Applied Network Science

RESEARCH Open Access

Neighborhood-based bridge node centrality tuple for complex network analysis

Natarajan Meghanathan*

*Correspondence:
natarajan.meghanathan@jsums.edu
Jackson State University, Jackson, MS 39217, USA

Abstract

We define a bridge node to be a node whose neighbor nodes are sparsely connected to each other and are likely to be part of different components if the node is removed from the network. We propose a computationally light neighborhood-based bridge node centrality (NBNC) tuple that could be used to identify the bridge nodes of a network as well as rank the nodes in a network on the basis of their topological position to function as bridge nodes. The NBNC tuple for a node is asynchronously computed on the basis of the neighborhood graph of the node that comprises of the neighbors of the node as vertices and the links connecting the neighbors as edges. The NBNC tuple for a node has three entries: the number of components in the neighborhood graph of the node, the algebraic connectivity ratio of the neighborhood graph of the node and the number of neighbors of the node. We analyze a suite of 60 complex real-world networks and evaluate the computational lightness, effectiveness, efficiency/accuracy and uniqueness of the NBNC tuple vis-a-vis the existing bridgeness related centrality metrics and the Louvain community detection algorithm.

Keywords: Bridge node, Centrality, Tuple, Complex networks, Clusters, Neighborhood, Algebraic connectivity, Components, Communities

Introduction

We define a bridge node to be a node whose neighbor nodes are sparsely connected to each other and are likely to be part of different components if the node is removed from the network (and are hence likely to be part of different clusters or different communities identified by a clustering algorithm or a community detection algorithm). The above definition for a "bridge" node encompasses three different topological positions for the node: (1) A node whose majority (but not all) of the incident links are with the nodes in its *home* cluster (i.e., in the same cluster as the cluster to which the bridge node belongs to) and the rest of the links (one or few links that are nevertheless critical for connectivity between the clusters) are to bridge nodes in the other *alien* clusters. We refer to a node playing such a role as "*bridge*: *hub*" node. (2) A node that is in the periphery of its home cluster whose majority of the links are with bridge nodes in the other alien clusters (that would not be connected or connected via a much longer path without this node), but has one or few links with nodes that are part of its home cluster. We refer to a node playing such a role as "*bridge*: *border*" node. (3) A node that is not part of any cluster,

Springer Open

but connects two or more clusters or nodes that would otherwise be not connected or sparsely connected. We refer to a node playing such a role as "*bridge*: *between-clusters*" node. The extent to which a node plays the function of a bridge node in one or more of these three topological positions would vary among the nodes in a complex network. To the best of our knowledge, all the existing work in the literature could effectively identify bridge nodes with respect to at most two of the above three topological positions, and not all the three. Hence, there is a need to comprehensively take into account the above three possible topological positions and quantify the extent to which a node plays the role of a bridge node in a complex network.

Bridge nodes are essential for various phenomena in complex networks. Due to their likely proximity to several clusters, bridge nodes are the perfect choice for cluster heads in communication networks. For information cascade to be successful/complete in social networks, the bridge nodes of a cluster need to first adopt a decision before the internal nodes of the cluster (nodes that are not connected to any node outside their home cluster) can adopt the decision. Hence, bridge nodes are ideal candidates to be picked as initial adopters to accomplish complete information cascade (i.e., for a unanimous decision to be made by all the nodes in a network) (Tulu et al. 2018; Berahmand et al. 2019). To prevent an infection spread within a cluster, the bridge nodes of the cluster need to be meticulously identified and vaccinated (Ghalmane et al. 2019a; Masuda 2009) so that the internal nodes could be protected. In collaboration networks, bridge nodes could indicate authors who have collaborated with people who have themselves not collaborated with each other. The identification of such bridge nodes in collaboration networks could lead to new research collaborations among individuals of diverse expertise. Bridge nodes in social networks could lead to acquaintance (could eventually lead to close friendship) between people who had not known each other until then. Likewise, one could enumerate many such phenomena across various complex network domains in which bridge nodes would serve or identify a crucial role. Hence, it is important that we have a quantitative approach to efficiently and effectively identify bridge nodes in complex networks as well as be able to rank all the nodes in a network on the basis of the extent to which they can play the role of bridge nodes.

Centrality metrics quantify the topological importance of nodes in complex networks (Newman 2010). At a broader level (Meghanathan 2016; Oldham et al. 2019), centrality metrics could be neighborhood-based and shortest path-based. Among the neighborhood-based centrality metrics, the degree centrality (DC) and eigenvector centrality (EC) are the most prominent. While the degree centrality (Newman 2010) of a node is simply the number of neighbors of the node, the eigenvector centrality (Bonacich 1987) of a node is a measure of the degree of the node as well as the degree of its neighbors. The DC and EC metrics primarily capture the extent to which a node plays the role of a "hub" node in the network. The shortest path-based centrality metrics (Freeman 1977, 1979) quantify the importance of a node on the basis of either the node's location in the shortest path between any two nodes (betweenness centrality: BC) or the length of the shortest paths to the rest of the nodes in the network (closeness centrality: CC).

The approaches taken so far in the literature to propose centrality metrics to quantify the role of nodes as bridge nodes are primarily of two types: community-aware and community-unaware. Though multiple valid definitions exist in the literature for the term "community" in a network (Peel et al. 2017), we consider a community as a cluster of nodes that are more densely connected among themselves than to the rest of the nodes in the network (Cormack 1971; Hoffman et al. 2018). The underlying idea behind the community-aware approach is to first run a community detection algorithm on the network and then estimate the extent of bridgeness of a node by computing a weighted average of the measures quantifying the topological aspects of a node within its community as well as with other communities. Like in most of the works on community-aware centrality measures, in this paper, we only consider algorithms that are designed to detect non-overlapping communities. The underlying idea behind the community-unaware approach is to first determine the centrality metrics to individually quantify the extent of "hubness" (typically using neighborhood-based metrics such as DC or its variants) and "betweenness" (using shortest path-based metrics such as BC or its variants) and then mathematically combine these two metrics (typically, by taking the product or the ratio of these two metrics) to quantify the extent of "bridgeness" of a node.

The community-aware approach has the weakness of overlooking at the role of a node as "*bridge*: *between-clusters* node" that need not be part of any community (or form a sparsely connected smaller community with a handful of nodes) but connect two or more densely connected communities. In addition, the requirement for the community-aware approach to run a community detection algorithm a priori could become very taxing if the network is very large and we are interested in deciding whether or not just a particular node or a few selected nodes could serve as bridge node(s). On the other hand, with the community-unaware approach, the product or ratio of the metrics representing the hubness and betweenness (measures that quantify two structurally different characteristics) may not be an accurate measure of the extent to which a node functions as a *bridge*: *hub* node vis-a-vis as a *bridge*: *border* node or a *bridge*: *between-clusters* node. As stated earlier, there is a need to comprehensively take into account the extent to which a node serves in the three different topological positions for a bridge node, but in a quest to represent bridge node centrality as a scalar value, we opine that the existing formulations under both the community-aware and community-unaware approaches actually end up overestimating or underestimating the role of a node as a bridge node (see “Related work and motivation” section for more details).

To the best of our knowledge, centrality metrics are until now represented using a scalar value or as a vector of the classical centrality metrics or their variants that is converted to a scalar value using a weighted average formulation. Hence, for the first time in the literature, we propose a centrality tuple, referred to as the neighborhood-based bridge node centrality (NBNC) tuple, whose entries (a sequence of three terms) could be unequivocally used to rank the nodes on the basis of the extent to which they can play the role of a bridge node in a network. Our formulation for the NBNC tuple of a node stems from the definition of a bridge node (stated earlier in this section). In this context, we propose the notion of a "neighborhood graph" of a node that comprises of the neighbors of the node as vertices and the links connecting these neighbors as the edges. The first term of the NBNC tuple for a node is the number of components in its neighborhood graph and the second term is the algebraic connectivity ratio (Fiedler 1973) of the neighborhood graph. In order to break the ties for nodes having the same values for the first and second terms, we include a third term in the NBNC tuple, which is the number of nodes in the neighborhood graph. The NBNC tuple for a top-ranked bridge node is expected to include a larger value for the number of components, a lower value for the algebraic connectivity ratio and a larger value for the number of neighbors. The NBNC tuple can be asynchronously computed (i.e., can be computed just for the node in question without requiring to be simultaneously computed for every node in the network) just based on the two-hop neighborhood knowledge (i.e., local knowledge) of a node and is thus computationally light.

The rest of the paper is organized as follows: "Related work and motivation" section reviews related work in the literature to quantify the role of a node as a bridge node, highlights the weaknesses of the existing approaches and motivates the need for a centrality tuple (vis-a-vis a centrality metric) to rank nodes on the basis of their role as bridge nodes. "Neighborhood-based bridge node centrality (NBNC) tuple" section presents a detailed description of the computation procedure of the proposed neighborhood-based bridge node centrality (NBNC) tuple as well as presents the rules to rank the nodes in a network based on the entries in the NBNC tuple. "Neighborhood-based bridge node centrality (NBNC) tuple" section also demonstrates the computation and ranking procedures for the NBNC tuple using a toy example graph. "Analysis of real-world networks" section first introduces a suite of 60 real-world networks analyzed in this research and then demonstrates the computational lightness, effectiveness, efficiency/accuracy and uniqueness of the proposed NBNC tuple to rank nodes on the basis of the extent to which they function as bridge nodes in a real-world network. "Conclusions and future work" section concludes the paper and outlines plans for future work. All the real-world networks analyzed in this research are considered as undirected graphs.

Throughout the paper, the terms 'node' and 'vertex', 'link' and 'edge', 'network' and 'graph', 'cluster' and 'community' are used interchangeably. They mean the same. A network of nodes and links is referred to as a graph of vertices and edges. Accordingly, we refer to a node in a network as a vertex in a graph and a link in a network as an edge in a graph. Also, note that the neighbor nodes of a bridge node that are in different components in the neighborhood graph of the node could potentially be in different clusters/communities (identified by a clustering/community detection algorithm that is designed to detect non-overlapping clusters/communities).

Related work and motivation

In this section, we review the prominent community-aware and community-unaware approaches in the literature to quantify the role of a node as bridge node in a complex network. We will use the graph shown in Fig. 1 for reference to highlight the weaknesses of the existing approaches as well as motivate the need for a centrality tuple (vis-a-vis a scalar value) to comprehensively capture the extent to which a node plays the role of a bridge node (in the positions of a *bridge*: *hub* node, *bridge*: *border* node, *bridge*: *between-clusters* node). The non-overlapping clusters/communities shown in Fig. 1 are those determined using the well-known Louvain community detection algorithm (Blondel et al. 2008).

Among the ten nodes in Fig. 1: nodes 1, 2, 3, 5, 6, 7 and 8 qualify for the role of a bridge node; however, the extent to which they play the role of a bridge node varies with their

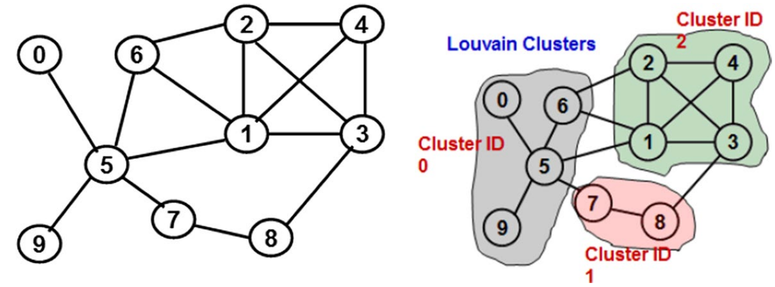


Fig. 1 Motivating example graph and its clusters/communities

topological positions. Node 5 effectively plays the roles of both a *bridge*: *hub* node and a *bridge*: *border* node. Without node 5, two of the three neighbors (i.e., nodes 0 and 9 among nodes {0, 6, 9}) in its home cluster would be disconnected from the rest of the network (in this context: node 5 strongly plays the role of a *bridge*: *hub* node) and the other three neighbors (nodes 1, 6 and 7) would have to communicate on a longer path (in this context: node 5 strongly plays the role of a *bridge*: *border* node). On the other hand, node 1 (that has the same degree as node 5) is weak in its role as a *bridge*: *hub* node and moderate in its role as a *bridge*: *border* node. The nodes in the home cluster of node 1 are not entirely dependent on node 1 to stay connected to each other as well with nodes in the other clusters. Hence, we argue that any centrality metric proposed in the literature to quantify and rank the nodes in Fig. 1 for the role of bridge nodes should rank node 5 higher than node 1. Likewise, nodes 7 and 8 that are part of their own cluster, but connected to the other two clusters, are ideal candidates to be considered as *bridge*: *between-clusters* nodes as well as *bridge*: *border* nodes and be ranked high as bridge nodes. Among nodes 1, 2 and 3, we argue that node 3 be ranked relatively higher because of its topological position to be the only node to directly connect its home cluster {1, 2, 3, 4} with an alien cluster {7, 8}; whereas, neither node 1 nor node 2 are the only nodes to directly connect their home cluster {1, 2, 3, 4} with an alien cluster. The extent to which each of nodes 1, 2 and 3 play the role of a *bridge*: *hub* node or *bridge*: *border* node appear to be about the same; however, the extent to which node 3 plays the role of a *bridge*: *between-clusters* node is relatively more stronger than that of nodes 1 and 2. In "Community-aware approach and Community-unaware approach" sections, we highlight the lapse in the existing work in the literature to differentiate nodes on the basis of the extent to which they play the role of a bridge node in the three topological positions and illustrate how they end up overestimating or underestimating the role of a node as a bridge node.

Community-aware approach

The community-aware approach requires a community detection algorithm to be run a priori and use the identified communities as the basis to quantify the role of a node as a bridge node. Most of the works on the community-aware approach assume the communities to be non-overlapping in nature (i.e., a node can be part of only one community).

Ghalmane et al. (2019a) propose the notion of a community hub-bridge (CHB) measure that appears to take into consideration the role of a bridge node both as a *bridge*: *hub* node and a *bridge*: *border* node, but not as a *bridge*: *between-clusters* node. Per

(Ghalmane et al. 2019a), the CHB measure for a node i in community C_k is the sum of the terms: $|C_k|$ * $\mathrm{DC}_{\mathrm{intra}}(i, C_k)$, quantifying the role of the node as a *bridge*: *hub* node and $\beta_c(i)$ * $\mathrm{DC}_{\mathrm{inter}}(i, C_k)$, quantifying the role of the node as a *bridge*: *border* node; where $|C_k|$, $\mathrm{DC}_{\mathrm{intra}}(i, C_k)$, $\beta_c(i)$ and $\mathrm{DC}_{\mathrm{inter}}(i, C_k)$ respectively represent the number of nodes in community C_k, the number of nodes connected to node i in its home cluster C_k, the number of neighboring communities for node i and the number of nodes connected to node i from other clusters. Table 1 lists the CHB scores for the nodes in the example graph of Fig. 1 with the Louvain clusters considered as the communities of the nodes. We observe *bridge*: *between-clusters* nodes such as nodes 7 and 8 incur CHB scores of 3 each, which is even less than the score for nodes 0, 4 and 9 that are all purely internal nodes within a community and cannot be considered as bridge nodes.

Ghalmane et al. (2019a) also propose that the actual number of disjoint communities (referred with an acronym: NDC in "Analysis of real-world networks" section for comparison purposes) to which a node has incident edges as the basis to rank nodes as bridge nodes: nodes with incident edges to several disjoint communities are considered to more suit the role of a bridge node. However, such a formulation will fail to consider the *bridge*: *between-clusters* nodes (like nodes 7 and 8 in Fig. 1) that are connected to only two or few disjoint communities as well as the *bridge*: *hub* nodes (like node 3 in Fig. 1) that tend to have a larger fraction of its links to nodes within the community, but connected to relatively fewer number of disjoint communities (however, the links to the outside communities are important from betweenness and connectivity point of view).

Gupta et al. (2016) propose a community centrality measure based on simply the intra-degree (number of edges to nodes within the community) and inter-degree (number of edges to nodes in other communities) of the nodes. For two nodes i and j, node i gets ranked higher than node j with respect to the community centrality measure, if node i has both a larger intra-degree as well as a larger inter-degree than node j (or) node i has an intra-degree that is not significantly smaller than that of node j, but has a larger inter-degree. For two nodes i and j with equal intra-degree (or equal inter-degree), node i is ranked higher than node j with respect to the community centrality measure if node i has a larger inter-degree (or larger intra-degree). We claim that the community

Table 1 Computation of the community hub-bridge (CHB) scores (Ghalmane et al. 2019a) for the Nodes in Fig. 1

Node i	Home cluster, C_k	$\lvert C_k \rvert$	$\mathrm{DC}_{\mathrm{intra}}(i, C_k)$	$\beta_c(i)$	$\mathrm{DC}_{\mathrm{inter}}(i, C_k)$	$\lvert C_k \rvert$ *$\mathrm{DC}_{\mathrm{intra}}(i, C_k)$	$\beta_c(i)$ *$\mathrm{DC}_{\mathrm{inter}}(i, C_k)$	CHB(i)
0	0	4	1	0	0	4	0	4
1	2	4	3	1	2	12	2	14
2	2	4	3	1	1	12	1	13
3	2	4	3	1	1	12	1	13
4	2	4	3	0	0	12	0	12
5	0	4	3	2	2	12	4	16
6	0	4	1	1	2	4	2	6
7	1	2	1	1	1	2	1	3
8	1	2	1	1	1	2	1	3
9	0	4	1	0	0	4	0	4

centrality measure cannot be a good measure to rank nodes with respect to the extent of "bridgeness" as the inter-degree of a node need not always correspond to the number of distinct communities a node is connected to, a very important characteristic to consider for a bridge node (with regards to the position as a *bridge*: *border* node). For instance, in Fig. 1, the (intra-degree, inter-degree) of both nodes 1 and 5 are (3, 2). Per the community centrality measure, both nodes 1 and 5 will be equally ranked as bridge nodes; whereas, we argue that node 5 should be ranked higher than node 1 (see the earlier discussion).

Magelinski et al. (2021) recently proposed the notion of "modularity vitality" (referred to as MVIT in “Analysis of real-world networks” section for comparison purposes) to identify nodes whose majority of incident links are inter-community links and exist in the periphery of a community (such nodes are referred to as bridge nodes in Magelinski et al. (2021)) and nodes whose majority of incident links are intra-community links and exist well-inside a community (such nodes are referred to as hub nodes in Magelinski et al. (2021)). In this pursuit, a node whose removal results in an increase in the overall modularity score of the network is considered to play the role of a bridge node and a node whose removal results in a decrease in the overall modularity score of the network is considered to play the role of a hub node. However, we claim that the decrease in the overall modularity score due to the removal of a node cannot be alone used to conclude that a node is "not a bridge node". For example, the removal of node 5 in Fig. 1 would result in the largest decrease in the overall modularity score for the network and per the modularity vitality measure, node 5 could be concluded as a hub node and not a bridge node. However, we claim that node 5 is a bridge node, serving as both a *bridge*: *hub* node as well as a *bridge*: *border* node. Thus, the modularity vitality approach has the weakness of not able to identify bridge nodes that could potentially serve as hub nodes of larger degree within their home cluster.

Ghalmane et al. (2019b) proposed to quantify the centrality of a node with respect to a classical centrality metric (DC, BC, EC or CC) as a vector of two entries that quantify the contribution of a node within its local network as well as in the global network with respect to the particular classical centrality metric. For a given network and the communities identified using a community detection algorithm, the local network is constructed by removing all the inter-community links in the original network and the global network is constructed by removing all the intra-community links in the original network. The intention is that nodes with larger centrality metric values in both the local and global networks could be identified as the bridge nodes. Per this scheme, considering degree centrality (DC) as the classical centrality metric, the vectors for both nodes 1 and 5 in Fig. 1 will be (3, 2). Ghalmane et al. (2019b) also proposed to use the fraction of intra-community links and inter-community links as weights for the centrality metric values computed for a node in respectively the local and global networks to arrive at a comprehensive centrality metric (referred to as *modular centrality* in Ghalmane et al. 2019b) value for a node. With nodes 1 and 5 in Fig. 1 having 3 intra-community links and 2 inter-community links, they will incur identical (scalar) values for the modular centrality with respect to DC. When betweenness centrality (BC) is considered as the classical centrality metric, the modular centrality vectors for nodes 2, 3, 7 and 8 would be (0, 0) each (however, the BC values of these vertices in the original network are each greater than zero), leading to a wrong conclusion that they are not bridge nodes. Note that in addition to being a scalar value, the modular centrality metric value is directly dependent on the underlying classical centrality metric used and the ranking of the nodes as bridge nodes would vary depending on the classical centrality metric considered. An extension of the above work for overlapping communities is available in Ghalmane et al. (2019c), wherein the local network for an overlapping node (that is part of more than one community) comprises of links to nodes in each of its participating communities and the global network for an overlapping node comprises of links to the communities that it is not part of.

Community-unaware approach

The community-unaware approach does not require a community detection algorithm to be run a priori. Instead, a community-unaware approach typically quantifies the extent of "bridgeness" of a node by considering the extents of "hubness" and "betweenness" of the node. Two notable centrality metrics in the literature that fall in the category of community-unaware approach and also include the word "bridge" as part of their name are: bridging centrality (BRC) (Hwang et al. 2008) and bridge node centrality (BNC) (Liu et al. 2019), both of which consider the betweenness centrality (BC) of the node on the (shortest) paths between any two nodes in the network as well as the degree of the node. Both BRC (see Eq. 1) and BNC (see Eq. 2) are computationally heavy, global and synchronous metrics (i.e., require knowledge about the entire network topology in order to be computed for a node as well as cannot be computed for a particular node without being computed for every node in the network). On the other hand, our proposed NBNC tuple is computationally light, locally computable and asynchronous metric (i.e., can be computed for just the node in question by using only the two-hop neighborhood information).

The BRC value (Hwang et al. 2008) for a node is the product of its BC and bridging coefficient (BCO: computed as the ratio of the resistance of the node and the sum of the resistances of its neighbors; see Eq. 1 for the formulations of BCO and BRC). The resistance of a node (i) is the inverse of the degree (k_i) of the node. Table 2 presents the computation of the BRC values of the vertices for the example graph of Fig. 1. Due to the incorporation of betweenness centrality and resistance in its formulation,

Table 2 Computation of the bridging centrality (BRC) values for vertices in the graph of Fig. 1

Node, i	Degree, k_i	$1/k_i$	Neighbors(i)	$\sum_{j \in Neighbors(i)} 1/k_j$	BCO(i)	BC(i)	BRC(i)
0	1	1	5	0.2000	5.0000	0.0000	0.0000
1	5	0.2	2, 3, 4, 5, 6	1.3667	0.1463	9.6667	1.4151
2	4	0.25	1, 3, 4, 6	1.1167	0.2239	1.3333	0.2978
3	4	0.25	1, 2, 4, 8	1.1167	0.2239	4.5000	1.0075
4	3	0.33	1, 2, 3	0.7000	0.4762	0.0000	0.0000
5	5	0.2	0, 1, 6, 7, 9	3.0333	0.0659	18.500	1.2198
6	3	0.33	1, 2, 5	0.6500	0.5128	1.8333	0.9385
7	2	0.5	5, 8	0.7000	0.7143	3.3333	2.3786
8	2	0.5	3, 7	0.7500	0.6667	1.8333	1.2200
9	1	1	5	0.2000	5.0000	0.0000	0.0000

BRC is likely to highly rank nodes with a lower degree, but connected to larger degree neighbors, and a larger betweenness. In this context, we observe the BRC metric to take a note of the "*bridge*: *between-clusters*" nodes 7 and 8 in the example graph and rank them highly as bridge nodes. Nevertheless, we still notice that BRC ranks node 1 relatively higher than node 5 (which exhibits a larger bridgeness compared to node 1), even though node 5 incurs a BC value that is twice of node 1: the reason is that node 1 has relatively more high-degree neighbors than node 5 and hence incurs a larger BCO. The product of BC and BCO results in a scalar value that is an overestimate for node 1 and an underestimate for node 5 with regards to their role as bridge nodes, a typical weakness of the approaches that aim at formulating a scalar centrality metric to quantify the role of a node as bridge node. Instead, we need a tuple that would have two or more terms to capture the extents of hubness and betweenness of the nodes.

$$BCO(i) = \frac{1/k_i}{\sum\limits_{j \in Neighbors(i)} 1/k_j} \qquad BRC(i) = BC(i) * BCO(i) \tag{1}$$

Liu et al. (2019), the authors proposed the notion of Bridge Node Centrality (BNC) to diminish the importance of high-degree nodes and give more importance to nodes with larger route betweeness and closeness to the rest of the nodes in the network. The BNC (Liu et al. 2019) of a node (see Eq. 2) is computed as the product of its route betweenness (RBW) and bridgeness coefficient (BCE) in a normalized scale; the RBW for a node is the ratio of the sum of the weights of the paths (between any two nodes in the network) that go through the node and the degree of the node; the BCE of a node (a measure of closeness of the node) is the inverse of the sum of the lengths (of length > 2) of the shortest paths from the node to the rest of the nodes in the network. The notations used in Eq. (2) represent the following: R is the total number of shortest path routes in the network between any two nodes; $\omega_j = P_j/L_j$, where P_j and L_j are the probability of information flow through route j and L_j is the length of route j; $\delta_j = 1$ if node i is in route j and is 0 otherwise; k_i is the degree of node i; dist(i, j) is the number of hops in the shortest path route between nodes i and j (note that only dist(i, j) > 2 are considered in the computation of the BCE values of a node); $\overline{RBW}(i)$ and $\overline{BCE}(i)$ respectively represent the normalized RBW and BCE values of node i.

$$RBW(i) = \sum\nolimits_{j=1}^{R} \frac{\omega_j \delta_j}{k_i} \quad BCE(i) = 1/\sum\nolimits_{j=1}^{n} dist(i,j) \quad BNC(i) = \overline{RBW}(i) * \overline{BCE}(i) \tag{2}$$

With the BNC metric, nodes with lower degree but a larger RBW (like the *bridge*: *between-clusters* nodes 7 and 8 in Fig. 1) are likely to have a larger BNC. However, nodes with larger degree are likely to suffer from a lower BNC unless they have a very high information flow. For example, a predominantly *bridge*: *hub* node like node 3 in Fig. 1 is likely to incur a lower BNC due to its larger degree and a moderate RBW.

Kleinberg et al. (2008), the authors refer to a bridge node as a structural hole filling the gap between two or more nodes that are otherwise not directly connected and evaluate the various benefits the structural holes bring to a network from a game-theoretic perspective. A recent work (Yang and An 2020) quantifies the importance of a node on the basis of a metric called the *degree and structural hole count* (DSHC) that is computed for a node i per the following formula:

$$DSHC_i = \sum_{j \in \Gamma_i} \left(\left(\frac{1}{k_i} + \frac{1}{k_j} \right) * \frac{1}{1 + \Delta_{ij}} \right)^2, \tag{3}$$

wherein Γ_i, k_i and k_j are respectively the set of neighbors of node i, degree of node i and degree of node j, and Δ_{ij} indicates the number of structural holes that node i fills by connecting node j with the former's other neighbor nodes. Per the above formula, high-degree nodes with high-degree neighbor nodes and a larger value for the number of structural holes with regards to its neighbors are likely to incur a lower DSHC value (highly ranked as a structural hole). While applying the above formula for vertices 1 and 5 in the example graph of Fig. 1, we observe the DSHC scores for vertices 1 and 5 to be 0.1465 and 0.1625 respectively: implying, node 1 is highly ranked as a structural hole compared to node 5, whereas node 5 should be the top-ranked bridge node in this graph. The above formula for DSHC suffers from the same weakness that we had earlier highlighted for the BRC and BNC metrics. The formulation tries to combine node degrees (a measure of the hubness of the node as well as its neighbors) with the number of structural holes and comes up with a single quantitative measure that could overestimate or underestimate the role of a node as the bridge node.

Berahmand et al. (2019), the authors propose a scalar centrality metric to quantify the extent to which a node can serve as an effective spreader of information. Without any formal name in Berahmand et al. (2019), the metric is simply referred to as DCL in Berahmand et al. (2019), probably due to its formulation (see Eq. 4 below) that involves terms based on the degree centrality, inverse of the clustering coefficient and the location information (considered in the form of the degree of the immediate neighbors) of a node. In Eq. (4), k_i and CC_i respectively denote the degree and clustering coefficient (Meghanathan 2017b) of node i, Γ_i denotes the set of neighbors of node i and $|E(\Gamma_i)|$ denotes the average of the degrees of the neighbors of node i. The clustering coefficient (Meghanathan 2017b) of a node is the ratio of the actual number of links connecting the neighbors of the node and the maximum possible number of links between the neighbors of the node. While the first and second terms of Eq. (4) respectively incorporate the degree and inverse of the clustering coefficient of a node, the third term incorporates the location information of a node on the basis of the degrees of the neighbors of the node.

$$DCL_i = \left\{ k_i \right\} * \left\{ \frac{1}{CC_i + (1/k_i)} \right\} + \left\{ \frac{\sum\limits_{j \in \Gamma_i} k_j}{|E(\Gamma_i)| + 1} \right\} \tag{4}$$

The DCL formulation tends to give preference for nodes with larger degree, but a lower clustering coefficient and connected to high degree neighbors. Table 3 illustrates the calculation of the DCL metric values for the nodes in Fig. 1. We observe node 5 to incur the largest DCL score, justifying its roles as a *bridge*: *hub* node and a *bridge*: *border* node. However, the DCL score (the second largest score) of node 1 is greater than the DCL score of node 3 as well as greater than the DCL scores of nodes

Table 3 Computation of the DCL values for vertices in the graph of Fig. 1

Node, i	Degree, k_i	$1/k_i$	CC_i	Γ_i	$\sum_{j\in\Gamma_i} k_j$	$\vert E(\Gamma_i)\vert$	DCL_i
0	1	1	1	5	5	5	1.3333
1	5	0.2	0.5	2, 3, 4, 5, 6	19	4.75	10.4472
2	4	0.25	0.67	1, 3, 4, 6	15	3.75	7.5057
3	4	0.25	0.5	1, 2, 4, 8	14	3.5	8.4444
4	3	0.33	1	1, 2, 3	13	4.33	4.6947
5	5	0.2	0.1	0, 1, 6, 7, 9	12	2.4	20.1961
6	3	0.33	0.67	1, 2, 5	14	4.67	5.4691
7	2	0.5	0	5, 8	7	3.5	5.5556
8	2	0.5	0	3, 7	6	3	5.5000
9	1	1	1	5	5	5	1.3333

7 and 8. Per the DCL metric, if a low degree node with lower clustering coefficient fails to have high degree neighbors (like nodes 7 and 8, one of whose neighbors have a lower degree), it is difficult to get highly ranked as a bridge node. Like the previously discussed metrics, the DCL metric also suffers from the weakness of combining multiple terms as part of its formulation in order to get computed as a scalar value.

Some of the other related approaches (primarily in the context of identifying the most influential nodes for spreading information) that exist in the literature are as follows: Ibnoulouafi et al. (2018), the authors propose to quantify the spreading capability of a node (referred to as M-Centrality) as a weighted average of the K-shell index number of the node (computed as part of K-shell decomposition (Wang et al. 2016)) and the degree variation in the neighborhood of the node; the weight for computing the M-Centrality metric is determined by applying the entropy technique of He et al. (2016) on the K-shell index numbers for the nodes and the extent of variation in node degree compared to the degrees of the neighbor nodes. The K-shell decomposition approach is likely to assign higher index numbers for nodes that are well-connected within a community (like node 4 in Fig. 1) but may not be even connected to nodes in other communities: not a favorable characteristic for identifying bridge nodes. On the other hand, the K-shell index numbers for lower-degree nodes as well as for higher-degree nodes connected to lower-degree nodes are likely to be lower. For example, in Fig. 1, the K-shell index number for critical bridge nodes like nodes 5, 7 and 8 is 2 each, whereas the K-shell index number for nodes 1, 2, 3 and 4 (that are part of a clique: a well-knit community) is 3 each. The variation in node degree compared to the degrees of the neighbor nodes is likely to be larger for nodes (like node 5) with higher degree, but connected to lower degree neighbors and lower for nodes (like nodes 7 and 8) with lower degree, but connected to higher-degree neighbors. Thus, from the perspectives of both the K-shell index numbers and the extent of variation in node degree compared to the neighbor nodes, the M-centrality metric may not be a suitable metric for identifying *bridge*: *between-clusters* nodes. For the problem of automatic keyword extraction, the authors in Vega-Oliveros et al. (2019) first construct a ranking matrix of the vertices (keywords) in a co-occurrence graph of keywords with respect to a suite of nine different centrality metrics (local, global and intermediate-level metrics). They then subject this matrix to principal component analysis and rank the keywords based on the first principal component. Recently

(Bucur 2020), a multi-centrality dataset of the nodes (i.e., containing the centrality values of the nodes with respect to two or more metrics) was used for training a (binary) classifier to predict whether or not a node with certain values for the centrality metrics will be a super spreader.

Overall, we observe the community-unaware bridgeness metrics such as BNC and BRC to give preference for nodes (such as the *bridge*: *between-clusters* nodes) with lower degree and larger information flow/betweenness or metrics such as the DCL give preference for nodes (*bridge*: *border* nodes) with larger degree, but lower clustering coefficient and connected to high-degree neighbors. On the other hand, we observe the community-aware bridgeness metrics to give preference for nodes (*bridge*: *hub* nodes) with larger degree and have one or more inter-community links. A common weakness that we noticed for several metrics under both the approaches is the intention to come up with a scalar score (typically a weighted average of two or more different measures: say, the extent of hubness and betweenness or a weighted average of the entries in a tuple that is an amalgamation of the classical centrality metrics). In this section, we have highlighted several scenarios wherein such a formulation would either underestimate or overestimate the role played by a node as bridge node and cannot comprehensively consider all the three possible topological positions of a bridge node. We need a tuple-based formulation that would take a fundamentally new approach to comprehensively capture the extent to which a node could serve as a bridge node with respect to all the three topological positions.

Neighborhood-based bridge node centrality (NBNC) tuple

We hypothesize that a node could effectively serve the role of a bridge node with respect to all the three topological positions (*bridge*: *hub*, *bridge*: *border* and *bridge*: *between-clusters*) if it has several (more than a few) neighbors, but the connectivity among the neighbors is low. In order to capture such a topological setup and quantify the same, we put forward the notion of a "neighborhood graph of a node" that comprises of the neighbors of the node as vertices and the links connecting these neighbors as the edges. We seek to determine the number of components in the neighborhood graph (NG) as well as the connectivity of the NG spanning these components. The graph-theoretical definition of a component (Cormen et al. 2009) in a graph is: the subset of the vertices and the associated edges such that there is no edge to vertices outside the subset. Note that the neighborhood graph of a node is the ego network (Arnaboldi et al. 2016) of a node (widely used in social network analysis) minus the ego itself. Accordingly, in the context of social network analysis, the vertices comprising the neighborhood graph of a node can be referred to as 'alters' and edges represent the ties among the alters.

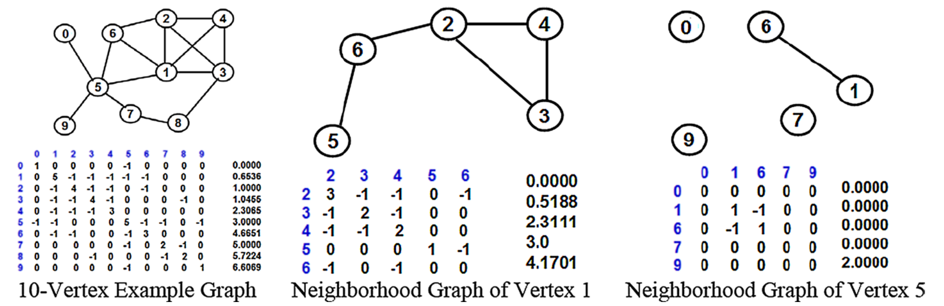
Algebraic connectivity

The algebraic connectivity (AC (Fiedler 1973)) of a network is a measure of the connectivity of the nodes in the network. The AC of a graph is computed by conducting spectral analysis (Mieghem 2010) of the Laplacian matrix of the graph. The Laplacian matrix (Godsil and Royle 2001) of a graph is a square symmetric matrix whose diagonal entries correspond to the degree of the vertex. A non-diagonal entry (i, j) would be -1 if there is an edge between vertices i and j in the graph or 0 if there is no edge between i and j.

Spectral analysis of an *n* x *n* Laplacian matrix would return *n* eigenvalues (in the sorted order, starting from 0) and their corresponding eigenvectors (Mieghem 2010; Godsil and Royle 2001). The eigenvalues (Mieghem 2010; Godsil and Royle 2001) of the Laplacian matrix would be either 0 or positive. The number of 0 s among the eigenvalues of the Laplacian matrix of a graph would correspond to the number of components in the graph (Mieghem 2010; Godsil and Royle 2001). Hence, if the entire graph is connected, there will be only one zero among the eigenvalues of its Laplacian matrix. The smallest non-zero positive eigenvalue of the Laplacian matrix of a graph is a measure of the connectivity of the graph and is referred to as the Algebraic connectivity (AC) (Fiedler 1973). The AC values can range from 4/(*n***D*) to *n*, where *n* is the number of vertices in the graph and *D* is the diameter of the graph (Gross et al. 2013). We refer to the Algebraic Connectivity Ratio (ACR) of a graph as the ratio of its algebraic connectivity to the number of nodes in the graph. The ACR values of a graph would always be in the range [0, …, 1]. Figure 2 presents the Laplacian matrix and the 10 eigenvalues of a 10-vertex running example graph used throughout this section and "Related work and motivation" section (referred to as the motivating example graph in Fig. 1). We observe the eigenvalues output by the spectral analysis program to be in sorted order. There is only one entry for a 0 among the eigenvalues, indicating all the 10 vertices exist as a single component. The smallest positive eigenvalue (0.6536) is the algebraic connectivity of this graph. The ACR value is 0.6536/10 = 0.06536 as it is the ratio of AC and the number of vertices in the graph. The lower ACR value can be attributed to the presence of node 5 and edges 5–9 and 0–5, any of which if removed will disconnect the graph.

Neighborhood graph

The neighborhood graph of a vertex *i* (denoted NG_i) comprises the neighbors of the vertex as the vertices and the edges connecting the neighbors. Note that the neighborhood graph of a vertex does not include the vertex and the edges to its neighbors. Figure 2 presents the neighborhood graphs of vertices 1 and 5 in the running example graph of this section. For vertex 1, all its five neighbors are in a single component. For vertex 5, the neighborhood has a total of four components: vertices 1 and 6 are in one component and vertices 0, 7 and 9 are each in a separate component.



	0	1	2	3	4	5	6	7	8	9	
0	1	0	0	0	0	-1	0	0	0	0	0.0000
1	0	5	-1	-1	-1	-1	-1	0	0	0	0.6536
2	0	-1	4	-1	-1	0	-1	0	0	0	1.0000
3	0	-1	-1	4	-1	0	0	0	-1	0	1.0455
4	0	-1	-1	-1	3	0	0	0	0	0	2.3065
5	-1	-1	0	0	0	5	-1	-1	0	-1	3.0000
6	0	-1	-1	0	0	-1	3	0	0	0	4.6651
7	0	0	0	0	0	-1	0	2	-1	0	5.0000
8	0	0	0	-1	0	0	0	-1	2	0	5.7224
9	0	0	0	0	0	-1	0	0	0	1	6.6069

10-Vertex Example Graph

	2	3	4	5	6	
2	3	-1	-1	0	-1	0.0000
3	-1	2	-1	0	0	0.5188
4	-1	-1	2	0	0	2.3111
5	0	0	0	1	-1	3.0
6	-1	0	-1	0	0	4.1701

Neighborhood Graph of Vertex 1

	0	1	6	7	9	
0	0	0	0	0	0	0.0000
1	0	1	-1	0	0	0.0000
6	0	-1	1	0	0	0.0000
7	0	0	0	0	0	0.0000
9	0	0	0	0	0	2.0000

Neighborhood Graph of Vertex 5

Fig. 2 Examples for neighborhood graph of a vertex, Laplacian matrix and its eigenvalues

Computation of NBNC tuple

We propose to quantify the extent to which a node i can serve the role of a bridge node in the form of a 3-term Neighborhood-based Bridge Node Centrality (NBNC) tuple whose entries are in this order: $\left(NG_i^{\#comp}, NG_i^{ACR}, |NG_i|\right)$, where $NG_i^{\#comp}$ is the number of components in the neighborhood graph NG_i of node i, NG_i^{ACR} is the ratio of the algebraic connectivity of NG_i and the number of nodes in NG_i (the latter represented as $|NG_i|$, which is also the third term in the tuple). We denote the NBNC tuple of a node i as $NBNC(i)$. To compute the $NBNC(i)$, we build the neighborhood graph NG_i of node i and determine the eigenvalues of the Laplacian matrix of NG_i. The number of 0 s among the eigenvalues of the Laplacian matrix of NG_i denote the number of components ($NG_i^{\#comp}$) in NG_i and is the first term in $NBNC(i)$. The smallest non-zero positive eigenvalue of the Laplacian matrix of NG_i denotes the algebraic connectivity (NG_i^{AC}) of NG_i; the ratio $NG_i^{AC}/|NG_i|$ is the algebraic connectivity ratio of NG_i (denoted as NG_i^{ACR}) and is the second term of $NBNC(i)$. We use the NG_i^{ACR} ratio (rather than just NG_i^{AC}) as part of $NBNC(i)$ to compare neighborhoods on the basis of their connectivity vis-a-vis the number of nodes that are part of the neighborhood; the use of NG_i^{ACR} also facilitates comparing neighborhoods on a uniform scale ranging from 0.0 to 1.0.

For vertex 1 in the example graph of Figs. 1 and 2, we observe its neighborhood graph NG_1 to be connected and hence $NG_1^{\#comp} = 1$. The algebraic connectivity of NG_1 is $NG_1^{AC} = 0.5188$ (the smallest non-zero positive eigenvalue of the Laplacian matrix of NG_1) and the algebraic connectivity ratio of NG_1 is $NG_1^{ACR} = NG_1^{AC} / |NG_1| = 0.5188/5 = 0.1038$. The number of vertices in NG_1 is $|NG_1| = 5$. Hence, the NBNC tuple for vertex 1 is $NBNC(1)$ is (1, 0.1038, 5). Likewise, for vertex 5: we observe its neighborhood graph NG_5 to have four components (there are four 0 s among the eigenvalues of NG_5). The smallest non-zero positive eigenvalue of the Laplacian matrix of NG_5 is 2.0000 and $NG_5^{ACR} = 2.0000/5 = 0.4000$. Hence, the NBNC tuple for vertex 5 is $NBNC(5)$ is (4, 0.4000, 5). The NBNC tuples of the ten vertices in the running example graph are presented in Table 4.

NBNC-based ranking of the vertices

In this sub section, we propose the rules to compare the NBNC tuples of two vertices that can be incorporated in any comparison-based sorting algorithm to obtain a ranking

Table 4 NBNC tuples for the nodes in the example graph of Figs. 1 and 2

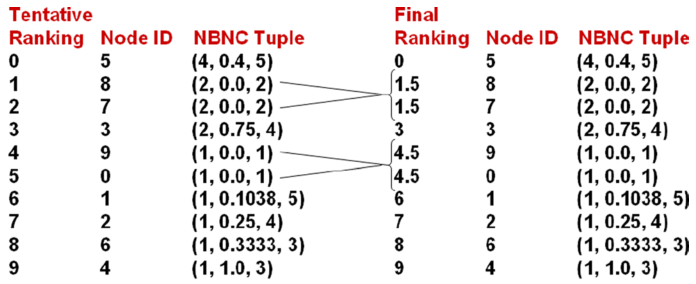
Node ID	***NBNC* tuple**
0	(1, 0.0, 1)
1	(1, 0.1038, 5)
2	(1, 0.25, 4)
3	(2, 0.75, 4)
4	(1, 1.0, 3)
5	(4, 0.4, 5)
6	(1, 0.3333, 3)
7	(2, 0.0, 2)
8	(2, 0.0, 2)
9	(1, 0.0, 1)

of the vertices (a sorted order of the node IDs) with respect to the extent to which they serve the role of a bridge node in the network. The sorting algorithm is to be designed in such a way that vertices which are topologically better qualified to function as bridge nodes are ranked higher and appear earlier in the sorted order. In this pursuit, we define below the '>' operation to compare the NBNC tuples of any two vertices u and v and decide the relative ranking of the two vertices. We propose that for any two vertices u and v, NBNC(v) be considered > NBNC(u) per the following rules:

(1) If $(NG_v^{\#comp} > NG_u^{\#comp})$ then, NBNC(v) > NBNC(u)
(2) If $(NG_v^{\#comp} = NG_u^{\#comp})$ and $(NG_v^{ACR} < NG_u^{ACR})$ then, NBNC(v) > NBNC(u)
(3) If $(NG_v^{\#comp} = NG_u^{\#comp})$ and $(NG_v^{ACR} = NG_u^{ACR})$ and $(|NG_v| > |NG_u|)$ then, NBNC(v) > NBNC(u)

Among the three terms in the NBNC tuple for a vertex v, we give preference to the $NG_v^{\#comp}$ term: a node whose neighborhood graph has several components is an ideal candidate to serve as bridge node. If the $NG_v^{\#comp}$ and $NG_u^{\#comp}$ of two vertices v and u are the same, we break the tie in favor of the vertex that has a lower value for the NG^{ACR} term, a measure of the connectivity of the neighborhood graph vis-a-vis the number of vertices in the graph. A lower NG_v^{ACR} value for a vertex v implies that its neighborhood graph is sparsely connected among the $|NG_v|$ neighbors. If two vertices u and v are still tied on the basis of their $NG^{\#comp}$ and NG^{ACR} values, we break the tie in favor of the vertex with the larger number of nodes in the neighborhood graph (the neighborhood graph of such a vertex is relatively more disconnected among the vertices that are tied on the basis of the first and second terms, and the vertex with a larger degree is more qualified to serve as a bridge node).

If two vertices u and v are tied on the basis of all the three terms in their NBNC tuples (i.e., when NBNC(u) = = NBNC(v)), then we let the sorting algorithm to break the tie in favor of the vertex with the larger ID and generate a sorted array of the node IDs that represents the tentative rankings of the vertices (i.e., the tentative ranking for a vertex is the index for the vertex in the sorted array of node IDs). We eventually generate a final ranking of the vertices as follows: If the NBNC tuple of a vertex is not tied with any vertex, its final ranking is the same as its tentative ranking. For vertices whose NBNC tuples are tied, their final ranking is the average of the tentative rankings of these vertices. The above procedure used to rank tied nodes is adapted from the procedure for calculating the Spearman's rank-based correlation coefficient (the correlation measure used in "Uniqueness of the NBNC tuple" section to assess the uniqueness of the proposed NBNC tuple) (Strang 2006; Lohninger 2021). Figure 3 presents the tentative rankings and final rankings (per the above rules for ranking and tie-breaking procedure) of the ten vertices in the example graph of Figs. 1 and 2. We prefer to use $\Theta(1)$-space complexity in-place sorting algorithms like Heap sort (of time complexity $\Theta(nlogn)$ to sort n elements) or Insertion sort (with a best case time complexity of $\Theta(n)$ and a worst case time complexity of $\Theta(n^2)$) rather than sorting algorithms like Merge sort (of time complexity $\Theta(n\log n)$, but space complexity $\Theta(n)$) that are not in-place with respect to memory requirement, especially when we have to sort the node IDs in networks with a larger number of nodes.



Tentative Ranking	Node ID	NBNC Tuple
0	5	(4, 0.4, 5)
1	8	(2, 0.0, 2)
2	7	(2, 0.0, 2)
3	3	(2, 0.75, 4)
4	9	(1, 0.0, 1)
5	0	(1, 0.0, 1)
6	1	(1, 0.1038, 5)
7	2	(1, 0.25, 4)
8	6	(1, 0.3333, 3)
9	4	(1, 1.0, 3)

Tentative Ranking of the Vertices

Final Ranking	Node ID	NBNC Tuple
0	5	(4, 0.4, 5)
1.5	8	(2, 0.0, 2)
1.5	7	(2, 0.0, 2)
3	3	(2, 0.75, 4)
4.5	9	(1, 0.0, 1)
4.5	0	(1, 0.0, 1)
6	1	(1, 0.1038, 5)
7	2	(1, 0.25, 4)
8	6	(1, 0.3333, 3)
9	4	(1, 1.0, 3)

Final Ranking of the Vertices

Fig. 3 NBNC tuple-based tentative and final rankings of vertices in the example graph of Fig. 1 and 2

Justification of the Rules for Ranking: The formulation of the proposed NBNC tuple incorporates all the three topological positions of a bridge node and a node is ranked as a bridge node on the basis of the extent to which it is critical for connectivity among its neighbors and the number of such neighbors that would be otherwise disconnected without this node. We now justify the order of preference given for the three terms in the NBNC tuple. The first term (the number of components in the neighborhood graph of a node) of the NBNC tuple takes into account the role of a node as both a *bridge*: *hub* node and a *bridge*: *border* node and quantifies the extent to which the neighborhood of a node would otherwise be disconnected without this node. Only a node with larger degree and connected to several other clusters (that would not be otherwise reachable to each other if the node and its incident edges are removed from the graph) is likely to incur a larger value for the number of components in its neighborhood graph. The second term (the algebraic connectivity ratio) of the NBNC tuple is more likely to be lower or even zero for the neighborhood graph of the *bridge*: *between-clusters* nodes that typically have a low-moderate degree, but a larger betweenness. As degree and betweenness centrality metrics exhibit a moderate-strong positive correlation (Meghanathan 2017c), we anticipate a larger degree node with a larger betweenness would get highly ranked as a bridge node on the basis of the first term itself (i.e., such a node is expected to have a larger number of components in its neighborhood graph, justifying its larger betweenness and larger degree). For *bridge*: *hub* nodes (like node 5 in our example graph) with larger degree, but sparse connectivity (with algebraic connectivity ratio greater than zero) among the neighbors, the node could still get ranked higher on the basis of the first term if the number of components involving its neighbors is relatively larger than the rest of the nodes in the network. If there is a relatively fewer number of components in the neighborhood graph of a larger degree node (like node 1), it implies the neighbors are reachable to each other without going through the node (such a node could get classified as a bridge node only based on the second term or the third term: in case of a tie, but not the first term).

Thus, the three terms (considered in the order of preference, as explained above) of the NBNC tuple unequivocally capture the extent to which a node plays the role of a bridge node with respect to the three topological positions (*bridge*: *hub*, *bridge*: *border* and *bridge*: *between-clusters* nodes). To the best of our knowledge, we opine that such a comprehensive ranking of the nodes as bridge nodes cannot be obtained by coming up with a scalar value that is a weighted linear combination of the three terms or an aggregation of the ranking based on the individual terms (Madotto and Liu 2016) or by simply putting together discrete values of related classical metrics (such as degree, betweenness, etc.) as a tuple and coming up with a preference rule for these centrality metrics to rank the nodes on the basis of the extent to which they play the role of a bridge node.

Average and worst case time complexity to compute the NBNC tuples

The NBNC tuple can be locally computed (in parallel) for any vertex in a graph. The Eigenvalues computation of an n x n Laplacian matrix takes O(n^2) time (Fiedler 1973). The Laplacian matrix of the neighborhood of a node i would have dimensions k_i x k_i where k_i is the degree of the node. Hence, the Eigenvalues computation of a k_i x k_i matrix would take O(k_i^2) time. The average case time complexity to compute the NBNC tuples for all the nodes in a network of *'n'* nodes in parallel would be then $O(\sum_{i=1}^{n} k_i^2/n)$. At the worst-case, the Laplacian matrix of the neighborhood of a node will have dimensions k_{max} x k_{max} where k_{max} is the maximum degree for any node in the network. The Eigenvalues computation for a k_{max} x k_{max} Laplacian matrix would take O($k_{\max}^2$) time, which is the worst case time complexity for computing the NBNC tuples in parallel for all the nodes in a network. In theory, k_{max} can be as large as n-1 where n is the number of nodes in the network; however, in practice, k_{max} values for real-world networks (even if scale-free) are expected to be much less than n-1, especially for larger networks (see Table 5 for data on the number of nodes and the maximum degree of a node for the real-world networks analyzed in "Analysis of real-world networks" section). In comparison, the lowest worst case time complexity for any community detection algorithm is O($n$log$n$) (Lancichinetti and Fortunato 2009), corresponding to the Louvain algorithm (Blondel et al. 2008). Several other well-known deterministic community detection algorithms (like Greedy modularity (Chen and Kuzmin 2014), Spectral clustering (Ng et al. 2001), Girvan–Newman (Newman and Girvan 2004), etc.) are of time complexity O(n^2) or O(n^3) or even larger.

Analysis of real-world networks

In this section, we illustrate the computational lightness, effectiveness, efficiency/accuracy and uniqueness of the proposed NBNC tuple to rank the nodes in a real-world network with respect to the extent to which they function as bridge nodes in the network. We consider a suite of 60 real-world networks with diverse degree distributions and domains. For studies on effectiveness and uniqueness of the NBNC tuple, we compare it with four representative metrics chosen from the category of community-aware metrics such as the number of disjoint communities (NDC) (Ghalmane et al. 2019a), the community hub-bridge (CHB) (Ghalmane et al. 2019a), the modular degree centrality (MDC) (Ghalmane et al. 2019b) and the modularity vitality (MVIT) metric (Magelinski et al. 2021) and four representative metrics chosen from the category of community-unaware metrics such as betweenness centrality (BC) (Freeman 1977), bridging centrality (BRC) (Hwang et al. 2008), bridge node centrality (BNC) (Liu et al. 2019) and the

Table 5 Real-world networks used for the analysis

Net. #	Network name	Domain	# Nodes	# Edges	k_{max}	λ_k	Modularity	References
1	Word Adjacency Net	Co-appearance Net	112	425	49	1.733	0.271	Newman (2006)
2	Anna Karnenina Network	Co-appearance Net	138	493	71	2.475	0.371	Knuth (1993)
3	Jazz Band Network	Co-appearance Net	198	2742	100	1.445	0.444	Geiser and Danon (2003)
4	Cat Brain Network	Biological Network	65	730	45	1.197	0.271	Scannell et al. (1995)
5	C. Elegans Neural Net	Biological Network	297	2148	134	1.684	0.401	White et al. (1986)
6	C. Elegans Metabolic Net	Biological Network	453	2025	237	2.943	0.428	Duch and Arenas (2005)
7	Centrality Literature Net	Literature Network	118	613	66	1.859	0.288	Hummon et al. (1990)
8	Copperfield Network	Co-appearance Net	87	406	82	1.831	0.372	Knuth (1993)
9	CS Dept. Aarhus Net	Employee Network	61	219	60	2.120	0.059	Magnani et al. (2013)
10	PhD Theoretical CS Net	Researchers Net	1025	1043	46	3.347	0.923	Johnson (1984)
11	Dolphin Network	Social Network	62	159	12	1.403	0.515	Lusseau et al. (2003)
12	Dutch Literature 1976 Net	Literature Network	35	80	12	1.488	0.300	Nooy (1999)
13	EU Air Transportation Net	Infrastructural Net	418	1999	145	3.811	0.247	Cadrillo et al. (2013)
14	Facebook Network	Social Network	324	2218	58	1.811	0.622	Blagus et al. (2012)
15	Friendship, Hi-Tech Firm	Employee Network	33	147	16	1.440	0.313	Krackhardt (1999)
16	Flensburg Fjord Food Web	Web Network	180	1577	116	1.804	0.318	Zander (2011)
17	FL Food Web Carbon Exc	Web Network	128	2106	110	1.222	0.178	Ulanowicz and Donald (2005)
18	Flying Teams Cadet Net	Social Network	48	170	16	1.210	0.455	Mareno (1960)
19	US Football Network	Game Network	115	613	12	1.011	0.604	Girvan and Newman (2002)
20	College Dorm Fraternity	Social Network	58	967	52	1.111	0.060	Bernard et al. (1979)
21	GD'96 Network	Citation Network	180	229	27	2.375	0.651	Batagelj and Mrvar (2006)
22	Hypertext 2009 Network	Researchers Net	113	2163	97	1.208	0.103	Isella et al. (2011)
23	Infectious Soc. Pattern Net	Social Network	309	1924	47	1.688	0.575	Isella et al. (2011)
24	Java Dependency Network	Collaboration Net	1538	8032	755	4.254	0.470	Batagelj and Mrvar (2006)
25	Karate Club Network	Social Network	34	78	17	1.466	0.416	Zachary (1977)
26	Korea Family Plan Net	Social Network	35	84	13	1.525	0.448	Rogers and Kincaid (1980)
27	Lazega Law Firm Network	Employee Network	71	205	70	2.632	0.001	Lazega (2001)
28	Les Miserables Network	Co-appearance Net	77	254	36	1.820	0.553	Knuth (1993)

Table 5 (continued)

Net. #	Network name	Domain	# Nodes	# Edges	k_{max}	λ_k	Modularity	References
29	Macaque Dominance Net	Biological Network	62	1167	55	1.037	0.097	Takahata (1991)
30	Madrid Train Bomb. Net	Collaboration Net	64	486	29	1.782	0.380	Hayes (2006)
31	Manuf. Comp. Emp. Net	Employee Network	77	2326	76	1.121	0.211	Cross et al. (2004)
32	Mexican Political Network	Political Network	35	117	17	1.225	0.357	Gil-Mendieta and Schmidt (1996)
33	ModMath Network	Social Network	30	61	11	1.257	0.416	Batagelj and Mrvar (2006)
34	US Politics Books Net	Co-appearance Net	105	441	25	1.421	0.521	Krebs (2003)
35	Prim. School Contact Net	Social Network	238	5539	88	1.224	0.360	Gemmetto et al. (2014)
36	Prison Friendship Network	Social Network	67	182	11	1.319	0.572	MacRae (1960)
37	Rhesus Cortical Network	Biological Network	93	2262	92	1.219	0.000	Guimera et al. (2003)
38	San Juan Sur Family Net	Social Network	75	155	12	1.294	0.595	Loomis et al. (1953)
39	Senator Press Release Net	Co-appearance Net	92	477	41	1.566	0.256	Grimmer (2010)
40	Slovenian Magazine Net	Literature Network	124	5972	123	1.048	0.027	Batagelj and Mrvar (2006)
41	Soccer World Cup'98 Net	Game Network	35	118	19	1.448	0.219	Batagelj and Mrvar (2006)
42	Strike Saw Mill Network	Employee Network	24	38	7	1.222	0.562	Michael (1997)
43	Taro Exchange Network	Social Network	22	39	6	1.058	0.447	Schwimmer (1973)
44	UK Faculty Friendship Net	Social Network	81	577	41	1.354	0.449	Nepusz et al. (2008)
45	US Airports 1997 Net	Infrastructural Net	332	2126	139	3.220	0.327	Batagelj and Mrvar (2006)
46	US States Network	Geographical Net	49	107	9	1.247	0.578	Meghanathan (2017a)
47	Residence Hall Friend Net	Social Network	217	1839	56	1.272	0.431	Freeman et al. (1998)
48	Western US Power Grid	Infrastructural Net	4941	6594	19	2.804	0.931	Watts and Strogatz (1998)
49	Wind Surfers Beach Net	Social Network	43	336	31	1.160	0.255	Freeman et al. (1989)
50	World Trade Network	Trade Network	80	875	77	1.377	0.220	Smith and White (1992)
51	Facebook Net. of Politicians	Social Network	5908	41,706	323	1.424	0.867	Rozemberczki et al. (2019)
52	Erdos992 Collaboration Net	Collaboration Net	5094	7515	61	3.821	0.771	Batagelj and Mrvar (2000)
53	Mobile Phone Call Network	Social Network	6809	7680	261	6.897	0.872	Eagle and Pentland (2006)
54	WHOIS Network	Technological Net	7476	56,943	1079	2.323	0.580	Mahadevan et al. (2006)
55	dmela Protein Interac. Net	Biological Network	7393	25,569	190	4.756	0.455	Singh et al. (2008)
56	Grid Fruitfly Network	Biological Network	7274	24,894	176	4.514	0.452	Rossi and Ahmed 2015)

Table 5 (continued)

Net. #	Network name	Domain	# Nodes	# Edges	k_{max}	λ_k	Modularity	References
57	Autonomous Systems Net	Technological Net	6474	13,895	1459	5.083	0.620	Rossi and Ahmed (2015)
58	Gnutella Peer-to-Peer Net	Technological Net	6301	20,777	97	3.000	0.448	Rossi and Ahmed (2015)
59	High Energy Physics Collab	Collaboration Net	9877	51,971	65	2.676	0.761	Rossi and Ahmed (2015)
60	LastFM Users Network	Social Network	7624	27,806	216	5.051	0.814	Rozemberczki and Sarkar (2020)

degree, clustering coefficient and location-based DCL metric (Berahmand et al. 2019). For discussion purposes, the above eight metrics are collectively referred to as "bridgeness" metrics, and the NBNC tuple and the bridgeness metrics are collectively referred to as "bridgeness" measures.

Computational lightness: The NBNC tuple can be asynchronously computed (i.e., just for a particular node without requiring to be computed for all the nodes in the network) as well as in parallel and is hence expected to be computationally light. To illustrate the computational lightness of the NBNC tuple, we compare the average and worst case time complexities incurred in computing the NBNC tuples in parallel for every node in a real-world network with that of the lowest worst case time complexity that can be incurred for any community detection algorithm. ***Effectiveness***: To illustrate the effectiveness of the NBNC tuple and the bridgeness metrics in identifying and ranking bridge nodes on the basis of the extent of their contribution, we determine the smallest value of the fraction (ρ_{min}) of the top-ranked bridge nodes that need to be removed to fragment a real-world network in such a way that the fraction of initial nodes in the largest connected component of the resulting network is below a threshold. ***Efficiency/accuracy***: We anticipate the first term of the NBNC tuple to efficiently (without running any clustering/community detection algorithm) and accurately quantify the number of different clusters in which the neighbor nodes of a node would be part of. In this pursuit, we compute the root mean squared error (RMSE) values for the number of different clusters perceived to exist in the neighborhood of a node as per the NBNC tuple versus the actual number of different clusters in the neighborhood of a node as per a community detection algorithm (we use the well-known Louvain community detection algorithm (Blondel et al. 2008)). ***Uniqueness***: To illustrate the uniqueness of the NBNC tuple, we compute the Spearman's rank-based correlation coefficient (Strang 2006) of the ranking of the vertices on the basis of the NBNC tuples in each real-world network vis-a-vis the bridgeness metrics. We anticipate the Spearman's rank-based correlation coefficient values for NBNC versus a bridgeness metric to be appreciably lower than 1.0 to vindicate the uniqueness of the NBNC tuple in ranking nodes on the basis of the extent to which they play the role of a bridge node.

Real-World Networks

Table 5 lists the 60 real-world networks (with a unique identification number 1–60) along with their name, domain, number of nodes and edges, the spectral radius ratio for node degree (Meghanathan 2014) as well as the modularity scores (Brandes et al. 2008) for these networks (determined using Gephi (2020)), as per the Louvain community detection algorithm. The spectral radius ratio for node degree (denoted: λ_k and $\lambda_k \geq 1.0$) is a measure of the variation in node degree, but unlike standard deviation, the λ_k values are not dependent on the number of nodes or edges. The larger the λ_k value for a network, the greater is the variation in its node degree. Scale-free networks (Barabasi and Albert 1999) are expected to incur a larger λ_k value, while random networks (Erdos and Renyi 1959) are expected to incur λ_k values closer to 1.0. The networks analyzed are spread over several domains, as listed below (inside the parenthesis: we indicate the number of networks analyzed in these domains): Biological networks (7), Citation network (1), Co-appearance networks (7), Collaboration networks (4), Employee networks (5), Game networks (2), Geographical network (1), Infrastructural networks (3), Literature networks (3), Political network (1), Researchers networks (2), Social networks (18), Technological networks (3), Trade network (1) and Web networks (2). All the 60 real-world networks are transformed into undirected graphs. The first 50 real-world networks (Net. #s 1 through 50) have less than 5000 nodes for each of which we compute the NBNC tuple and all the eight bridgeness metrics. The last 10 real-world networks (Net. #s 51 through 60) have more than 5,000 nodes: we do not compute the three computationally heavy betweenness related community-unaware metrics (BC, BRC and BNC) for these ten networks, but compute the NBNC tuple and the other five bridgeness metrics.

Computational lightness of the NBNC tuple

The NBNC tuple for any node can be determined locally for the specific node and it requires only computation of the Eigenvalues of the Laplacian matrix of the neighborhood graph of a node whose size is bounded by the degree of the node. In this sub section, we substantiate the computational lightness of the NBNC approach through actual run-time measurements conducted on the 60 real-world networks. We measured the actual average run-time for the computation of the NBNC tuple per node as well as the actual worst case run-time for the computation of the NBNC tuple for any node for each of the 60 real-world networks and compared them with the actual average run-time it took for the Louvain community detection algorithm to determine the disjoint clusters in these networks. All our implementations were done in Java and the code was run on a desktop Windows 7 computer (Intel i7-2620M CPU @ 2.70 GHz with 8 GB RAM). To account for the NBNC tuple's potential to be computed in parallel, we measure the actual average run-time to compute the NBNC tuple per node in a real-world network as the total of the run-times to compute the NBNC tuples of all the nodes in the network divided by the number of nodes in the network. The worst case run-time for the computation of the NBNC tuple for any node in a real-world network is the maximum of the run-times measured for the individual nodes in the network. The results presented in Table 6 (the run-time values are shown in micro seconds) are average values for 10 executions of the NBNC computations (per the procedure described above) and the Louvain algorithm for each of the 60 real-world networks.

For 51 of the 60 real-world networks (i.e., for more than 5/6th of the real-world networks) and for 26 of the 60 real-world networks (i.e., for more than 2/5th of the

Table 6 Actual average run-time (per node, accounting for parallelization) and worst case run-time (for any node) for the computation of the NBNC tuple versus the actual average run-time for the Louvain community detection algorithm (*all run-times are measured in micro seconds*)

Net. #	NBNC Average	NBNC Worst Case	Louvain Average	Net. #	NBNC Average	NBNC Worst Case	Louvain Average
1	**526.25**	17,396.80	5524.08	31	465.83	2595.06	216.78
2	**248.77**	7391.75	1527.20	32	**23.84**	**53.98**	77.24
3	**1011.34**	13,486.16	2037.22	33	**15.55**	**34.21**	117.36
4	461.64	1259.71	299.87	34	**37.12**	**210.58**	261.18
5	**277.66**	17,949.87	2438.87	35	**1287.68**	5377.89	1304.89
6	**458.56**	118,169.16	8408.95	36	**12.30**	**53.22**	224.16
7	**110.82**	2554.00	476.21	37	1192.59	7253.01	521.10
8	**103.52**	3716.40	309.79	38	**13.47**	**38.77**	298.17
9	**72.60**	998.95	132.05	39	**65.05**	606.67	237.34
10	**9.20**	**150.53**	729.25	40	8348.81	20,317.62	1158.03
11	**26.11**	**49.04**	276.23	41	**24.25**	94.65	64.30
12	**23.80**	**39.53**	143.81	42	**15.51**	**13.30**	144.54
13	**265.46**	16,574.99	2869.35	43	**18.12**	**13.68**	131.71
14	**163.00**	1903.63	1529.07	44	**114.83**	830.56	253.71
15	**54.76**	**140.26**	211.05	45	**422.03**	23,666.83	3405.91
16	**287.17**	10,258.21	1028.03	46	**16.14**	**24.71**	201.86
17	**626.45**	9987.57	739.58	47	**152.55**	1936.32	1039.97
18	**27.46**	**60.44**	124.76	48	**29.84**	**118.60**	19,917.89
19	**50.42**	**96.55**	527.81	49	103.14	365.67	88.77
20	630.69	1577.49	198.24	50	352.03	3601.61	307.25
21	**11.57**	**125.82**	743.12	51	**531.26**	238,219.14	65,190.28
22	825.26	8313.15	680.94	52	**32.85**	**2158.68**	42,756.09
23	**70.73**	**1079.91**	1249.50	53	**75.88**	**39,459.92**	50,222.32
24	**7432.13**	8,218,348.54	132,836.96	54	**8488.45**	22,503,607.50	395,161.64
25	**16.62**	**36.49**	80.79	55	**62.90**	**26,729.05**	70,350.26
26	**19.54**	**57.78**	111.69	56	**97.18**	**184,555.40**	555,970.77
27	**37.62**	778.10	77.62	57	**11,770.01**	65,022,191.97	2,503,644.01
28	**17.84**	220.47	108.25	58	**35.05**	**3791.66**	32,048.08
29	484.22	1128.95	137.79	59	**86.33**	**4563.68**	171,441.53
30	**21.86**	153.57	87.72	60	**144.33**	**79,320.47**	167,157.83

The entries for NBNC Average and NBNC Worst case are represented in bold in Table 6 if their computation times are lower than the computation time for Louvain Average for the corresponding networks

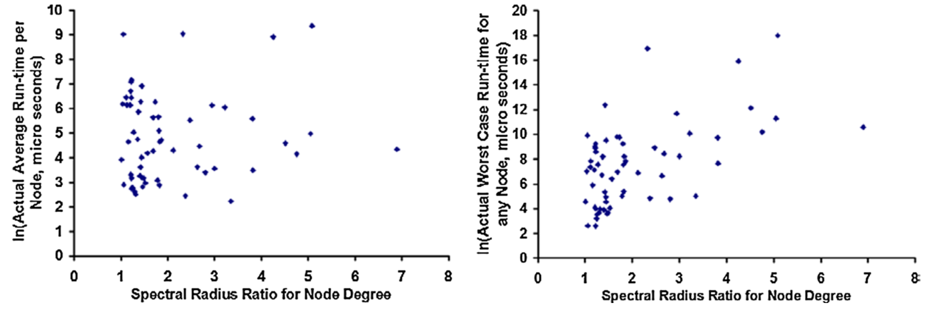


Fig. 4 Impact of spectral radius ratio for node degree on the run-times to compute the NBNC tuples for the real-world networks

real-world networks), we observe the actual average run-time and actual worst case run-time for the computation of the NBNC tuples of the nodes to be respectively lower than the actual average run-time for the Louvain community detection algorithm (we highlight the corresponding cells in Table 6). With regards to the impact of spectral radius ratio for node degree on the run-times to compute the NBNC tuple (see Fig. 4, wherein we plot the logarithm of the run-times to capture all the data points in a single scale), two observations could be made: the actual average run-time per node does not show a particular pattern of increase or decrease with increase in the spectral radius ratio for node degree; whereas, the actual worst case run-time for any node shows a pattern of increase with increase in the spectral radius ratio for node degree (especially, for larger networks) and the latter observation is as expected. The relatively lower values for the actual average run-times per node for more than 5/6th of the real-world networks (including the larger scale-free networks, #s 51–60) and the absence of any correlation between the spectral radius ratio for node degree and the actual average run-time per node portend well for the use of the NBNC-approach (local and synchronous) rather than a community detection algorithm (global and synchronous) to efficiently decide the *extent of bridgeness* for the majority of the nodes in larger scale-free real-world networks.

Effectiveness of the NBNC tuple

In this sub section, we evaluate the effectiveness of the NBNC tuple vis-a-vis the eight bridgeness metrics with respect to meticulously identifying the bridge nodes in a real-world network. We adapt the fragmentation approach of da Cunha et al. (2015) to evaluate the effectiveness of the bridgeness measures in identifying the bridge nodes in the real-world networks of Table 5. Under the original fragmentation approach of Cunha et al. (2015): one needs to first run a community detection algorithm and identify the nodes that connect two or more communities; sort these nodes in the decreasing order of their values with respect to a bridgeness measure and the *attack* (network fragmentation) process then begins with the removal of one node at a time, starting from the first node in the sorted list. A plot of the fraction ρ of nodes removed versus the fraction σ of the original nodes in the largest component of the remaining network of nodes gets evolved with each node removal. When all the nodes in the list have been removed, the attack process stops. For a chosen value of σ, the bridgeness measure that required the lowest value for the fraction ρ of nodes to be removed is considered the most effective.

Since NBNC and the community-unaware metrics do not require a community detection algorithm to be run a priori, we cannot apply the above-described fragmentation approach as it is. Hence, we propose a modified version of the fragmentation approach that can be seamlessly applied for all the bridgeness measures. We developed a binary search algorithm to determine the lowest value for the fraction ρ of nodes to be removed (denoted by ρ_{min}) that would result in the σ value falling below a targeted threshold, σ_{thresh} (set as 0.05 in this paper) for a real-world network.

With a search space of [0.0, ..., 1.0], the binary search algorithm maintains the following invariants throughout its execution: (1) For any value of the left index (*LI*, initialized to 0.0) fraction of nodes removed from the network, the fraction of nodes in the largest component of the resulting network is always greater than or equal to σ_{thresh}. (2) For any value of the right index (*RI*, initialized to 1.0) fraction of nodes removed from the network, the fraction of nodes in the largest component of the resulting network is always less than σ_{thresh}. In each iteration of the binary search algorithm, we determine the middle index (*MI*) as the average of the latest values of *LI* and *RI*. We remove the *MI* fraction of nodes from the network and determine the fraction of nodes in the largest component of the resulting network; if this fraction is less than σ_{thresh}, we move the *RI* to the left and set $RI = MI$; if the fraction is greater than or equal to σ_{thresh}, we move the *LI* to the right and set $LI = MI$. We continue the iterations until the absolute difference between the *RI* and *LI* is within a threshold (we use a value of 0.01 for the threshold difference between the *RI* and *LI* in the binary search algorithm). The latest value of *RI* at the time of termination of the algorithm is the smallest value of ρ (ρ_{min}) that would lead to the σ value falling below σ_{thresh} (0.05).

The lower the ρ_{min} value ($0.0 \leq \rho_{min} \leq 1.0$) for a bridgeness measure, the more effective is the measure in identifying the bridge nodes in a network. With a search space ranging from 0.0 to 1.0 and 0.01 as the threshold difference between the *RI* and *LI* for the algorithm to terminate, the number of iterations of the binary search algorithm for *any* given bridgeness measure and real-world network is just $\log_2^{((1.0-0.0)/0.01)} \sim 7$. We used the above binary search algorithm to measure the effectiveness (ρ_{min}) for the NBNC tuple and each of the eight bridgeness metrics for real-world networks 1 through 50 as well as for the NBNC tuple and the four community-aware bridgeness metrics and the DCL community-unaware metric for real-world networks 51 through 60 (see Fig. 5).

There exists one caveat with the use of a threshold value for σ to evaluate the effectiveness of a bridgeness measure. It is possible that the ρ versus σ plot for a bridgeness measure on a real-world network could start with a steep drop, but then slowly decrease (or almost stay flat) to eventually incur a larger ρ_{min} value before the σ value falls below the threshold σ_{thresh}. In such cases, the ρ_{min} value determined with the binary search approach would only serve as an approximate estimate of the effectiveness of the bridgeness measure. Nevertheless, the binary search approach proposed here is very time-efficient (in terms of the number of iterations needed) and is a viable option for larger networks and more expensive bridgeness measures, especially the ones that are variants of betweenness centrality. More accurate methods exist in the literature (for example, the approach used in Magelinski et al. (2021) overcomes the "steep drop, then almost flat line" scenarios) to determine effectiveness, but at the cost of computational power and time.

It is to be also noted that the binary search approach requires the values for the nodes with respect to a bridgeness measure to be computed "a priori" before the beginning of the node removals. Such a node removal strategy has been referred to as "simultaneous attack on the network" in Cunha et al. (2015) and is considered to reveal the structural weaknesses of the network (which is also the focus of our research in this paper). Cunha et al. (2015), the authors observe that the alternate strategy (referred to as *sequential attack on the network*) of recomputing the bridgeness measures for the residual nodes (nodes that are not yet removed) in a network is more useful to analyze the dynamical properties of complex networks, which is not our focus in this research.

Figure 5 presents the results of the study in the form of a 1 + plot for each of the 60 real-world networks (the network #s listed in Table 5 are indicated alongside): the larger yellow-colored circle represents the effectiveness incurred with the NBNC

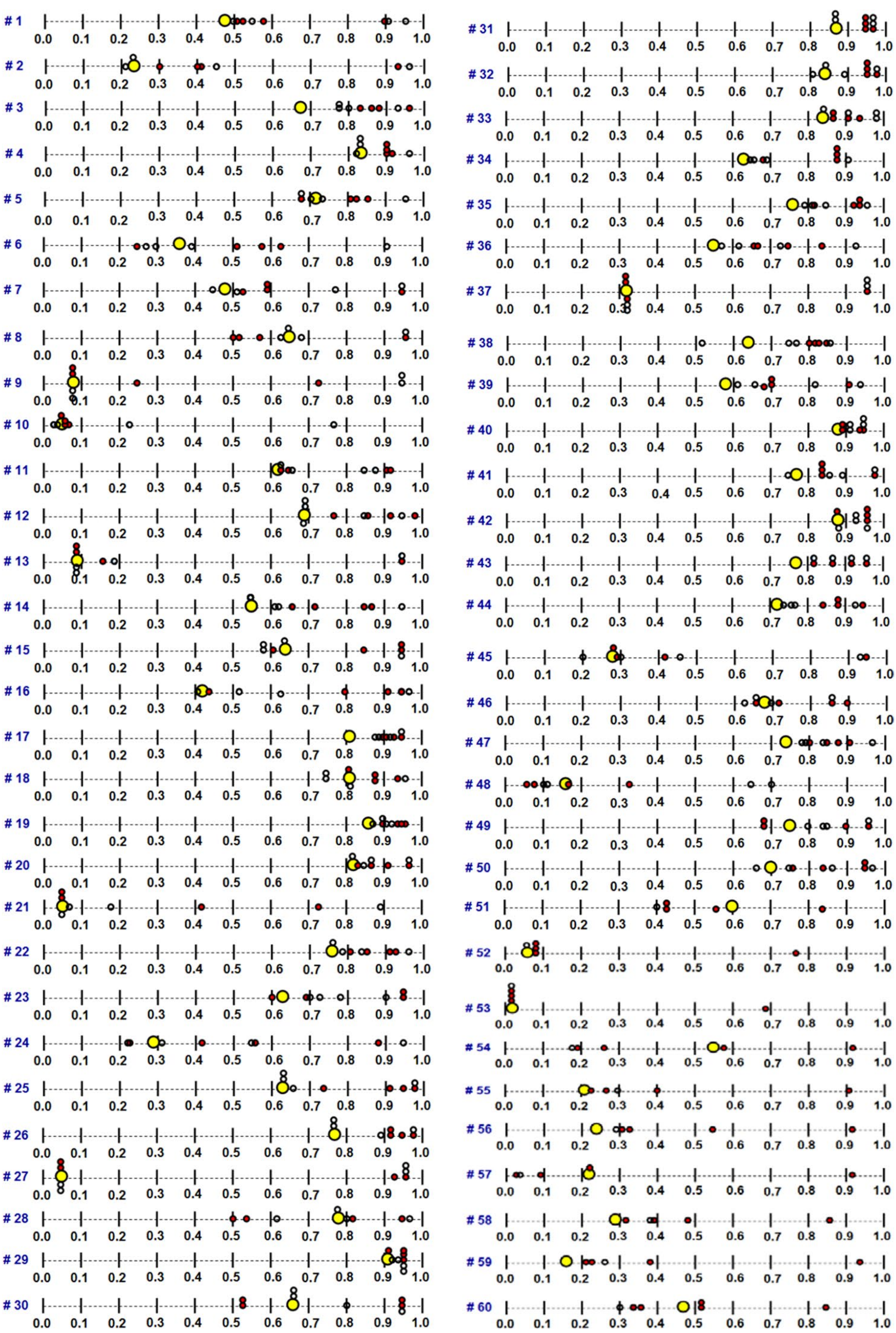


Fig. 5 Effectiveness of NBNC tuple versus community-aware and community-unaware bridgeness metrics

tuple; the red-colored smaller circles represent the effectiveness incurred with the community-aware metrics and the colorless circles represent the effectiveness incurred with the community-unaware metrics. In case of a tie (i.e., more than one bridgeness measure incurring the same effectiveness value), we pileup the circles corresponding to these measures. We observe the NBNC tuple to incur the lowest of the ρ_{min} values for 34 of the 60 real-world networks (i.e., for more than 50% of the real-world networks studied). NBNC's closest competitor is the BC metric (Betweenness Centrality) that incurs the lowest of the ρ_{min} values for 17 of the first 50 real-world networks (i.e., for about 1/3rd of the networks). The DCL and MDC metrics each incur the lowest of the ρ_{min} values for 16 of the 60 real-world networks. The NDC, MVIT and BRC metrics are observed to be least effective in disconnecting networks when nodes are removed based on their ranking as bridge nodes with respect to these metrics. Relatively, between the community-aware and community-unaware metrics, we observe the community-unaware centrality metrics to be more effective in identifying the appropriate nodes as bridge nodes in the network. As we claimed earlier, each of the eight bridgeness metrics considered only at most two of the three topological positions of a bridge node, whereas the proposed NBNC tuple comprehensively captures the extent to which a node serves as bridge node in all the three topological positions. The results seen in this sub section justify our claim.

Figure 6 presents a plot of the spectral radius ratio for node degree versus the effectiveness of the NBNC tuple. We observe the NBNC tuple to be relatively more effective for networks with larger values for the spectral radius ratio for node degree (a characteristic of scale-free networks). For example, we observe the US Airports Networks ($\lambda_k = 3.21$, whose degree distribution exhibits a power-law pattern, characteristic of scale-free networks) to incur a ρ_{min} value of 0.2813; whereas, the Football Network ($\lambda_k = 1.01$, whose degree distribution exhibits a bell-shaped-curve pattern, characteristic of random networks) incurs a ρ_{min} value of 0.86. As many large real-world networks show (at least approximately) scale-free structure, we opine the trend observed in Fig. 6 augurs well for use of the NBNC tuple with larger real-world networks too. The λ_k versus ρ_{min} values for real-world networks 51 through 60 provide credence to our claim.

Efficiency/accuracy of the NBNC tuple

In this sub section, we illustrate the efficiency (i.e., without running any clustering/community detection algorithm) of the NBNC tuple with respect to accurately quantifying the number of non-overlapping or disjoint clusters in which the neighbor nodes of a node would be part of. There is currently no direct approach available in the literature to

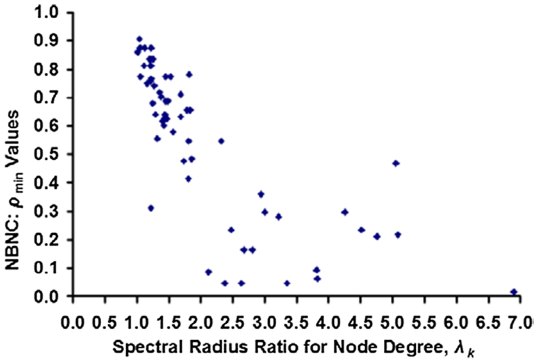
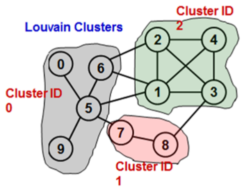


Fig. 6 NBNC tuple: spectral radius ratio for node degree versus effectiveness

determine the number of disjoint clusters in the neighborhood of a node. When posed with such a question, the approach one would hitherto take is to run a clustering algorithm to first determine the non-overlapping clusters of a network, identify the cluster memberships of the nodes and then determine the number of different clusters that the neighbors of a node are part of. On the other hand, the first entry in the NBNC tuple is a direct measure of the number of (disjoint) components that could span the neighborhood of a node. Our premise is that if two vertices in the neighborhood graph of a node are not reachable to each other, they are more likely to exist in two different clusters/communities among the clusters/communities identified by any clustering/community detection algorithm that is designed to determine non-overlapping clusters/communities. Accordingly, the number of components in the neighborhood graph of the nodes are perceived to be the same as the number of disjoint clusters to which the neighbors of the nodes are expected to belong to.

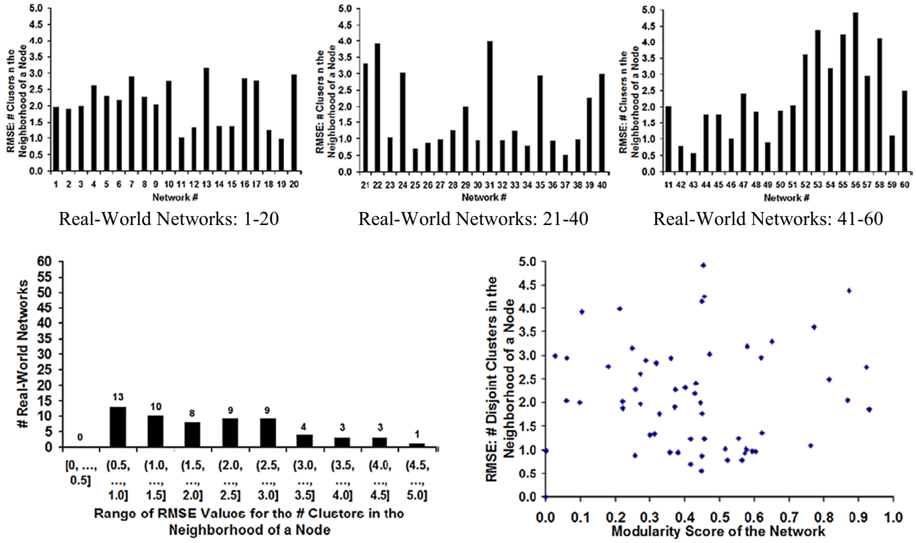
In this sub section, we showcase that the NBNC tuple could be used to efficiently and accurately estimate the number of disjoint clusters that the neighbors of a node would be part of *without running any clustering algorithm*. We quantitatively evaluate the extent to which the number of different components (perceived to be identified as different clusters that would need to be connected by a node i) according to the NBNC tuple of node i would be the same as the actual number of clusters that are observed to exist in the neighborhood of node i according to a clustering algorithm. We use the well-known Louvain community detection algorithm (computationally light among the existing community detection algorithms and is of time complexity O($n$log$n$) for a network of n nodes) to determine the actual number of clusters for the real-world networks. We used Gephi (2020) (that employs the Louvain algorithm as the community detection algorithm (Blondel et al. 2008) to determine non-overlapping communities/clusters) to determine the clusters in each of the 60 real-world networks and identified the cluster IDs of the nodes: using this information, we determined the actual number of clusters in which the neighbors of each node are part of. Recent research (Traag et al. 2019) has indicated that the Louvain algorithm could potentially identify communities that are disconnected within. We verified the communities identified by the Louvain algorithm implementation of Gephi for each of the 60 real-world networks and did not observe any such disconnectedness in the communities identified for any real-world network.

For each real-world network, we determined the root mean square error (RMSE) in the values for the number of components for the NBNC tuple of the nodes (perceived as the number of disjoint clusters in the neighborhood of the nodes) and the number of disjoint clusters actually observed to exist among the neighbors of a node on the basis of the Louvain community detection algorithm. Ideally, we expect these two numbers to be identical for a node and thereby the RMSE value for a real-world network is expected to be 0. However, for a real-world network, it is difficult to expect the difference between the above two numbers to be zero for every node in the network (i.e., the RMSE value for a real-world network is likely to be greater than 0). We prefer the RMSE value for the number of disjoint clusters perceived (on the basis of the number of components of the NBNC tuple) versus the number of disjoint clusters observed to actually exist in the neighborhood of the nodes in a real-world network to be as close to 0 as possible.



Node ID	Neighbors	Cluster ID	Clusters of Neighbors	# Clusters	# Components	Absolute	Square of the
		(Louvain)	(Louvain)	(Louvain)	(NBNC)	Difference	Abs. Diff.
0	5	0	{5}	1	1	0	0
1	2, 3, 4, 5, 6	2	{2, 3, 4}; {5, 6}	2	1	1	1
2	1, 3, 4, 6	2	{1, 3, 4}; {6}	2	1	1	1
3	1, 2, 4, 8	2	{1, 2, 4}; {8]	2	2	0	0
4	1, 2, 3	2	{1, 2, 3}	1	1	0	0
5	0, 1, 6, 7, 9	0	{0, 6, 9}; {1}; {7}	3	4	1	1
6	1, 2, 5	0	{1, 2}; {5}	2	1	1	1
7	5, 8	1	{5}; {8}	2	2	0	0
8	3, 7	1	{3}; {7}	2	2	0	0
9	5	0	{5}	1	1	0	0
							Sum: 4

Fig. 7 Louvain clusters for the example graph of Fig. 1 and the Computation of the root mean square error (RMSE) value for the number of disjoint clusters in the neighborhood of a node



Real-World Networks: 1-20

Real-World Networks: 21-40

Real-World Networks: 41-60

Distribution of the Real-World Networks with respect to the Ranges of the RMSE Values

Modularity Scores of the Real-World Networks vs. the RMSE Values

Fig. 8 Root mean square error (RMSE) values for # disjoint clusters in the neighborhood of a node: NBNC approach versus Louvain community detection algorithm for the real-world networks

Figure 7 presents the computation procedure for the RMSE value for the number of disjoint clusters in the neighborhood of the nodes in the example graph of Fig. 1. We calculate the sum (4) of the squares for the absolute differences between the number of components in the neighborhood graph of a node per the NBNC tuple (perceived as the number of disjoint clusters in the neighborhood of a node) and the number of disjoint clusters actually observed in the neighborhood of a node. The ratio (mean square error) is 4/10 = 0.4, where 10 is the number of nodes in the graph. The square root of the mean square error value of 0.4 is 0.63, which is the RMSE value for the perceived versus actually observed number of disjoint clusters in the neighborhood of a node. Similarly, lower RMSE values were observed for the 60 real-world networks (see Fig. 8).

Figure 8 presents the RMSE values observed for the 60 real-world networks, the distribution of the values for different ranges as well as the distribution of the modularity scores of the networks (listed in Table 5) versus the RMSE values. The median of the RMSE values is 1.996 and the RMSE value for any real-world network is less than 5.0. We notice 32 of the 60 real-world networks to incur RMSE values less than or equal to

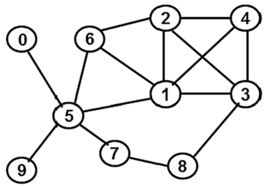
2.0 and 49 of the 60 real-world networks to incur RMSE values less than or equal to 3.0. Interestingly, 12 of the 18 social networks incurred RMSE values less than 2.0 and 16 of the 18 social networks incurred RMSE values less than 3.0. Thus, the lower RMSE values observed for the real-world networks corroborate our claim that the number of disjoint components in the neighborhood graph of a node determined as part of the NBNC tuple is a reliable estimate for the number of disjoint clusters spanning the neighbors of a node that need to be determined using a clustering algorithm. In Fig. 8, we also plot the distribution of the modularity scores of the networks versus the RMSE values incurred. When separately viewed (i.e., for Net. #s 1-50 and for Net. #s 51-60), the RMSE values appear to exhibit a slight tendency to decrease with increase in the modularity scores of the networks in each of the two sets. But, when viewed together (i.e., for Net. #s 1-60), we do not see any appreciable correlation between the modularity scores of the networks and the RMSE values incurred with the NBNC tuple. Overall, we claim the NBNC tuple approach can be efficiently and accurately used (i.e., with lower RMSE values) for both modular and non-modular networks.

Uniqueness of the NBNC tuple

In this sub section, we seek to demonstrate the uniqueness of the NBNC tuple vis-a-vis the existing bridgeness metrics by analyzing the correlations in their rankings of the vertices. We use the Spearman's rank-based correlation coefficient measure (Strang 2006) to assess the equivalence in the rankings of the nodes based on the NBNC tuple versus all the eight bridgeness metrics for real-world networks 1 through 50 as well as for the NBNC tuple versus the four community-aware bridgeness metrics and the DCL community-unaware metric for real-world networks 51 through 60 (see Fig. 10).

For all the eight bridgeness metrics, the larger the metric value for a vertex, the more suitable is the vertex for the role of a bridge node. For any real-world network, we first obtain a tentative ranking of the vertices with respect to a bridgeness metric, with the ties broken in favor of the vertices with the larger ID. For vertices whose bridgeness metric value is different from that of others, their final ranking is the same as their tentative ranking. For vertices (referred to as tied vertices) that incur identical values for a bridgeness metric, their final ranking is the average of the tentative rankings of the corresponding tied vertices. Note that one could also adopt any other procedure to break the ties among the vertices as long as vertices with identical values for a bridgeness metric are eventually assigned the same rank before computing the rank-based correlation coefficient.

For a data set of '*n*' elements, the formula for the Spearman's rank-based correlation coefficient is given by: $1 - \frac{6 * \sum_n d^2_{\substack{final\\rank}}}{n(n^2-1)}$, where $d^2_{\substack{final\\rank}}$ indicates the squares of the absolute difference in the final rankings of the *n* elements with respect to the two measures considered for the correlation analysis. The range of possible values for the Spearman's rank-based correlation coefficient is − 1 to 1 (Strang 2006). Correlation coefficient values of 0.8 or higher (or − 0.8 or lower) are typically considered indicators for a strong positive (or negative) correlation; correlation coefficient values in the range of [0.6, ..., 0.8), [0.4, ..., 0.6) and (0, ..., 0.4) are typically considered indicators for moderate, weak and very



Node ID	NBNC Tuple	Tentative	Final	B'C	Tentative	Final	Abs. Difference in	Square of
		Rank	Rank		Rank	Rank	Final Rank	the Difference
0	(1, 0.0, 1)	5	4.5	0	9	8	3.5	12.25
1	(1, 0.1038, 5)	6	6	9.7	1	1	5	25
2	(1, 0.25, 4)	7	7	1.33	6	6	1	1
3	(2, 0.75, 4)	3	3	4.5	2	2	1	1
4	(1, 1.0, 3)	9	9	0	8	8	1	1
5	(4, 0.4, 5)	0	0	18.5	0	0	0	0
6	(1, 0.3333, 3)	8	8	1.83	5	4.5	3.5	12.25
7	(2, 0.0, 2)	2	1.5	3.33	3	3	1.5	2.25
8	(2, 0.0, 2)	1	1.5	1.83	4	4.5	3	9
9	(1, 0.0, 1)	4	4.5	0	7	8	3.5	12.25
								Sum = 76

Fig. 9 Computation of the Spearman's rank-based correlation coefficient between NBNC and BC for the vertices in the example graph of Fig. 1

weak positive correlation respectively (and likewise, the negative scales for negative correlation) (Strang 2006).

The procedure to compute the Spearman's rank-based correlation coefficient is illustrated in Fig. 9 with regards to the NBNC tuple versus BC-based ranking of the vertices for the example graph of "Related work and motivation" section. We observe the NBNC-BC rank-based correlation coefficient to be 0.54, indicating a weak correlation between the two of them. Consider node 1 in the graph of Fig. 1: node 1 (with the second largest BC value of 9.7) is ranked # 1 with respect to BC in a rank-scale ranging from 0 to 9. However, the five neighbors of node 1 will stay connected even if node 1 and its associated edges are removed from the graph, implying the existence of an appreciable number of edges among these neighbors in order for the neighborhood graph of node 1 to be connected. Such neighbor nodes are likely to be part of the same cluster and node 1 is not critical for their connectivity: justifying a relatively lower ranking (ranking of 6 in a scale of 0 to 9) for node 1 as a bridge node with respect to the NBNC tuple.

Figure 10 displays the Spearman's rank-based correlation coefficient values obtained for the real-world networks with respect to the ranking of the bridge nodes on the basis of the NBNC tuple versus each of the four community-aware bridgeness metrics (in red colored circles) and the four community-unaware bridgeness metrics (in colorless circles). The uniqueness of the NBNC approach is justified by the magnitude of the rank-based correlation coefficient values. About 42% of the 450 correlation coefficient values (#1-#50 real-world networks and 8 bridgeness metrics; #51-#60 real-world networks and 5 bridgeness metrics) fall in the category of very weak correlation, whereas only 12% of the correlation coefficient values fall in the category of strong correlation, indicating the correlation between NBNC and the existing bridgeness metrics is predominantly very weak rather than strong. Thus, the NBNC approach (primarily, to determine the number of components in the neighborhood graph of a node and quantify the extent of connectivity of the neighborhood graph of a node) is a unique attempt at quantifying the importance of nodes as bridge nodes in a complex network.

Relatively, with regards to the community-aware versus community-unaware bridgeness metrics, NBNC appears to be more positively/strongly correlated with the community-unaware metrics. This could be attributed to the formulations of the community-unaware metrics to consider nodes with a lower degree but a larger betweenness or lower clustering coefficient (i.e., the role played as *bridge*: *between-clusters*

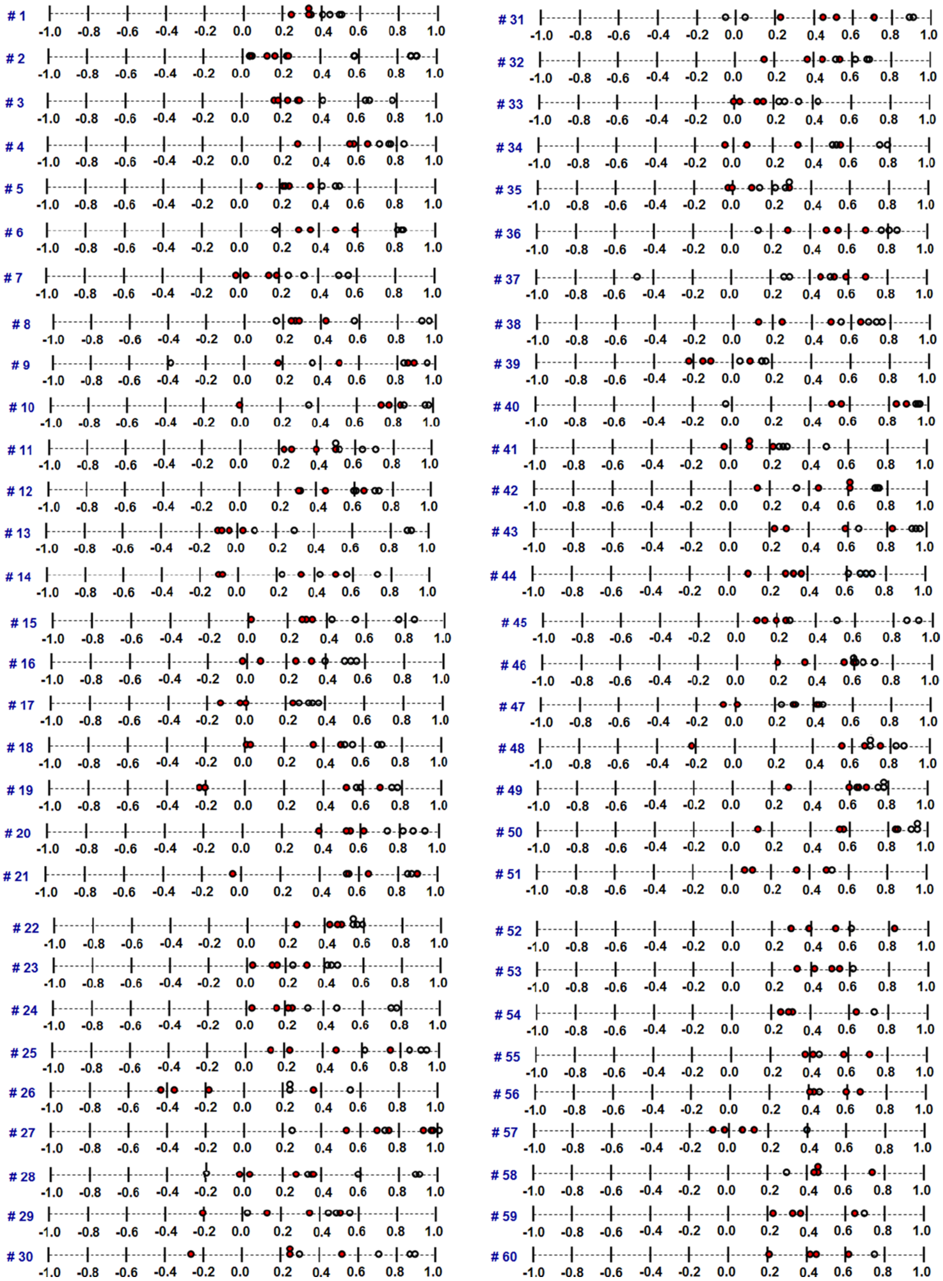


Fig. 10 Spearman's rank-based correlation coefficient values for the real-world networks: NBNC versus community-aware and community-unaware bridgeness metrics

nodes in the NBNC parlance) as bridge nodes of the network. A key weakness with a majority of the community-aware metrics is that they completely ignore the role played as *bridge*: *between-clusters* nodes and focus primarily on the role played as *bridge*: *hub* nodes and *bridge*: *border* nodes. On the other hand, the community-unaware metrics give more importance to both the *bridge*: *between-clusters* nodes and *bridge*: *border* nodes, but fail to take note of the role as *bridge*: *hub* nodes, just simply because the latter have a larger degree.

Conclusions and future work

The high-level contribution of this paper is the proposal of a centrality tuple (instead of a scalar centrality metric or a vector of classical centrality metrics that is not converted to a scalar) for the first time in the literature of complex network analysis. The proposed centrality tuple referred to as Neighborhood-based Bridge Node Centrality (NBNC) tuple can be used to rank nodes on the basis of the extent to which a node could function as a bridge node with respect to three different topological positions (*bridge*: *hub* nodes, *bridge*: *border* nodes and *bridge*: *between-clusters* nodes) identified in this research. We show that the extent of contributions of a node as bridge node in these three topological positions cannot be comprehensively captured using the approaches (that are categorized as community-aware and community-unaware) taken by the existing bridgeness centrality metrics.

We propose the notion of the neighborhood graph (NG) of a node (comprising of the neighbors of the node as vertices and the links connecting them as edges) and formulate the NBNC tuple for a node to be an amalgamation of three terms: the number of disjoint components in the NG of the node, the connectivity (measured as algebraic connectivity ratio) of the NG of the node and the number of nodes in the NG of the node (essentially, the degree of the node). We propose a ranking rule that prefers nodes with a larger number of disjoint components in their NG to be ranked high as bridge nodes and break any ensuing ties in favor of nodes whose NG has lower connectivity (and further break any ties in favor of nodes with a larger degree). Though there is no one-to-one correspondence between any term in the NBNC tuple with any of the three topological positions for a bridge node, we claim that the above ranking rule applied for a NBNC tuple would comprehensively cover the contributions of a node as bridge node with respect to all the three topological positions envisioned in this research. The NBNC tuple for a node can be asynchronously computed based on the two-hop local neighborhood knowledge for a node (hence, it is a computationally light approach); whereas, most of the existing bridgeness metrics in the literature need to be computed synchronously and some are computationally heavy.

We considered a suite of 60 real-world networks of diverse domains for our analysis and evaluated the computational lightness, effectiveness, efficiency/accuracy and uniqueness of the NBNC tuple. ***Computational lightness***: We measured the average of the actual run-times of the procedure to compute the NBNC tuple per node in the network as well as the maximum (worst case time complexity) of the actual run-times to compute the NBNC tuple for any node in the network. We observed the average of the actual run-times and the maximum of the run-times to be lower than the average of the run-times to determine the actual clusters in the networks using the Louvain community detection algorithm (considered to incur the lowest of the time complexity among the community detection algorithms in the literature) for more than 5/6th and 2/5th of the 60 real-world networks respectively. ***Effectiveness***: We validate our claim (the proposed NBNC tuple can effectively identify the bridge nodes in a network compared to the existing bridgeness metrics) by computing the smallest value (ρ_{min}) for the fraction of the top-ranked bridge nodes that need to be removed to bring down the fraction of nodes in the largest component of the resulting network (i.e., the network after the removal of the top-ranked bridge nodes) below a lower threshold value (0.05 is the threshold value used for evaluation purposes). We observe the NBNC tuple to outperform the eight different bridgeness metrics (both community-aware and community-unaware metrics) by incurring the lowest of the ρ_{min} values for 34 of the 60 real-world networks, while the closest competitor betweenness centrality (BC) was observed to incur the lowest of the ρ_{min} values for 17 of the first 50 real-world networks. We observe the NBNC tuple to be relatively more effective for scale-free networks. ***Efficiency/accuracy***: We demonstrate that the first term of the NBNC tuple of a node in a real-world network can be perceived as an efficient and accurate measure of the number of disjoint clusters that could exist in the neighborhood of the node *without running any clustering algorithm* and evaluate the accuracy of such a perception by computing the root mean square error (RMSE) value vis-a-vis the number of disjoint clusters in the neighborhood of a node that are actually observed through a community detection algorithm run on the real-world network. The RMSE values were not more than 5.0 and were less than 3.0 for 49 of the 60 real-world networks analyzed in this research. We observe the RMSE values to be almost independent of the modularity scores of the real-world networks, indicating the NBNC tuple to be equally efficient/accurate for both modular and non-modular networks. ***Uniqueness***: We also computed the Spearman's rank-based correlation coefficient values for the ranking of the vertices based on the NBNC tuple vis-a-vis eight different bridgeness metrics proposed in the literature and predominantly observed a very weak correlation rather than a strong correlation, justifying the uniqueness in the NBNC tuple approach.

As part of future work, we plan to apply the proposed NBNC tuple to identify and employ bridge nodes for various domain-specific problems of complex real-world networks, like: vaccination to prevent/reduce infection spread in epidemics/pandemics-affected community networks (Liu and Hu 2005), enhancement of collaboration among diverse researchers in co-authorship networks, friendship suggestion in social networks, protein folding in protein–protein interaction networks, cluster head selection and hierarchical data gathering in wireless sensor networks, and etc. We also plan to evaluate the effectiveness of the NBNC tuple to rank bridge nodes in networks with overlapping communities by using relevant approaches of Kumar et al. (2018), Chakraborty et al. (2016) and CSoNet (2016) proposed in the literature. Sciarra et al. (2018), the authors had proposed a statistical-estimation based approach to evaluate the importance of nodes in a complex network and demonstrate its application to deduce the degree and eigenvector centrality metrics. As part of future work, we plan to examine the suitability of this approach to deduce the bridgeness centrality metrics and the NBNC tuple as well.

Abbreviations

NBNC: Neighborhood-based bridge node centrality; DC: Degree centrality; EC: Eigenvector centrality; BC: Betweenness centrality; CC: Closeness centrality; BRC: Bridging centrality; BNC: Bridge node centrality; RBW: Route betweenness; BCO: Bridging coefficient; BCE: Bridgeness coefficient; DCL: Degree, clustering coefficient and location-based centrality; RMSE: Root mean square error; NG: Neighborhood graph; AC: Algebraic connectivity; ACR: Algebraic connectivity ratio; DSHC: Degree and structural hole count; CHB: Community-hub bridge; NDC: Number of disjoint communities; MDC: Modular degree centrality; MVIT: Modularity vitality.

Acknowledgements

Not applicable.

Authors' contributions

NM was solely responsible for all work presented in this paper.

Funding
This work was partially supported through the sub contract received from University of Virginia titled: Global Pervasive Computational Epidemiology, with the National Science Foundation as the primary funding agency. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the funding agencies.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Competing interests
The author declares that they have no competing interests.

Received: 5 January 2021 Accepted: 16 June 2021
Published online: 30 June 2021

References

Arnaboldi V, Conti M, La Gala M, Passarella A, Pezzoni F (2016) Ego network structure in online social networks and its impact on information diffusion. Comput Commun 76:26–41
Barabasi A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5349):509–512
Batagelj B, Mrvar A (2000) Some analyses of Erdos collaboration graph. Soc Netw 22(2):173–186
Batagelj V, Mrvar A (2006) Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/
Berahmand K, Bouyer A, Samadi N (2019) A new local and multidimensional ranking measure to detect spreaders in social networks. Computing 101:1711–1733
Bernard HR, Killworth PD, Sailer L (1979) Informant accuracy in social network data IV: a comparison of clique-level structure in behavioral and cognitive network data. Soc Netw 2(3):191–218
Blagus N, Subelj L, Bajec M (2012) Self-similar scaling of density in complex real-world networks. Phys A 391(8):2794–2802
Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exp 2008(10):P10008
Bonacich P (1987) Power and centrality: a family of measures. Am J Sociol 92(5):1170–1182
Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, Wagner D (2008) On modularity clustering. IEEE Trans Knowl Data Eng 20(2):172–188
Bucur D (2020) Top influencers can be identified universally by combining classical centralities. Sci Rep 10:1–14 (**article number: 20550**)
Cadrillo A, Gomez-Gardenes J, Zanin M, Romance M, Papo D, Pozo F, Boccaletti S (2013) Emergence of network features from multiplexity. Sci Rep 3(1344):2013
Chakraborty DD, Singh AA, Cherifi H (2016) Immunization strategies based on the overlapping nodes in networks with community structure. In: Nguyen HH, Snasel V (eds) Computational social networks, CSoNet 2016, Lecture Notes in Computer Science, vol 9795. Springer, Cham
Chen M, Kuzmin K (2014) Community detection via maximization of modularity and its variants. IEEE Trans Comput Soc Syst 1(1):46–65
Cormack RM (1971) A review of classification. J R Stat Soc Ser A (general) 134(3):321–367
Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge, MA
Cross RL, Parker A, Cross R (2004) The hidden power of social networks: understanding how work really gets done in organizations, 1st edn. Harvard Business Review Press, Brighton
da Cunha BR, González-Avella JC, Goncalves S (2015) Fast fragmentation of networks using module-based attacks. PLoS ONE 10(11):e0142824
de Nooy W (1999) A literary playground: literary criticism and balance theory. Poetics 26(5–6):385–404
Duch J, Arenas A (2005) Communication detection in complex networks using extremal optimization. Phys Rev E 72:027104
Eagle N, Pentland A (2006) Reality mining: sensing complex social systems. Pers Ubiquit Comput 10(4):255–268
Erdos P, Renyi A (1959) On random graphs I. Publ Math 6:290–297
Fiedler M (1973) Algebraic connectivity of graphs. Czechoslov Math J 23(98):298–305
Freeman L (1977) A set of measures of centrality based on betweenness. Sociometry 40(1):35–41
Freeman L (1979) Centrality in social networks: conceptual classification. Soc Netw 1(3):215–239
Freeman LC, Freeman SC, Michaelson AG (1989) How humans see social groups: a test of the Sailer-Gaulin models. J Quant Anthropol 1:229–238
Freeman LC, Webster CM, Kirke DM (1998) Exploring social structure using dynamic three-dimensional color images. Soc Netw 20(2):109–118
Geiser P, Danon L (2003) Community structure in jazz. Adv Complex Syst 6(4):563–573
Gemmetto V, Barrat A, Cattuto C (2014) Mitigation of infectious disease at school: targeted class closure vs. school closure. BMC Infect Dis 14(695):1–10
Gephi (2020) https://gephi.org/. Last accessed: 25 Dec 2020
Ghalmane Z, El Hassouni M, Cherifi H (2019a) Immunization of networks with non-overlapping community structure. Soc Netw Anal Min 9(1):45
Ghalmane Z, El Hassouni M, Cherifi C, Cherifi H (2019b) Centrality in modular networks. EPJ Data Sci 8(1):1–27

Ghalmane Z, Cherifi C, Cherifi H, El Hassouni M (2019c) Centrality in complex networks with overlapping community structure. Sci Rep 9(10133):1–29
Gil-Mendieta J, Schmidt S (1996) The political network in Mexico. Soc Netw 18(4):355–381
Girvan M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826
Godsil C, Royle GF (2001) Algebraic graph theory, 1st edn. Springer, New York, p 2001
Grimmer J (2010) A Bayesian hierarchical topic mode for political texts: measuring expressed agendas in senate press releases. Polit Anal 18(1):1–35
Gross JL, Yellen J, Zhang P (eds) (2013) Handbook of graph theory, 2nd edn. CRC Press, Boca Raton
Guimera R, Danon L, Diaz-Guilera A, Giralt F, Arenas A (2003) Self-similar community structure in a network of human interactions. Phys Rev E 68:065103
Gupta N, Singh A, Cherifi H (2016) Centrality measures for networks with community structure. Phys A 452:46–59
Hayes B (2006) Connecting the dots. Am Sci 94(5):400–404
He D, Xu J, Chen X (2016) Information-theoretic-entropy based weight aggregation method in multiple-attribute group decision-making. Entropy 18(6):171
Hoffman M, Steinley D, Gates KM, Prinstein MJ, Brusco MJ (2018) Detecting clusters/communities in social networks. Multivar Behav Res 53(1):57–73
Hummon NP, Doreian P, Freeman LC (1990) Analyzing the structure of the centrality-productivity literature created between 1948 and 1979. Sci Commun 11(4):459–480. https://doi.org/10.1177/107554709001100405
Hwang W-C, Zhang A, Ramanathan M (2008) Identification of information flow-modulating drug targets: a novel bridging paradigm for drug discovery. Clin Pharmacol Ther 84(5):563–572
Ibnoulouafi A, El Haziti M, Cherifi H (2018) M-centrality: identifying key nodes based on global position and local degree variation. J Stat Mech: Theory Exp 2018(7):073407
Isella L, Stehle J, Barrat A, Cattuto C, Pinton JF, Van den Broeck W (2011) What's in a crowd? Analysis of face-to-face behavioral networks. J Theor Biol 271(1):166–180
Johnson DS (1984) The genealogy of theoretical computer science: a preliminary report. ACM SIGACT News 16(2):36–44
Kleinberg J, Suri S, Tardos E, Wexler T (2008) Strategic network formation with structural holes. In: Proceedings of the 9th ACM conference on electronic commerce, pp 284–293, Chicago, USA, July 2008
Knuth DE (1993) The Stanford GraphBase: a platform for combinatorial computing, 1st edn. Addison-Wesley, Reading, MA
Krackhardt D (1999) The ties that torture: Simmelian tie analysis in organizations. Res Sociol Organ 16:183–210
Krebs V (2003) Proxy networks: analyzing one network to reveal another. Bull Méthodol Sociol 79:61–40
Kumar M, Singh A, Cherifi H (2018) An efficient immunization strategy using overlapping nodes and its neighborhoods. In: Proceedings of the web conference WWW'18, pp 1269–1275, San Francisco, CA, USA, April 2018
Lancichinetti A, Fortunato S (2009) Community detection algorithms: a comparative analysis. Phys Rev E 80:056117
Lazega E (2001) The collegial phenomenon: the social mechanisms of cooperation among peers in a corporate law partnership, Illustrated. Oxford University Press, Oxford
Liu Z, Hu B (2005) Epidemic spreading in community networks. Europhys Lett 72(2):315–321
Liu W, Pellegrini M, Wu A (2019) Identification of bridging centrality in complex networks. IEEE Access 7:93123–93130
Lohninger H (2021) Fundamentals of statistics. http://www.statistics4u.com/fundstat_eng/, Oct 2012. Last accessed: 10 April 2021
Loomis CP, Morales JO, Clifford RA, Leonard OE (1953) Turrialba social systems and the introduction of change. The Free Press, Glencoe, IL, pp 45–78
Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(3):396–405
MacRae D (1960) Direct factor analysis of sociometric data. Sociometry 23(4):360–371
Madotto A, Liu J (2016) Super-spreader identification using meta-centrality. Sci Rep 6:1–10 (**article number: 38994**)
Magelinski T, Bartulovic M, Carley KM (2021) Measuring node contribution to community structure with modularity vitality. IEEE Trans Netw Sci Eng 8(1):707–723
Magnani M, Micenkova B, Rossi L (2013) Combinatorial analysis of multiple networks. arXiv:1303.4986 [cs.SI]
Mahadevan P, Krioukov D, Fomenkov M, Dimitropoulos X, Claffy KC, Vahdat A (2006) The internet AS-level topology: three data sources and one definitive metric. ACM SIGCOMM Comput Commun Rev 36(1):17–26
Mareno JL (1960) The sociometry reader. The Free Press, Glencoe, IL, pp 534–547
Masuda N (2009) Immunization of networks with community structure. New J Phys 11(12):123018
Meghanathan N (2014) Spectral radius as a measure of variation in node degree for complex network graphs. In: Proceedings of the 3rd international conference on digital contents and applications, (DCA 2014), pp 30–33, Hainan, China, 20–23 Dec 2014
Meghanathan N (2016) Assortativity analysis of real-world network graphs based on centrality metrics. Comput Inf Sci 9(3):7–25
Meghanathan N (2017a) Complex network analysis of the contiguous United States graph. Comput Inf Sci 10(1):54–76
Meghanathan N (2017b) A computationally-lightweight and localized centrality metric in Lieu of betweenness centrality for complex network analysis. Vietnam J Comput Sci 4(1):23–38
Meghanathan N (2017c) Evaluation of correlation measures for computationally-light vs. computationally-heavy centrality metrics on real-world graphs. J Comput Inf Technol 25(2):103–132
Michael JH (1997) Labor dispute reconciliation in a forest products manufacturing facility. For Prod J 47(11–12):41–45
Nepusz T, Petroczi A, Negyessy L, Bazso F (2008) Fuzzy communities and the concept of bridgeness in complex networks. Phys Rev E 77(1):016107
Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104
Newman MEJ (2010) Networks: an introduction, 1st edn. Oxford University Press, Oxford
Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69:026113

Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: analysis and an algorithm. In: Proceedings of the 14th international conference on neural information processing systems, pp 849–856, Vancouver, Canada, January 2001
Oldham S, Fulcher B, Parkes L, Arnatkeviciute A, Suo C, Fornito A (2019) Consistency and differences between centrality measures across distinct classes of networks. PLoS ONE 14(7):e0220061
Peel L, Larremore DB, Clauset A (2017) The ground truth about metadata and community detection in networks. Sci Adv 3(5):e1602548
Rogers EM, Kincaid DL (1980) Communication networks: toward a new paradigm for research. Free Press, Glencoe, IL
Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the 29th AAAI conference on artificial intelligence, pp 4292–4293, Austin, TX, USA, January 2015
Rozemberczki B, Sarkar R (2020) Characteristic functions on graphs: birds of a feather, from statistical descriptors to parametric models. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 1325–1334
Rozemberczki B, Davies R, Sarkar R, Sutton C (2019) GEMSEC: graph embedding with self clustering. In: *Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining*, pp. 65–72, Vancouver, Canada, August 2019
Scannell JW, Blakemore C, Young MP (1995) Analysis of connectivity in the cat cerebral cortex. J Neurosci 15(2):1463–1483
Schwimmer E (1973) Exchange in the social structure of the Orokaiva: traditional and emergent ideologies in the Northern District of Papua. C Hurst and Co-Publishers Ltd., London
Sciarra C, Chiarotti G, Laio F, Ridolfi L (2018) A change of perspective in network centrality. Sci Rep 8:1–9 (**article number: 15269**)
Singh R, Xu J, Berger B (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. Proc Natl Acad Sci USA 105(35):12763–12768
Smith DA, White DR (1992) Structure and dynamics of the global economy: network analysis of international trade 1965–1980. Soc Forces 70(4):857–893
Strang GG (2006) Linear algebra and its applications, 4th edn. Brooks Cole, Pacific Grove, CA
Takahata Y (1991) Diachronic changes in the dominance relations of adult female Japanese monkeys of the Arashiyama B Group. In: Fedigan LM, Asquith PJ (eds) The monkeys of Arashiyama. State University of New York Press, Albany, pp 124–139
Traag VA, Waltman L, van Eck NJ (2019) From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep 9(1):1–12
Tulu MM, Hou R, Younas T (2018) Identifying influential nodes based on community structure to speed up the dissemination of information in complex network. IEEE Access 6:7390–7401
Ulanowicz R, Donald D (2005) Network analysis of trophic dynamics in South Florida ecosystems. In: US Geological Survey Program on the South Florida Ecosystem, pp 114–115
Van Mieghem P (2010) Graph spectra for complex networks, 1st edn. Cambridge University Press, Cambridge
Vega-Oliveros DA, Gomes PS, Milios EE, Berton L (2019) A multi-centrality index for graph-based keyword extraction. Inf Process Manag 56(6):102063
Wang Z, Zhao Y, Xi J, Du C (2016) Fast ranking influential nodes in complex networks using a K-shell iteration factor. Phys A: Stat Mech Appl 461:171–181
Watts DJ, Strogatz SH (1998) Collective dynamics of "small-world" networks. Nature 393:440–442
White JG, Southgate E, Thomson JN, Brenner S (1986) The structure of the nervous system of the nematode *Caenorhabditis elegans*. Philos Trans B 314(1165):1–340
Yang H, An S (2020) Critical nodes identification in complex networks. Symmetry 12(1):123
Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473
Zander CD et al (2011) Food web including metazoan parasites for a brackish shallow water ecosystem in Germany and Denmark. Ecology 92(10):2007

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.