The Delta-Betweenness Centrality

Alexander Plutov and Michael Segal, *Senior Member, IEEE*

Communication Systems Engineering Department
Ben-Gurion University of the Negev

***Abstract*—In this paper we consider the extension of the betweenness centrality measure which is used in social and computer communication networks to estimate the potential monitoring and control capabilities a node may have on data flowing in the network. Unlike the standard betweenness centrality measure which takes into account only the shortest paths between the nodes of given network, our, so-called Δ-*betweenness centrality measure* is based on short paths between the nodes. We present an efficient algorithm for computing this measure and show experimental results supporting the importance of newly defined measure.**

I. Introduction

In order to determine the relative importance of every node in a given network (or graph), various centrality measures such as degree, closeness, and betweenness [1, 4, 5] have been proposed. In this paper we extend the standard definition of betweenness centrality measure, defined by Freeman [1] in order to estimate the control an individual may have over communication flows in social networks. For a given undirected, unweighted graph $G=(V,E)$, denote the number of shortest paths from $s \in V$ to $t \in V$ by σ_{st}. Let $\sigma_{st}(v)$ be the number of shortest paths from s to t that some $v \in V$ lies on. The standard (shortest path) betweenness centrality of a node *v* is defined as the sum of fractions of all shortest paths between each pair of vertices in a network that traverse a given node: $\sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$. Brandes [2] and Newman [9] have shown that shortest paths from a single source *s* to all other vertices can be efficiently aggregated by traversing the vertices in the order of a non-increasing distance from *s*. Efficient aggregation of shortest paths is used to compute the measure of all vertices in a network in $O(|V||E|)$ time. Below we show how to extend this definition to deal with not necessarily shortest paths, but rather Δ-shortest paths which were introduced by Jiang et al. [10]. However, their proposed solution in [10] works only for walks and has exponential (in Δ) runtime and, thus, cannot be used for real world application. Moreover, their solution in [10] cannot deal with the weighted version of Δ-shortest paths.

The Δ-shortest paths can exist in some types of real networks. For example, in communication networks, the routing protocols with the purpose of disseminating data traffic build the shortest path, but link fading, node loading, and load balancing problems cause the use of other alternative paths. In the real network the traffic could flow along the Δ-shortest paths with non-uniform distribution for different values of Δ. The traffic flow in the network may use 60% of the shortest paths, 20% of the 1-shortest paths, 15% of the 2- shortest paths and 5% of *k*-shortest paths, $k \geq 3$. In this case, in order to approximate the real flow in the network, we need to take into account this distribution. Another example is coming from the field of well-known reactive routing protocols (DSR, AODV): route discovery and route maintenance problems. After each link failure, the rediscovery process is necessary in order to establish alternate paths. The practical implementation of Δ-shortest paths in this case is that a routing protocol may be a priori oriented not only by using the shortest paths, but also to find alternative Δ-shortest paths. The main goal of this routing protocol is to provide an efficient and low cost routing for MANET. Using Δ-shortest paths may reduce the number of active path failures, implicitly inducing the increasing path lifetime with minimal increase in the control traffic consumption.

II. Preliminaries

Define a *path* $P(s,t)$ from $s \in V$ to $t \in V$ as an alternating sequence of edges $e=(i,j) \in E$ and distinct nodes $i \in V$, beginning with s and ending with t, such that each edge connects it's preceding with its succeeding node. The *length* of a path in unweighted graph is the number of its edges. We use d_{st} to denote the *distance* between nodes s and t, i.e., the minimum length of any path connecting s and t in G. Define a *shortest path* between nodes s and t as any path of length equal to d_{st}. Let $N(v)$ denote the number of neighbors of some $v \in V$. Define the set of predecessors of a node v on shortest paths from s as $P_s(v)=\{j \in V:(v,j) \in E, d_{sv}=d_{sj}+\omega_{jv}\}$, where ω_{jv} is a weight of edge (j,v). Note, that in (our) case of unweighted graph we have $\omega_{jv}=1$ and, thus,

$d_{sv}=d_{sj}+1$. Therefore, combinatorial shortest-path counting can be done following [2].

Lemma 1[2] *For* $s \neq v \in V$, *the number of shortest paths* $\sigma_{sv} = \sum_{j \in P_s(v)} \sigma_{sj}$

Let $\sigma_{st}(v)$ be the number of shortest paths from s to t that some $v \in V$ lies on. The pair-dependency of a pair $s,t \in V$ on intermediate node $v \in V$ defined as $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$, i.e., the ratio of the number of shortest paths between s to t that v lies on [2]. The sum of all pair-dependencies *of node* $s \in V$ *on a single node* $v \in V$ is defined as $\delta_{s*}(v) = \sum_{t \in V} \delta_{st}(v)$. The crucial observation is that partial sums obey recursive relation (Theorem 1 [2]).

Theorem 1 [2] *The pair-dependency of* $s \in V$ *on any* $v \in V$ *obeys* $\delta_{s*}(v) = \sum_{j: v \in P_s(j)} \frac{\sigma_{sv}}{\sigma_{sj}} (1 + \delta_{s*}(j))$.

We denote by $d_{st}^{(\Delta)} = d_{st} + \Delta$ the *strong* Δ*-shortest distance* between nodes s and t in G if there exists a path $P^{(\Delta)}(s,t)$ between s and t of length $l_{st} = d_{st} + \Delta$ for any positive integer Δ. Such path is called a *strong* Δ*-shortest path*. The set of *strong* Δ*-shortest paths* between nodes s and t, $\{P_1^{(\Delta)}(s,t), P_2^{(\Delta)}(s,t) \ldots P_l^{(\Delta)}(s,t)\}$ is denoted by $P^{(\Delta)}(s,t)$. Let $\sigma_{st}^{(\Delta)}$ be the *number of strong* Δ*-shortest paths* from $s \in V$ to $t \in V$; therefore $\sigma_{st}^{(\Delta)} = |P^{(\Delta)}(s,t)|$. Note that in case $\Delta = 0$, the number of strong Δ-shortest paths is equal to the number of shortest paths, i.e. $\sigma_{st}^{(0)} = \sigma_{st}$. We define the set of strong Δ-predecessors of node v on Δ-shortest paths from s $P_s^{(\Delta)}(v) = \{ j \in V : (v,j) \in E, d_{sv} + \Delta = d_{sj}^{(k)} + \omega_{jv}$, *if exists* k *such that* $0 \le k \le \Delta \}$.

Let $\sigma_{st}^{(\Delta)}(v)$ be the number of strong Δ-shortest paths from s to t that some $v \in V$ lies on. We define the *strong* Δ*-pair-dependency* of a pair $s,t \in V$ on intermediate node $v \in V$ as

$$\delta_{st}^{(\Delta)}(v) = \frac{\sigma_{st}^{(\Delta)}(v)}{\sigma_{st}^{(\Delta)}} \qquad (1)$$

i.e. the ratio of the number of strong Δ-shortest paths from s to t that v lies on to the total number of strong Δ-shortest paths from s to t. The Δ*-shortest path* $\hat{P}_i^{(\Delta)}(s,t)$ between nodes s and t is any path of length l_{st}^i not exceeding $d_{st}^{(\Delta)}$, i.e. $l_{st}^i \le d_{st} + \Delta$. The set of Δ-*shortest paths* between nodes s and t $\{\hat{P}_1^{(\Delta)}(s,t), \hat{P}_2^{(\Delta)}(s,t) \ldots \hat{P}_k^{(\Delta)}(s,t)\}$ is denoted by $\hat{P}^{(\Delta)}(s,t)$. Let $\hat{\sigma}_{st}^{(\Delta)}$ be the *number of* Δ*-shortest paths* from $s \in V$ to $t \in V$; therefore, $\hat{\sigma}_{st}^{(\Delta)} = \sum_{n=0}^{\Delta} \sigma_{st}^{(n)} = |\hat{P}^{(\Delta)}(s,t)|$. Let define the set of Δ-predecessors of node v on Δ-shortest paths from s as $\hat{P}_s^{(\Delta)}(v) = \{ j \in V : \{v,j\} \in E, d_{sv} + \Delta \ge d_{sj}^{(k)} + \omega_{jv}, 0 \le k \le \Delta \}$ and the set of Δ-successors of v on Δ-shortest paths from s as $S_s^{(\Delta)}(v) = \{ j \in V : \{v,j\} \in E, d_{sj} + \Delta \ge d_{sv}^{(k)} + \omega_{jv}, 0 \le k \le \Delta \}$ where ω_{jv} is a weight of edge (j,v). Let $\hat{\sigma}_{st}^{(\Delta)}(v)$ denote the *number of* Δ*-shortest paths* from s to t that some $v \in V$ lies on; thus, $\hat{\sigma}_{st}^{(\Delta)}(v) = \sum_{n=0}^{\Delta} \sigma_{st}^{(n)}(v)$. Let define the Δ*-pair-dependency* of a pair $s,t \in V$ on an intermediate $v \in V$ as

$$\hat{\delta}_{st}^{(\Delta)}(v) = \frac{\hat{\sigma}_{st}^{(\Delta)}(v)}{\hat{\sigma}_{st}^{(\Delta)}} = \frac{\sum_{n=0}^{\Delta} \sigma_{st}^{(n)}(v)}{\sum_{n=0}^{\Delta} \sigma_{st}^{(n)}} \qquad (2)$$

i.e. the ratio of the number of Δ-shortest paths from s to t that v lies on to the total number of Δ-shortest paths from s to t. The (Δ, m)*-shortest path from* s *to* t that some $v \in V$ lies on is a Δ-shortest path from s to t, composed of combination of $(\Delta - m)$-shortest path from s to v and $\tilde{m}$-strong shortest path from v to t, where $\tilde{m} = d_{st} - d_{sv} - d_{vt} + m$, for any positive integer m such that $m \le \Delta$. The length of the Δ-shortest path from s to t in this case is bounded by $d_{st}+m \le (l_{sv}+l_{vt}) \le d_{st}+\Delta$ since $l_{st} = l_{sv} + l_{vt}$ and $d_{st} + m \le l_{st} \le d_{st} + \Delta$. In addition, the length of a path from v to t is equal to $l_{vt} = d_{st} - d_{sv} + m = d_{vt} + \tilde{m}$. Let $\hat{\sigma}_{st}^{(\Delta,m)}(v)$ denote the number of (Δ,m)-shortest paths from s to t for some particular $v \in V$ that lies on a Δ-shortest path from s to t. Let us define $\hat{\delta}_{st}^{(\Delta,m)}(v) = \hat{\sigma}_{st}^{(\Delta,m)}(v) / \hat{\sigma}_{st}^{(\Delta)}$ be the (Δ,m)*-pair-dependency score* of a pair $s,t \in V$ on an intermediate node $v \in V$. To eliminate the need for explicit summation of all Δ-pair-dependencies, we introduce the (Δ,m)-dependency score of a node $s \in V$ on a single node $v \in V$, as $\hat{\delta}_{s*}^{(\Delta,m)}(v) = \sum_{t \in V} \hat{\delta}_{st}^{(\Delta,m)}(v)$. Let $\hat{\sigma}_{st}^{(\Delta,m)}(w,v)$ denote the number of (Δ,m)-shortest paths from s to t such that node $v \in P_s^{(\Delta)}(w)$ and edge $(w,v) \in E$ lies on paths from v to t. We extend Δ-pair-dependency score definition to include an edge $e=(w,v) \in E$ by defining $\hat{\delta}_{st}^{(\Delta,m)}(w,v) = \hat{\sigma}_{st}^{(\Delta,m)}(w,v) / \hat{\sigma}_{st}^{(\Delta)}$ as the *edge* (Δ,m)*-pair-dependency score* of a pair $s,t \in V$ when $v \in P_s^{(\Delta)}(w)$ and edge $(w,v) \in E$ lies on paths from v to t, such that Δ-shortest paths from s to t contain combinations of $\tilde{m}$-shortest paths from s to v and m strong shortest paths from v to t. The DBC

measure is defined as:

$$\hat{C}_B^{(\Delta)}(v) = \sum_{s \neq v \neq t \in V} \hat{\delta}_{st}^{(\Delta)}(v) = \sum_{s \neq v \neq t \in V} \frac{\sum_{n=0}^{\Delta} \sigma_{st}^{(n)}(v)}{\sum_{n=0}^{\Delta} \sigma_{st}^{(n)}} \quad (3)$$

where Δ is positive integer constant. Notice that we can add weight to each shortest path, thus, obtaining the weighted version of the problem.

III. Fast 1-Betweenness Centrality Algorithm in Unweighted Undirected Graph

In this section, we describe fast algorithm for computing DBC in an unweighted undirected graph in the case when $\Delta = 1$. We start by performing a Breadth First Search algorithm (BFS) to determine the number of the shortest paths and strong 1-shortest paths to other nodes from source s (distance zero). For this purpose, for all nodes of some current distance from s we execute two calculation loops. In the first one, we sum the number of the shortest paths and strong 1 -shortest paths from all $i \in N(j)$ such that $d_{si} = d_{sj} + 1$. In the second one, in order to calculate strong 1-shortest paths, we sum the number of shortest paths of the same distance neighbors' nodes. In addition, we assign initial values of (1,0) and (1,1)-pair-dependency scores $\hat{\delta}_{si}^{(1,0)}(i) = 1$ and $\hat{\delta}_{si}^{(1,1)}(i) = 0$. In the next step, for each node we compute the contribution (ratio) of node's (1,0) -pair-dependency $\hat{\delta}_{si}^{(1,0)}(i) = 1$, that "spread" to (1,0) and (1,1)-pair-dependency of other nodes, on Δ-shortest paths between nodes i and s. Hence, we begin by performing a "reverse process", i.e. a bottom-up BFS starting at leaves of node s with maximal distance value and executing of two calculation loops. In the first one, for given node i we calculate the ratio of the (1,1) -pair-dependency for all $N(i)$ by splitting (1,0) -pair-dependency values of i and following composition of (1,1) -pair-dependency values obtained by same splitting process from other vertices having the same as i distance from s. In the second loop, we calculate the ratio of the (1,0) and (1,1)-pair-dependency of node i by splitting it and following composition on corresponding (1,0) and (1,1)-pair-dependency, over all neighbors' nodes with smaller distance. We perform these splitting/composition procedures on each "bottom-up" BFS distance till we reach the source. By repeating the process for all possible nodes $s \in G$ and summing the (1,0) and (1,1)-pair-dependency scores, we compute the 1-betweenness centrality for each node. Algorithm requires $O((n+h)n)$ time and $O(n+h)$ space, where $n = |V|$ and $h = |E|$.

A. Algorithm Analysis

We split the analysis according to the two major steps our algorithm consists of: top-down and bottom-up BFS with the corresponding calculations.

Step 1

In this step a number of shortest paths and strong Δ-shortest paths to other nodes is computed. In addition, in this step we assign initial values of a dependency scores $\hat{\delta}_{s*}^{(\Delta,0)}(i), \hat{\delta}_{s*}^{(\Delta,1)}(i)$ for step 2 as explained later.

Property 1. The number of *strong* Δ*-shortest-paths* is $\sigma_{st}^{(\Delta)} = 0$ for $\Delta < 0$.

Property 2. The number of *strong* Δ*-shortest-path* from s to t that some $v \in V$ lies on is $\sigma_{st}^{(\Delta)}(v) = 0$ for $\Delta < 0$.

Lemma 2 (Bellman-like criterion). A node $v \in V$ lies on a Δ-shortest path between nodes $s, t \in V$ if and only if $d_{sv} + d_{vt} - d_{st} \leq \Delta$.

Lemma 3 (Combinatorial strong Δ-shortest-path counting for unweighted graph). The number of strong Δ-shortest-paths

$$\sigma_{sv}^{(\Delta)} = \sum_{j \in P_s^A(v); d_{sv} = d_{sj}+1} (\sigma_{sj}^{(\Delta)} - \sigma_{sj}^{(\Delta)}(v)) + \sum_{l \in P_s^A(v); d_{sv} = d_{sl}} (\sigma_{sl}^{(\Delta-1)} - \sigma_{sl}^{(\Delta-1)}(v)) + \sum_{p \in P_s^A(v); d_{sv} = d_{sp}-1} (\sigma_{sp}^{(\Delta-2)} - \sigma_{sp}^{(\Delta-2)}(v))$$

for $s \neq v \in V$.

Following Lemma 3 above we can calculate the combinatorial number of strong Δ-shortest paths for $\Delta \leq 1$. The number of strong Δ-shortest paths for $\Delta = 0$ is $\sigma_{sv}^{(0)} = \sum_{j \in P_s^{(0)}(v); d_{sv} = d_{sj}+1} (\sigma_{sj}^{(0)} - 0) + \sum_{l \in P_s^{(0)}(v); d_{sv} = d_{sl}} (\sigma_{sl}^{(-1)} - 0) + \sum_{p \in P_s^{(0)}(v); d_{sv} = d_{sp}-1} (\sigma_{sp}^{(-2)} - 0)$

where $\sigma_{sj}^{(0)}(v) = 0$ and $\sigma_{sl}^{(-1)}(v) = \sigma_{sp}^{(-2)}(v) = \sigma_{sl}^{(-1)} = \sigma_{sp}^{(-2)} = 0$ (by Properties 1,2). Hence, the number of strong 0-shortest paths is $\sigma_{sv}^{(0)} = \sum_{j \in P_s^{(0)}(v); d_{sv} = d_{sj}+1} \sigma_{sj}^{(0)}$ as in Lemma 1.

The number of strong Δ-shortest paths for $\Delta = 1$ is $\sigma_{sv}^{(1)} = \sum_{j \in P_s^{(1)}(v); d_{sv} = d_{sj}+1} (\sigma_{sj}^{(1)} - 0) + \sum_{l \in P_s^{(1)}(v); d_{sv} = d_{sl}} (\sigma_{sl}^{(0)} - 0) + \sum_{p \in P_s^{(1)}(v); d_{sv} = d_{sp}-1} (\sigma_{sp}^{(-1)} - 0)$

where in this case $\sigma_{sj}^{(1)}(v) = \sigma_{sl}^{(0)}(v) = 0$ and $\sigma_{sp}^{(-1)}(v) = \sigma_{sp}^{(-1)} = 0$ (Properties 1,2). Hence, the number of strong 1 -shortest paths is

$$\sigma_{sv}^{(1)} = \sum_{j \in P_s^{(1)}(v); d_{sv} = d_{sj}+1} \sigma_{sj}^{(1)} + \sum_{l \in P_s^{(1)}(v); d_{sv} = d_{sl}} \sigma_{sl}^{(0)} \quad (4)$$

The Lemma 1 and equation (4) are used for calculations in Algorithm 1 at Step 1.

Step 2

Property 3. The Δ-pair-dependency $\hat{\delta}_{sv}^{(\Delta)}(v) = \frac{\hat{\sigma}_{sv}(v)}{\hat{\sigma}_{sv}} = 1$, for any target node $v \in V$.

Property 4. The sum of Δ-pair-dependency $\sum_{t \in V} \hat{\delta}_{st}^{(\Delta)}(s) = |V|$ for any source node $s \in V$.

Lemma 4. The $(\Delta,0)$-pair-dependency $\hat{\delta}_{sv}^{(\Delta,0)}(v) = 1$, for any target node $v \in V$.

The result of Lemma 4 is used when we assign initial values of a dependency score of $\hat{\delta}_{sv}^{(\Delta,0)}(v) = 1$, and it is subtracted from values $\hat{C}_B^{(\Delta)}(i)$.

Lemma 5. The sum of all $(\Delta,0)$ pair-dependencies $\hat{\delta}_{s*}^{(\Delta,0)}(v)=1$ for any farthest node ("leaf") $v\in V$.

The claim of Lemma 5 is used in Algorithm 1 Step 2 when we execute the "bottom- up" BFS starting at leaves towards node s, to calculate the contribution (score) of the DBC by splitting/composition of (Δ,m)-pair-dependency scores over all nodes (Claims 1 and 2 below).

Claim 1. The splitting of (Δ,m)-pair-dependency score of node v on (Δ,m)-edge pair-dependency scores of predecessor's nodes $j,l,p\in P_s^{(\Delta)}(v)$ calculated as following:

$$\hat{\delta}_{st}^{(\Delta,m)}(v)=\sum_{j\in P_s^{(\Delta)}(v);d_{sv}=d_{sj}+1}\hat{\delta}_{st}^{(\Delta,m)}(v,j)+\sum_{l\in P_s^{(\Delta)}(v);d_{sv}=d_{sl}}\hat{\delta}_{st}^{(\Delta,(m+1))}(v,l)+\sum_{p\in P_s^{(\Delta)}(v);d_{sv}=d_{sp}-1}\hat{\delta}_{st}^{(\Delta,(m+2))}(v,p).$$

Claim 2. The composition of (Δ,m)-pair-dependency score of node v from edge (Δ,m)-pair-dependency scores of successor's nodes $\{j,l,p\}\in S_s^{(\Delta)}(v)$ (i.e. node v is predecessor of one of the node j,l,p) calculated as following:

$$\hat{\delta}_{st}^{(\Delta,m)}(v)=\sum_{v\in P_s^{(\Delta)}(j);d_{sv}=d_{sj}+1}\hat{\delta}_{st}^{(\Delta,(m-2))}(j,v)+\sum_{v\in P_s^{(\Delta)}(l);d_{sv}=d_{sl}}\hat{\delta}_{st}^{(\Delta,(m-1))}(l,v)+\sum_{v\in P_s^{(\Delta)}(p);d_{sv}=d_{sp}-1}\hat{\delta}_{st}^{(\Delta,m)}(p,v).$$

Define the Δ-dependency of $s\in V$ on a node $v\in V$ as

$$\hat{\delta}_{s*}^{(\Delta)}(v)=\sum_{t\in V}\hat{\delta}_{st}^{(\Delta)}(v)=\sum_{t\in V}\sum_{m=0}^{\Delta}\hat{\delta}_{st}^{(\Delta,m)}(v)=\sum_{m=0}^{\Delta}\sum_{t\in V}\hat{\delta}_{st}^{(\Delta,m)}(v)=\sum_{m=0}^{\Delta}\hat{\delta}_{s*}^{(\Delta,m)}(v).$$

Thus, the Δ-betweenness centrality

$$\hat{C}_B^{(\Delta)}(v)=\sum_{s\neq v\neq t\in V}\hat{\delta}_{st}^{(\Delta)}(v)=\hat{\delta}_{s*}^{(\Delta)}(v)-\hat{\delta}_{sv}^{(\Delta)}(v)=\hat{\delta}_{s*}^{(\Delta)}(v)-1=\sum_{m=0}^{\Delta}\hat{\delta}_{s*}^{(\Delta,m)}(v)-1.$$

Lemma 6. The Δ-dependency score of $s\in V$ on any $v\in P_s^{(\Delta)}(p)$ and $\forall m:0\le m\le\Delta$ obeys

$$\hat{\delta}_{s*}^{(\Delta)}(v)=\begin{cases}\sum_{p:v\in P_s^{(\Delta)}(p);d_{sv}=d_{sp}-1}\left(\frac{\sum_{k=0}^{\Delta-m}(\sigma_{sv}^{(k)}-\sigma_{sv}^{(k)}(p))}{\sum_{k=0}^{\Delta-m}\sigma_{sp}^{(k)}}*\hat{\delta}_{s*}^{(\Delta,m)}(p)\right), & if\quad d_{sv}=d_{sp}-1\\ \sum_{l:v\in P_s^{(\Delta)}(l);d_{sv}=d_{sl}}\left(\frac{\sum_{k=0}^{\Delta-m}(\sigma_{sv}^{(k)}-\sigma_{sv}^{(k)}(l))}{\sum_{k=0}^{\Delta-(m-1)}\sigma_{sl}^{(k)}}*\hat{\delta}_{s*}^{(\Delta,(m-1))}(l)\right), & if\quad d_{sv}=d_{sl}\\ \sum_{j:v\in P_s^{(\Delta)}(j);d_{sv}=d_{sj}+1}\left(\frac{\sum_{k=0}^{\Delta-m}(\sigma_{sv}^{(k)}-\sigma_{sv}^{(k)}(j))}{\sum_{k=0}^{\Delta-(m-2)}\sigma_{sj}^{(k)}}*\hat{\delta}_{s*}^{(\Delta,(m-2))}(j)\right), & if\quad d_{sv}=d_{sj}+1\end{cases}$$

The *strong* Δ*-shortest path* $P_{st}^{(\Delta)}(v)$ from $s\in V$ to $t\in V$ that $v\in V$ lies on exists and is equal to $P_j^{(k)}(s,v)\cup P_l^{(\tilde{k})}(v,t)$, if and only if there is a strong k-shortest path $P_j^{(k)}(s,v)$ and strong $(\tilde{k})$-shortest path $P_l^{(\tilde{k})}(v,t)$, where $\tilde{k}=(d_{st}+\Delta)-(d_{sv}+k)-d_{vt}$ with the strong Δ-shortest path criterion $P_j^{(k)}(s,v)\cap P_l^{(\tilde{k})}(v,t)=\{v\}$ (we call the last condition as path criterion). The length of the strong Δ-shortest path $P_{st}^{(\Delta)}(v)$ is equal to $l_{st}=l_{sv}+l_{vt}$, and therefore $(d_{st}+\Delta)=(d_{sv}+k)+(d_{vt}+\tilde{k})$. The Cartesian product of *strong* k*-shortest path set* $P^{(k)}(s,v)$ and *strong* $(\tilde{k})$*-shortest path set* $P^{(\tilde{k})}(v,t)$ is defined by:

$$P^{(k)}(s,v)\times P^{(\tilde{k})}(v,t)=\{(P_j^{(k)}(s,v)\cup P_l^{(\tilde{k})}(v,t))\mid P_j^{(k)}(s,v)\in P^{(k)}(s,v),\ P_l^{(\tilde{k})}(v,t)\in P^{(\tilde{k})}(v,t)\}.$$

The set of *strong* k*-shortest paths* $P^{(k)}(s,v)$ between nodes s and v may contain k*-shortest paths* that $t\in V$ lies on, that do not satisfy the strong Δ-shortest path criterion (i.e. $P_j^{(k)}(s,v)\cap P_l^{(\tilde{k})}(v,t)\neq\{v\}$). For $P_{st_i}^{(\Delta)}(v)$ calculation, these paths are needed to be excluded from the result set. Let denote by $Q_{st}^{(k,\tilde{k})}(v)$ the number of elements of set $P^{(k)}(s,v)\times P^{(\tilde{k})}(v,t)$ that do not satisfy the path criterion above and need to be excluded from result set. Therefore, $\sigma_{st}^{(\Delta)}(v)$, for any strong Δ-shortest path, is

$$\sigma_{st}^{(\Delta)}(v)=\sum_{k=0}^{\Delta}(|P^{(k)}(s,v)\times P^{(\tilde{k})}(v,t)|-Q_{st}^{(k,\tilde{k})}(v))=\sum_{k=0}^{\Delta}M^{(k)}\quad(1)$$

$$M^{(k)}=\begin{cases}0, & if\ \ (((d_{sv}+k)+(d_{vt}+\tilde{k})\neq(d_{st}+\Delta))\ \ or\ \ (k<0)\ \ or\ \ (\tilde{k}<0))\\ (\sigma_{sv}^{(k)}*\sigma_{vt}^{(\tilde{k})})-Q_{st}^{(k,\tilde{k})}(v)\end{cases}$$

Theorem 2. The Δ-pair-dependency is $\hat{\delta}_{st}^{(\Delta)}(v)=\sum_{m=0}^{\Delta}\hat{\delta}_{st}^{(\Delta,m)}(v)$

Theorem 3: The fast DBC (for $\Delta=1$) algorithm in unweighted undirected graph work correctly and can be computed in $O((n+h)n)$ time and $O(n+h)$ space.

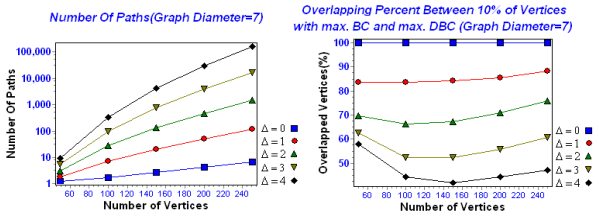
Proof: Follows from Theorem 2, Claim 1,2 and Lemma 2-6.■

IV. Simulations

We conducted extensive simulations to study the performance of Δ-metrics introduced in this paper for graphs with different characteristics. The graph is modeled by a connected unit disk graph with limited graph diameter. We simulate X experiments for each case and the average values for our characteristics have been calculated. The graph Δ-metric analyzed for the five different cases of Δ's values. Figures 1-3 summarize average graph Δ-metric performance for graphs with different number of nodes $|V|=n$. The values of different number of nodes in our experiments randomly placed on square with same neighborhood range are correlated with mean graph degree metric $\tilde{d}=\sum_{i=1}^{n}d_i/n$ and equals 1) $n=50,\tilde{d}=5.5$ 2) $n=100,\tilde{d}=8.1$, 3) $n=150,\tilde{d}=10.5$, 4) $n=200,\tilde{d}=13.2$ 5) $n=250,\tilde{d}=16.1$.

Figure 1(a) plots the average number of strong Δ-shortest paths between all pair of nodes versus total number of

nodes in graph for the five different Δ's values. We can see that the number of strong Δ-shortest paths $\sum_{s \neq v \neq t \in V} \sigma_{st}^{(\Delta)}$ have increased when the number of the nodes in graph is increased. Also, the number of strong Δ-shortest paths for different Δ's values with the same number of nodes is different.



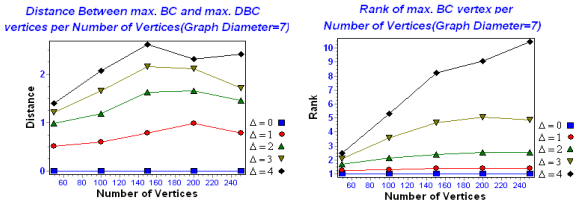
(a) (b)

Figure 1. The plots of the average number of strong Δ-shortest paths between all pairs of nodes versus the number of nodes in graph (a) and the plots of the overlapping percent between subset of 10% nodes with high DBC ranking from ranking list when $\Delta = 0$ and same nodes in ranking lists for $1 \leq \Delta \leq 4$ (b).

Figure 1(b) plots the overlapping percentage between the subset of nodes with high DBC ranking in ranking list when $\Delta = 0$ and same nodes in ranking lists for $1 \leq \Delta \leq 4$ versus number of nodes in graph. The size of the subset in this experiment is 10% of the total number of nodes.

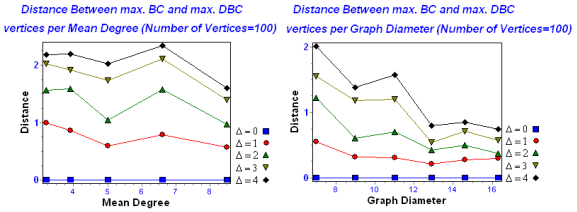
Figure 2 (a) plots the distance between the node with maximal betweenness centrality measure ($\Delta = 0$) and node with maximal $1 \leq \Delta \leq 4$ DBC versus the total number of nodes in same graph for the five different Δ's values. After computing Δ-betweenness centrality measure, each node v in graph G has different DBC value. Therefore we can rank the nodes of the graph by DBC values and find node with maximal DBC value (rank=1) in graph. Note, that for the different Δ values, we have different DBC ranking list. Since, DBC values with different Δ's are used as a relative measure of how nodes differ from each other, the induced ranking is more important than magnitude of the DBC values.

Figure 2 (b) plots the rank of node v with maximal DBC (rank=1) when $\Delta = 0$ and it ranks for $1 \leq \Delta \leq 4$ DBC in same graph versus number of nodes in graph for the five different Δ's values. The node's rankings produced by different DBC are not identical. For example, in the graph with $n = 200$ nodes, the node v with DBC rank=1 in case $\Delta = 0$ has rank=9 when $\Delta = 4$. As we can learn from Figure 2(a) the distance between the node with maximal betweenness centrality measure ($\Delta = 0$) and the node with maximal $1 \leq \Delta \leq 4$ DBC is changed when the number of nodes is increased. We also show the change in the distance between the nodes of centrality measures for the different Δ's values versus the mean degree (Figure 3 (a)) and the diameter of the graph (Figure 3 (b)). Our simulations show that there is a potential importance of the nodes with high Δ-betweenness centrality measure



(a) (b)

Figure 2. The plots of the distance between the node with maximal $\Delta = 0$ BC and the node with maximal $1 \leq \Delta \leq 4$ DBC versus the number of nodes in same graph (a) and the plots the ranks of node v with maximal DBC (rank=1) when $\Delta = 0$ versus the number of nodes (b).



(a) (b)

Figure 3. The plots of the distance between node with maximal $\Delta = 0$ BC and node with maximal $1 \leq \Delta \leq 4$ DBC against mean of graph degree (a) and graph diameter (b) in graph with constant number of nodes $n = 100$.

values and this should be investigated further. It would be interesting to generalize the results for weighted graphs and improve the runtime for the large values of Δ.

REFERENCES

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

[2] Brandes, U.: A faster algorithm for betweenness centrality, *Journal of Mathematical Sociology*, 25 (2), 2001, 163–177.

[3] Brandes, U.: On Variants of Shortest-Path Betweenness Centrality and their Generic Computation, *Social Networks*, 30, (2008), 136-145.

[4] Newman, M. E. J.: A measure of betweenness centrality based on random walks, *Social Networks*, 27, 2005, 39–54.

[5] Borgatti, S. P., and Everett, M. G.: A graph-theoretic perspective on centrality, *Social Networks*, 28, 4 (2006), 466–484.

[6] Krasikov, I, and Noble, S. D.: Finding next-to-shortest paths in a graph, *Information Processing Letters*, 92 (2004), 117–119.

[7] Li, S., Sun, G., and Chen, G.: Improved algorithm for finding next-to-shortest paths, *Information Processing Letters*, 99 (2006), 192–194.

[8] Kao, K.H., Chang, J.M., Wang, Y.L., and Juan, J.S.T.: A Quadratic Algorithm for Finding Next-to-Shortest Paths in Graphs, *Algorithmica*, 61 (2011), 402-418.

[9] Newman, M. E. J.: Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64 (2001), 016132.

[10] Jiang, K., Ediger, D., and Bader, D.: Generalizing k-Betweenness Centrality Using Short Paths and a Parallel Multithreaded Implementation. ICPP (2009), 542-549.