LA, Katsaros D (2011) New measures for characterizing the significance of nodes in wireless ad hoc networks via localized path-based neighborhood analysis. Soc Netw Anal Min. doi: 10.1007/s13278-011-0029-5

Newman M (2001) The structure of scientific collaboration networks. Proc Nat Acad Sci USA 98(2):404–409

Newman M (2005) A measure of betweenness centrality based on random walks. Soc Netw 27(1):39–54

Ortiz M, Hoyos J, Lopez M (2004) The social networks of academic performance in a student context of poverty in Mexico. Soc Netw 26(2):175–188

Ripeanu M, Iamnitchi A, Foster I (2002) Mapping the Gnutella network. Internet Comput IEEE 6(1):50–57

Said Y, Wegman E, Sharabati W, Rigsby J (2008) Social networks of author-coauthor relationships. Comput Stat Data Anal 52:2177–2184

Sala A, Cao L, Wilson C, Zablit R, Zheng H, Zhao B (2010) Measurement-calibrated graph models for social network experiments. In: 19th International conference on world wide web, pp 861–870

Singh B, Gupte N (2005) Congestion and decongestion in a communication network. Phys Rev E 71(5):055103

Stephenson K, Zelen M (1989) Rethinking centrality: methods and examples. Soc Netw 11:1–37