Structural Vulnerability Assessment of Community-Based Routing in Opportunistic Networks

Md Abdul Alim, Xiang Li, Nam P. Nguyen, My T. Thai, *Member, IEEE*, and Abdelsalam Helal, *Fellow, IEEE*

Abstract—Opportunistic networks enable mobile devices to communicate with each other through routes that are built dynamically, while messages are en route between the sender and the destination(s). The social structure and interaction of users of such devices dictate the performance of routing protocols in those networks. Community structures, commonly exhibited by social networks, is also observed in the encounter patterns in opportunistic networks and has an astounding impact in designing forwarding algorithms for such types of networks. In this paper, we explore the structural vulnerability of social-based forwarding and routing methods in opportunistic networks. In particular, we introduce Community Vulnerability Assessment (CVA), a new problem on assessing the performance reliability of opportunistic routing strategies in Delay Tolerant Networks (DTN) from a *community structure* point of view. Given a positive number k, CVA aims to find out the k most vulnerable devices in the network whose non-participation (due to out-of-service or permanent out-of-range) transforms the current network community structure to a totally different one. As the first study in this direction, we analyze and provide key insights into the separation of network communities, evaluated via the Normalized Mutual Information (NMI). Based on these findings, we suggest an approximation algorithm for the special case when $k = 1$, and a heuristic, genEdge, for the general case. To certify the effectiveness of our proposed approaches, we first test them on synthesized data with known community structures, and then we show the impact of node removal on community structures in real social networks. Finally we evaluate the performance via different forwarding and routing strategies in multiple real-world DTN traces. Our results indicate that, in many forwarding and routing methods, the nonparticipation of only some important devices is significant enough to degrade the entire network's performance.

Index Terms—Community structure, vulnerability assessment, routing and forwarding, delay tolerant network

1 Introduction

Mobile opportunistic networks are characterized by intermittent and non-deterministic connectivity, often due to interruptible wireless links, sparse network deployment and/or nodal mobility. Such opportunistic networking has been discussed in the context of delay/disruption-tolerant networks, sporadically connected sensor networks, vehicular networks, peer-to-peer mobile social networks, and 5G networks. These networks do not depend on any infrastructure but, instead, exploit opportunistic connections between mobile devices to enable device-to-device communication. In this paper, we focus on disruption or delay tolerant networks (DTN) which have recently drawn a great attention due to their wide application in pervasive environments such as military operations, space communication and dynamic wireless sensor deployments [1]. In general, DTNs are partitioned wireless ad-hoc networks with the notable characteristic of intermittent connectivity [1]. Due to this intermittent connectivity, DTNs display unstable network structures, are lack of instantaneous end-to-end connections and thus, shall never be fully connected at any point in time. In addition, they often incur a large transmission delay between participating devices together with a probability of unsuccessful transmission. These characteristics limit the use of traditional message forwarding protocols [2] since they rely on the establishment of a complete end-to-end route from the source to the destination.

Many forwarding and routing methods have been proposed for DTNs in the literature (see [1] and references therein). Nowadays, since wireless devices (such as smart phones, PDAs, etc) are usually carried by people, and because people have a tendency to move and communicate in groups, social-based forwarding and routing strategies exploring social interactions and centrality have emerged as potential solutions for this matter [1]. Particularly, community-aware approaches such as Label [3], Bubble-Rap [4], Social-based multicasting [5], and Friendship-based routing [6] have been shown to be very efficient and are among the best methods for DTNs. Here, a group or community in DTNs can be visualized as a group of frequently interacted wireless devices with less connectivity to other groups. Devices in the same community have higher chances to encounter each other to transfer carried messages. Therefore, the knowledge of the community structure could help the routing protocols to wisely choose better forwarding relays which can bring the message closer to the final destination,

- *M.A. Alim, X. Li, M.T. Thai, and A. Helal are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: {alim, xixiang, mythai}@cise.ufl.edu, npnguyen@towson.edunpnguyen@towson.edu.*
- *N.P. Nguyen is with the Department of Computer & Information Sciences, Towson University, Towson, MD 21252. E-mail: helal@cise.ufl.edu.*

Manuscript received 22 May 2015; revised 17 Jan. 2016; accepted 28 Jan. 2016. Date of publication 3 Feb. 2016; date of current version 31 Oct. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2524571

and hence, could significantly improve the chances of message delivery. Community-based forwarding and routing techniques in DTNs, as a result, rely on this knowledge as the heart of their decision-making process [1].

Although a lot of works have shown the effectiveness of community-aware forwarding schemes in DTNs, none of them really explored the structural vulnerability of these routing methods. In particular, the impact analysis of the stability of community structures on the performance measures of DTN routing in terms of different metrics have so far been an untrodden area. During the network operation, it is possible that some important wireless devices can potentially be out of service due to energy exhaustion, or they can be permanently out of communication range due to the intermittent connectivity in DTNs. The nonparticipation (or failure in service) of these important devices can incur a major structural change in the network topology, and consequently, can lead to a significant transformation of the network community structure. This undesirable reaction challenges the performance of community-based forwarding methods because forwarding routes now may not be feasible (if the missing devices were bridge nodes), or may take too many more steps for the message to be delivered (if the destination's community is loosely connected from the source). Those consequences shall degrade the message delivery rate and introduce an even longer transmission delay in the resulting DTN. Therefore, identifying key devices that are crucial to the network community structure is an extremely important task in maintaining the desired performance of community-aware routing strategies in DTNs as well as development of more secure and reliable forwarding and routing techniques. To the best of our knowledge, this vulnerability on community structure has not been widely addressed in the literature.

Exploring the structural vulnerability of network community structure has considerable significance. For instance, in DTNs the awareness of this vulnerability can help in designing a forwarding algorithm that does not overload those crucial devices by overflowing the limited queue capacity (if they happen to be the highly ranked ones in a community), or in designing an effective backup plan when some of them may fail at the same time. Not only in opportunistic networks does this vulnerability assessment come into use but also in other real world applications this has considerable impact. For instance, in worm containment application in online social networks (OSNs), this knowledge can provide helpful insights into the protection of those sensitive nodes (if they are indeed high influential users) once worms spread out in the network. However, under a minor structural change when a node is excluded from a community, this particular community can either stay intact if the removed node is less important, or can be broken down into smaller subcommunities which can further be merged to other communities if the current node is of great importance to the community. This unpredictable transformation of network communities makes their structural vulnerability assessment an extremely complicated yet challenging problem.

As a first study on this research direction, in this paper, we take the initial step on understanding how the failures of crucial devices in DTNs affect the structure of its communities. As mobile devices together with their connections and interactions can be expressed under graph theory using nodes/vertices and edges, we have utilized the graph theoretic approach to identify the critical nodes in the underlying graph removal of which causes maximal change of the network communities. Specifically, given the input network, the community detection algorithm $\mathcal{A}$ and a positive number k, we formulate the *Community Vulnerability Assessment* (CVA) problem which aims to find a set S of k nodes whose removal maximally transforms the current network community structure to a totally different one, evaluated via the Normalized Mutual Information (NMI) measure [7]. Our empirical results indicate that, in many forwarding and routing methods, the failure of only a small number of important devices is significant enough to degrade the performance of the entire network. The main contributions of this paper pertain to answering the following questions:

- How vulnerable are the social-based forwarding strategies in DTNs?
- How the performance of social-based routing and forwarding schemes are impacted if the community structure is changed?
- What is the role of important (hub and bridge nodes) devices in community-based routing schemes?

The rest of this paper is structured as follows. We formulate the problem for assessing vulnerability of social-based forwarding and routing strategies in DTNs from a community structure point of view in Section 2. In Section 3 we analyze potential conditions that can possibly lead to the minimization of NMI on community structures. We suggest an approximation algorithm for the case $k = 1$, and propose genEdge, a heuristic for CVA problem when $k > 1$ in Section 4. We conduct experiments on both synthesized data with known community structures along with real world social network data, and finally on different forwarding and routing strategies in multiple real-world DTN traces in Section 5 to show the structural vulnerability of these schemes. Section 6 discusses literature review on vulnerability assessment of DTN routing and finally Section 7 concludes the paper.

2 Problem Formulation

Graph notations. Let $G = (V, E)$ be an undirected unweighted graph representing the input network. Necessary terms related to graph structure are described in Table 1.

Community structure. Denote by $\mathcal{A}$ the community detection algorithm on G, by $X = \{X_1, X_2, \ldots, X_{c_X}\}$ and $Y = \{Y_1, Y_2, \ldots, Y_{c_Y}\}$ the two community structures of c_X and c_Y communities detected by $\mathcal{A}$ *before* and *after* the removal of a set S of k nodes in G, respectively. Mathematically, X and Y are represented as $X = \mathcal{A}(G)$ and $Y = \mathcal{A}(G[V \backslash S])$, where $G[V \backslash S]$ is the subgraph induced by G on $V \backslash S$. For any $i = 1, \ldots, c_X$ and $j = 1, \ldots, c_Y$, let $x_i = |X_i|$, $y_j = |Y_j|$, $n_{ij} = |X_i \cap Y_j|$ and l_i be the number of removed (lost) nodes in X_i, respectively. Finally, let $\bar{x} = \sum_{i=1}^{c_X} x_i$, $\bar{y} = \sum_{j=1}^{c_Y} y_j$, and $\bar{n} = \sum_{i=1}^{c_X} \sum_{j=1}^{c_Y} n_{ij}$ in this order be the total size of communities in X, Y, and the total number of common nodes shared between X and Y.

NMI. In order to evaluate how much the network community structure changes after the removal of important

TABLE 1
Summary of Notations

Symbol	Description
G	Input network
V	Set of nodes (mobile devices)
E	Set of edges (relationship between devices)
n	Number of nodes
m	Number of edges
C	A single community
n_C	Number of nodes in C
m_C	Number of internal edges in C
d_C	Maximum degree of a node in C
$N(u)$	Set of all neighbors of node u
d_u	Degree of node u in G
d_u^C	Degree of node u in C
X	Set of communities before node removal
c_X	Number of communities in X
Y	Set of communities after node removal
c_Y	Number of communities in Y
x_i	Number of nodes in community X_i
y_j	Number of nodes in community Y_j
n_{ij}	Number of common nodes between X_i and Y_j
l_i	Number of removed nodes from community X_i
$\bar{x}$	Total size of communities in X
$\bar{y}$	Total size of communities in Y
$\bar{n}$	Total common nodes between X and Y

nodes, we use Normalized Mutual Information [7]. Basically, given two community structures A and B, $NMI(A,B)$ is 1 if A and B are identical and is 0 if they are totally separated, and the higher $NMI(A,B)$ score the more similar A and B are believed to be. Thus, it is a well-suited metric dedicated for certifying the quality of detected community structures, and the effectiveness of this widely-accepted measure has also been extensively verified in the literature [8]. More details about NMI will be elaborated in our analysis in Section 3. Finally, the Community structure Vulnerability Assessment (CVA) problem is formulated as follows:

Definition 1 (CVA). *Given a mobile network represented by a graph $G=(V,E)$, a specific community detection algorithm $\mathcal{A}$, and a positive integer $k \leq n$, we seek for a subset $S \subseteq V$ such that*

$$S = \underset{S' \subseteq V, |S'|=k}{\operatorname{argmin}} \{NMI_X(S')\},$$

where $NMI_X(S') = NMI(X, \mathcal{A}(G[V \backslash S']))$ for any subset $S' \subseteq V$.

Specifically, CVA problem seeks for a subset $S \subseteq V$ of k nodes whose removal results in the maximum difference between the original community structure X and the new community structure Y detected by $\mathcal{A}$ on $G[V \backslash S]$. We call S the *node vulnerability set* of G since its removal maximally transforms network communities of G to different structures.

The above formulation of CVA problem requires the community detection algorithm $\mathcal{A}$ as an input parameter. Because there is not yet a universal agreement or accepted definition of a network community, this input is necessary in the sense that different algorithms with different objective functions might favor different sets of nodes, and thus, a good solution set for one community detection algorithm may not be good for the others. Nevertheless, a node selection strategy that relies more on the input network and less on the community detection algorithm is always of desire.

3 Analysis of NMI Measure

In this section, we first derive some important properties of NMI measure, and then investigate possible conditions on sizes and the number of communities that potentially lead to the minimization of $NMI(X,Y)$ in both disjoint and overlapped community structures. We stress that these conditions are by no means universal or exhaustive since some of them might not hold true simultaneously. Indeed, what we hope for is these conditions would provide us key insights into the selection of important nodes to maximally separate X and Y.

3.1 Formulation

To evaluate $NMI(X,Y)$ with $X=\{X_1, X_2, \ldots, X_{c_X}\}$ and $Y=\{Y_1, Y_2, \ldots, Y_{c_Y}\}$, we consider community labels X_i and Y_j, where X_i and Y_j indicate the community labels of a node t in X and Y, respectively. We can also assume that the labels X_i and Y_j are values of two random "variables" X and Y [7] (here we reuse notations X and Y to denote the random variables), with joint distributions $P(X_i, Y_j) = P(X = X_i; Y = Y_i) = n_{ij}/(n-k)$, and individual distribution $P(X_i) = x_i/n; P(Y_j) = y_j/(n-k)$. Using these notations, the entropies (or uncertainties) $H(X)$ and $H(Y)$ of X and Y [9] are formulated as: $H(X) = -\sum_{i=1}^{c_X} P(X_i) \log P(X_i) = -\sum_{i=1}^{c_X} \frac{x_i}{n} \log \frac{x_i}{n}$, and similarly, $H(Y) = -\sum_{j=1}^{c_Y} P(Y_j) \log P(Y_j) = \frac{1}{n-k}\left(\bar{y} \log (n-k) - \sum_{j=1}^{c_Y} y_j \log y_j\right)$. Note that in CVA problem, X can be found based on $\mathcal{A}$ and G, and as a result, x_i's and $H(X)$ can also be inferred from these input parameters. Therefore, we consider them as constants in this paper.

The Mutual Information $I(X,Y)$ [9] of X and Y is: $I(X,Y) = \sum_{i=1}^{c_X} \sum_{j=1}^{c_Y} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(X_i)P(Y_j)}$. $I(X,Y)$ is symmetric and tells us how much we know about variable (or structure) Y if we already know about variable X, and vice versa. However, as indicated in [7], [8], Mutual Information itself is not ideal as a global similarity metric since any subpartition of a given community structure X would result in the same mutual information with X, even though they can possibly be very different from each other. The authors in [7] introduce NMI which can overcome that limitation. Formally, NMI of two random variables X and Y is defined as $NMI(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)}$. Particularly, $NMI(X, Y)$ is written as

$$\frac{2\sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} n_{ij} \log \frac{n_{ij} n}{x_i y_j}}{(n-k)H(X) + \bar{y} \log (n-k) - \sum_{j=1}^{c_Y} y_j \log y_j}. \quad (1)$$

3.2 Properties

We derive some important properties of NMI measure in terms of the set of excluded nodes. Intuitively, one expects that the larger the set L of excluded nodes the lower the $NMI_X(L)$ score would be. However, we show that this is not the case in general, i.e., there exists a specific network in which the removal of more nodes results in a higher $NMI_X()$

score. As a consequence, $NMI_X()$ function is not submodular in terms of the set of excluded nodes. Note that hereafter we consider equation (1) without the constant factor 2.

Lemma 1. *There is a graph $G = (V, E)$ in which there are subsets $L \subseteq T \subseteq V$ such that $NMI_X(T) \geq NMI_X(L)$ (here, L and T are sets of excluded nodes).*

Theorem 1 (below) generalizes Lemma 1 and realizes the nonsubmodularity of $NMI_X()$ in terms of community assignments. Here, given two community assignments A and B, we write $A \subseteq B$ if every community defined by A is a subcommunity defined by B. The proofs for the following theorem along with the above lemma are presented in Appendix A , which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2016.2524571.

Theorem 1. *Given community assignments $A \subseteq B$, there is $s \notin A, B$ such that $NMI_X(A + s) - NMI_X(A) < NMI_X(B + s) - NMI_X(B)$. This implies that $NMI_X()$ is not a submodular function.*

3.3 Minimizing $NMI(X, Y)$ in a Nonoverlapped Community Structure

In a disjoint community structure, we have $X_i \cap X_{i'} = \emptyset$, $\cup_{i=1}^{c_X} X_i = V$, $Y_j \cap Y_{j'} = \emptyset$, and $\cup_{j=1}^{c_Y} Y_j = V \backslash S$ for $i, i' = 1, \ldots, c_X$ and $j, j' = 1, \ldots, c_Y$. Therefore: $\bar{x} = n$, $\bar{y} = (n-k)$, $\sum_{i=1}^{c_X} n_{ij} = y_j$, $\sum_{j=1}^{c_Y} n_{ij} = x_i - l_i$, and $\bar{n} = \sum_{ij} n_{ij} = (n-k)$. (2)

3.3.1 Minimizing NMI within a Community

We first investigate the behavior of $NMI(X, Y)$ in a special case where only one specific community of X is affected by the removal of set S of k nodes while other communities stay intact. We can assume that X_1 is the targeted community which is further divided into p smaller subcommunities of sizes $s_1, s_2, \ldots, s_p$ satisfying $\sum_{j=1}^{p} s_j = x_1$. In this case, $H(Y) = H(X) + x_1\left(\frac{\log(n-k)}{n-k} - \frac{\log \frac{n}{x_1}}{n}\right) - \sum_{j=1}^{p} s_j \log s_j$, and $I(X, Y) = \sum_{j=1}^{c_Y} \frac{x_j}{n-k} \log \frac{n}{x_j} = \frac{n-k}{n} H(X)$. Thus, $NMI(X, Y)$ is minimized when $\sum_{j=1}^{p} s_j \log s_j$ is minimized. Since function $s \log s$ is strictly convex for any $s > 0$, we can apply Jensen's inequality [9] and get $\frac{1}{p}\sum_{j=1}^{p} s_j \log s_j \geq \frac{\sum_{j=1}^{p} s_j}{p} \log \frac{\sum_{j=1}^{p} s_j}{p} = \frac{x_1}{p} \log \frac{x_1}{p}$, where equality holds when all s_j's are equal to each other.

It reveals from this inequality that, in order to further minimize the RHS quantity, one can try to break X_1 into *as many smaller communities of the relatively same size as possible* (i.e., to enlarge p as much as possible while ensuring s_i's are all equal). This observation makes sense since a new structure of X_1 with all singleton communities will incur $\sum_{j=1}^{p} s_j \log s_j = 0$, and hence, will maximize $H(Y)$ and in turn will minimize $NMI(X, Y)$. However, since the new structure of X_1 depends on the community detection algorithm $\mathcal{A}$, the all-singleton communities scenario might not always be the case. Will this crucial observation hold true in a general disjoint community structure? We tend to lean over the affirmative answer through the analysis in the following section.

3.3.2 Minimizing NMI in a Disjoint Structure

In general, the equalities in (2) simplify equation (1) to

$$\frac{\sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} n_{ij} \log \frac{n_{ij} n}{x_i y_j}}{(n-k)H(X) + (n-k)\log(n-k) - \sum_{j=1}^{c_Y} y_j \log y_j}.$$

In order to minimize the above ratio, one would seek for the conditions in which the numerator of $NMI(X, Y)$ is minimized while its denominator is also maximized. To maximize the latter quantity, we need to minimize $\sum_{j=1}^{c_Y} y_j \log y_j$. Now, applying Jensen's inequality gives $\frac{1}{c_Y}\sum_{j=1}^{c_Y} y_j \log y_j \geq \frac{\bar{y}}{c_Y} \log \frac{\bar{y}}{c_Y} = \frac{n-k}{c_Y} \log \frac{n-k}{c_Y}$, and thus $\sum_{j=1}^{c_Y} y_j \log y_j$ can attain its minimum at $(n-k)\log \frac{n-k}{c_Y}$ where equality holds when all y_j's are equal to each other.

As n and k are input parameters, $\log \frac{n-k}{c_Y}$ can further be minimized when c_Y is as large as possible, while requiring y_j's to be equal to each other. Mathematically, this can be achieved when Y contains exactly $c_Y = (n-k)$ singleton communities. However, since our problem is community detection algorithm dependent, this inequality advises that, in order to minimize $NMI(X, Y)$ measure, the new community structure Y should contain as many communities of relatively the same size as possible.

To minimize the numerator of $NMI(X, Y)$, we write $I(X, Y) = \frac{1}{n-k}(\sum_{ij} n_{ij} \log \frac{n_{ij} n}{y_j} - \sum_{ij} n_{ij} \log x_i)$. Next, applying Log Sum Theorem [9] to the first summand, we get $I(X, Y) \geq \frac{1}{n-k}(\bar{n} \log \frac{n \cdot \bar{n}}{c_X \bar{y}} - \sum_{ij} n_{ij} \log x_i) = \log \frac{n}{c_X} - \frac{1}{n-k}\sum_i (x_i - l_i) \log x_i$, since $\bar{n} = \bar{y} = n - k$ and $\sum_{j=1}^{c_Y} n_{ij} = x_i - l_i$, $\forall i = 1, \ldots, c_X$, where l_i is the number of deleted (or lost) nodes in community X_i, and l_i's satisfy $\sum_{i=1}^{c_X} l_i = k$. The equality holds when n_{ij}/y_j is a constant, say $\gamma \geq 0$, for all $i = 1, \ldots, c_X, j = 1, \ldots, c_Y$. If we assume that this is the case, then $\sum_{j=1}^{c_Y} n_{ij} = \gamma \sum_{j=1}^{c_Y} y_j = \gamma(n-k)$, which in turn implies $n - k = \sum_{ij} n_{ij} = \gamma c_X (n-k)$. Hence, $\gamma = 1/c_X$ and thus, $l_i = x_i - (n-k)/c_X$.

Therefore, to minimize the second summand, the equation $l_i = x_i - (n-k)/c_X$ advises that we should put more focus on (i.e., remove more nodes in) big-sized communities X_i of X to break it into smaller modules. This breaking down of big-sized communities partially supports the prior observation that communities of Y should have relatively the same size. Note that in this analysis, we have assumed that n_{ij}/y_j is a constant for all pair of i and j. In practice, this might not always be the case since real communities can be distributed differently based on the underlying detection algorithm. Nevertheless, we find this observation helpful as it suggests a general direction for selecting important nodes in the network.

3.4 Minimizing $NMI(X, Y)$ in an Overlapped Community Structure

The minimization of $NMI(X, Y)$ measure is much more complicated when network communities can overlap with each other. In particular, the conditions $\cup_{i=1}^{c_X} X_i = V$ and $\cup_{j=1}^{c_Y} Y_j = V \backslash S$ still hold in this case; however, $X_i \cap X_{i'}$ and $Y_j \cap Y_{j'}$ might not be empty for some $i, i' = 1, \ldots, c_X$ and $j, j' = 1, \ldots, c_Y$. These facts indicate that $\bar{x} = \sum_{i=1}^{c_X} x_i \geq n$, $\bar{y} = \sum_{j=1}^{c_Y} y_j \geq n - k$ and $\bar{n} = \sum_{ij} n_{ij} \geq n - k$.

Our analysis strategy in this case is similar to the prior one as we also strive for maximizing the denominator while minimizing the numerator of $NMI(X,Y)$ (eq. (1)). Because $\bar{n} \geq n-k$, the minimization of the top term $I(X,Y)$ no longer depends only on x_i's. One way to work around this issue is to investigate the relative correlation between the total community size $\bar{y}$ and the number of communities c_Y. Let $\alpha_{\mathcal{A}} = \frac{\bar{y}}{c_Y}$ be the ratio between these two quantities, or in other words, the averaged community size. Using this notation, the denominator of $NMI(X,Y)$ is evaluated as: $\bar{y}\log(n-k) - \sum_{j=1}^{c_Y} y_j \log y_j \leq \bar{y}(\log(n-k) - \frac{\log(\bar{y}/c_Y)}{c_Y}) = \bar{y}\log\ (n-k) - \alpha_{\mathcal{A}}\log\alpha_{\mathcal{A}}$, with equality holding when all y_j's are equal to each other. To further maximize this denominator, we need $\bar{y}$ to be as large as possible while keeping $\alpha_{\mathcal{A}}$ as small as possible, i.e., the new community structure Y should contain more and more communities so as to increase c_Y as well as to lower down $\alpha_{\mathcal{A}}$.

Due to the dependence on the specific detection algorithm $\mathcal{A}$, this optimization on the correlation between $\bar{y}$ and c_Y might not be globally achieved. However, a coarse analysis between $\bar{y}$ and c_Y can relatively be conducted in the following sense: if we assume that $\bar{y}$ is within a constant factor of the total number of actual nodes $(n-k)$, i.e., $\bar{y} \leq a_0(n-k)$ for some constant $a_0 > 1$, we can then increase the value of the RHS by breaking as many communities as possible while retaining similar size (i.e., enlarge c_Y and keep all y_j's the same), which helps to reduce the impact of $\alpha_{\mathcal{A}}\log\alpha_{\mathcal{A}}$. This observation, though relative, agrees with what we achieved in the case of disjoint community structure. In an unfortunate case where $\bar{y}$ is not known to be within any constant factor of $(n-k)$, the observation might not hold since both $\bar{y}$ and c_Y can be arbitrary large and thus, $\alpha_{\mathcal{A}}\log\alpha_{\mathcal{A}}$ could still be relatively small.

Next, applying Log Sum Theorem on the numerator yields $I(X,Y) = \sum_{ij} n_{ij}\log\frac{n_{ij}n}{x_i y_j} \geq \bar{n}\log\frac{n\cdot\bar{n}}{\bar{x}\bar{y}}$,

with equality holding when $\frac{n_{ij}n}{x_i y_j}$ is a constant for all $i = 1,\ldots,c_X$ and $j = 1,\ldots,c_Y$. Thus, one can try to minimize $I(X,Y)$ by deleting nodes in such a way that $\bar{n}$ is maximized and $\bar{y}$ is minimized while making sure that $\frac{n_{ij}n}{x_i y_j}$ is a constant. As a result, this minimization of $I(X,Y)$ is a multiple-objective optimization problem which may not have a feasible solution. However, if we assume that the later condition is imposed, i.e., $\frac{n_{ij}n}{x_i y_j} = \beta_{\mathcal{A}}$ for some constant $\beta_{\mathcal{A}} > 0$, then $n_{ij} = \frac{\beta_{\mathcal{A}} x_i y_j}{n}$, and thus $\bar{n} = \frac{\beta_{\mathcal{A}}}{n}\bar{x}\bar{y}$. This reduces the above inequality to $I(X,Y) \geq \frac{\bar{x}}{n}\beta_{\mathcal{A}}\bar{y}\log\beta_{\mathcal{A}}n$. The RHS of the inequality advises that, in order to minimize $I(X,Y)$, *the total size of network communities should not be too large while the overlapping ratio of every community should be equal to each other and be as small as possible*. This is a different criterion from the disjoint community structure's point of view.

4 Solutions to CVA

In the following paragraphs, we consider the scenario where maximizing the internal density [10] is the objective function for finding network communities, i.e., communities of G are assumed to have optimized internal densities. In this connection, we first prove the computational complexity of the CVA problem by showing its NP-completeness. Then, we present an approximation algorithm in the special case $k = 1$, and subsequently propose genEdge, a heuristic algorithm for CVA problem that is independent of the underlying community detection algorithm $\mathcal{A}$. Our target strategy will try to break larger communities to as many small ones as possible while looking for those to have the relatively same size with small overlapping ratios.

4.1 NP-Completeness of CVA

We show the NP-completeness of the CVA problem by reducing the decision version of it from the well-known maximum vertex coverage problem [11].

The decision version of CVA:

Definition 2 (CVAD). *Given a mobile network represented by a graph $G = (V,E)$, a specific community detection algorithm $\mathcal{A}$, and a positive integer $k \leq n$, CVAD asks that whether there exists a subset $S \subseteq V, |S| = k$ such that $NMI_X(S) \leq a$, where $NMI_X(S) = NMI(X, \mathcal{A}(G[V\backslash S]))$.*

As both the community detection algorithm $\mathcal{A}$ and the calculation of $NMI_X(S)$ takes polynomial time with a given $S \subseteq V$, it is clear that a solution $S \subset V$ of CVAD can be verified in polynomial time. Thus, CVAD is in NP.

To prove the NP hardness of CVAD, we reduce it from the maximum vertex coverage problem on bipartite graphs (MVC-B), which is proved to be a NP-complete [11] problem. The decision version of this problem is as follows.

Definition 3 (MVC-B). *Given a bipartite graph $G = (V,E)$, $V = V_L \cup V_R, V_L \cap V_R = \emptyset$ and positive integers b, c, the vertex cover problem asks if there exists a subset of vertices $S \subseteq V$ with size b that at least c edges are incident to nodes in S.*

Proof. Given an instance of MVC-B with bipartite graph $G = (V,E), V = V_L \cup V_R, V_L \cap V_R = \emptyset$ and positive integers b, c. Now we construct the CVA instance $G' = (V', E')$ by connecting V_L, V_R to cliques K_L, K_R, respectively. We create an edge between all $\{(u,v)\ |u \in K_L, v \in V_L\}$ and all $\{(u,v)|u \in K_R, v \in V_R\}$. We choose the size of K_L, K_R in a way that when at least c edges are removed from E, $\mathcal{A}$ will detect two communities $K_L \cup V_L, K_R \cup V_R$ (and exclude the removed vertices) and one community otherwise. Let $k = b$ and

$$a = \max_{S_L \subseteq V_L, S_R \subseteq V_R, |S_L|+|S_R|=b} NMI(\mathcal{A}(G'), Y'), \tag{2}$$

where

$$Y' = \{K_L \cup V_L\backslash S_L, K_R \cap V_R\backslash S_R\}. \tag{3}$$

Assume we have a solution $S, |S| = b$ to MVC-B. Then at least c edges in E are incident to vertices in S. If we remove all vertices in S, by construction, $\mathcal{A}(G'[V'\backslash S])$ will output two communities, $K_L \cup V_L\backslash S'_L$ and $K_R \cup V_R\backslash S'_R$. Then we have

$$NMI(\mathcal{A}(G'), \mathcal{A}(G'[V'\backslash S])) \leq a \tag{4}$$

as a is the maximum NMI value for communities in the form of $\{K_L \cup V_L\backslash S_L, K_R \cap V_R\backslash S_R\}$. Therefore, we have a solution $S, |S| = b = k$ for CVAD.

Now assume we have a solution $S, |S| = k$ to CVAD. As $NMI(\mathcal{A}(G'), \mathcal{A}(G'[V'\backslash S])) \leq a$, $\mathcal{A}(G'[V'\backslash S])$ must contain two communities. By construction, number of edges removed from E is at least c. If $S \subseteq V$, we directly obtain a solution for MVC-B. If $\exists v \in S, v \notin V$, we can always find a vertex $u \in V, u \notin S$ and update S to $S' = S \cup \{u\}\backslash\{v\}$ while keeping the number of edges incident to S greater than c. Therefore, we have a solution $S', |S'| = k = b$ for MVC-B.

Since CVAD has a solution if and only if MVC-B has a solution, CVAD is NP-complete and so is CVA. □

4.2 An Approximation Algorithm for the Special Case $k = 1$

We analyze the special case when there is only one node to be excluded from the current network. Note that this case, or the case where k is a constant number, can theoretically be solved for optimality by iteratively visiting all k-tubes of nodes, and then selecting the one resulting to the lowest NMI score in comparison to the original structure. However, this brute-force searching approach is computationally intractable requiring $O(n^k \times time(\mathcal{A}))$ time as shown in the previous section. Thus, our goal is to provide an alternative approach that takes less time for finding the most important node.

The intuition behind our algorithms for this case is as follows: since the targeted node u belongs to some community X_t of X, its exclusion can possibly break X_t into smaller subcommunities which can further be merged with existing communities. Per our analysis in the previous section, the more number of new as well as merged communities we have, the lower NMI measure the new community structure shall potentially be. Therefore, our strategy emphasizes on selecting node that can break a community of X into many more subcommunities of relatively the same size as possible. By proceeding in this way, we can prove that the obtained NMI score is at most the number of newly formed subcommunities times the optimal NMI score. The procedure is described in Algorithm 1.

Algorithm 1. A Solution for CVA Problem when $k = 1$

Input: Network $G = (V, E)$, the community detection algorithm $\mathcal{A}$, set of communities $X = \{X_1, \ldots, X_{c_X}\}$;
Output: The targeted node u;
1: Run $\mathcal{A}$ on all communities X_i's of X.
2: Choose community X_t of X having the most number of subcommunities. If there is a tie, choose $X_t = C$ having the smallest size difference between its subcommunities $\sum_{X_{t_s}, X_{t_l} \in C} |X_{t_s} - X_{t_l}|$.
3: Choose the node u that is adjacent to most subcommunities in X_t.

4.2.1 Analysis

Let u be the node identified by Algorithm 1, X_t be the community u belongs to, n_t be the number of subcommunities resulted from the exclusion of u from X_t, and Y be the new community structure detected by $\mathcal{A}$ on $G[V\backslash\{u\}]$. Denote by u^* the optimal solution whose removal results in the lowest NMI score. We have the following connection

Theorem 2. *If $n_t \geq 2$, the exclusion of u from G will result in $NMI_X(\{u\}) \leq \min\{1, n_t \times NMI_X(\{u^*\})\}$.*

Proof. (Concise) Suppose X_t is broken into two or more subcommunities. Let P_1, P_2 be sets of subcommunities that are not merged and are merged with other existing communities, respectively. We have $I(X,Y) = \frac{1}{n-1}\big((n-1)\log n - H(X) + \sum_{P_2} s_{p_2} \log \frac{s_{p_2}}{x_{p_2}+s_{p_2}}\big)$ and $H(Y) = \frac{1}{n-1}\big((n-1)\log\ (n-1) + \sum_{P2}(x_{p_2}+s_{p_2})\log(x_{p_2}+s_{p_2})\big)$.

Therefore,

$$NMI_X(\{u^*\}) = \frac{A_1 + \sum_{P_2} s_{p_2} \log \frac{s_{p_2}}{x_{p_2}+s_{p_2}}}{C_1 + \sum_{P2}(x_{p_2}+s_{p_2})\log(x_{p_2}+s_{p_2})}.$$

where $A_1 = (n-1)\log n - H(X)$ and $C_1 = (n-1)H(X) + (n-1)\log(n-1)$. As we optimally minimize both terms of $NMI_X(\{u\})$, we will have the NMI measure that is theoretically smaller than those incurred by any other node $u' \in V$. This implies

$$NMI_X(\{u^*\}) \geq \frac{A_1 + x_t \log \frac{x_t}{x_{p_2}+x_t}}{C_1 + (x_{p_2}+x_t)\log(x_{p_2}+x_t)}.$$

One can show that

$$\begin{aligned} NMI_X(\{u\}) &\leq n_t \times \frac{A_1 + x_t \log \frac{x_t}{x_{p_2}+x_t}}{C_1 + (x_{p_2}+x_t)\log(x_{p_2}+x_t)} \\ &\leq n_t \times NMI_X(\{u^*\}) \end{aligned}$$

and thus, the conclusion follows. □

4.3 genEdge: A Heuristic for CVA

We present genEdge, an algorithm for CVA problem that is independent of the underlying community detection algorithm $\mathcal{A}$. Our strategy will try to break larger communities to as many small ones as possible while looking for those having relatively same size with a small overlapping ratio.

4.3.1 Intuitions

The idea of our strategy is based on the following intuition: since communities in X are optimized for their internal density, they are likely to contain strong substructures that are tightly connected which form the cores of these communities. As a result, the removal of crucial nodes in a core might potentially break the community into smaller modules. Moreover, as nodes in a core are tightly connected, there should be some edges that generate them, i.e., nodes in the core are incident to both endpoints of this edge. Inspired by this intuition, our strategy works towards the identification of these generating edges of a community, and then seek for the minimum set of generating edges that composes the original communities.

Let $\Psi(C) = \frac{2m_C}{n_C(n_C-1)}$ be the internal density of any $C \subseteq V$, and $\tau(C) = \frac{n_C(n_C-1)}{2} - \frac{2}{n_C(n_C-1)}$ be the threshold function on the internal density of C. There are several reasons for using the internal density as the objective function compared to other functions which are worth noting down. First and foremost, internal density facilitates the fundamental concept of a

network community even with community overlap. Second, internal density functions $\tau(C)$ and $\Psi(C)$ locally process the candidate community C only and neither require any predefined thresholds or user-input parameters. Third, $\Psi(C)$ and $\tau(C)$ are increasing functions and closely approach C's full number of connections, i.e., the number of edges in a clique of size $|C|$. That makes the internal density function a powerful tool for detecting local communities, i.e., densely connected parts of the network.

For any nodes $u, v \in C$, if edge (u, v) is not in E, we call it a missing edge in C. In addition, we call an edge in C "negative" if it is incident to a missing edge in C, and "positive" otherwise. We define the concept of *generating edges* of C as follows.

Definition 4 (Generating edge). *For any edge (u, v) in C, if $C = (C \cap N(u) \cap N(v)) \cup \{u, v\}$ and $\Psi(C) \geq \tau(C)$, we call (u, v) a generating edge of C. We further call C a local core generated by (u, v) and write $gen(u, v) = C$.*

For any community C of G, a set $L \subseteq E$ is called a "generating edge set" of a C if $\cup_{(u,v)\in L} gen(u, v) = C$. Since C can be generated by different generating edge sets and we are constrained on the node budget, we would intuitively seek for the generating edge set of minimal cardinality.

Definition 5 (The Minimum Generating Edge Set). *Given a community C of G, the MGES problem seeks for a generating edge set L^* of C with the smallest cardinality.*

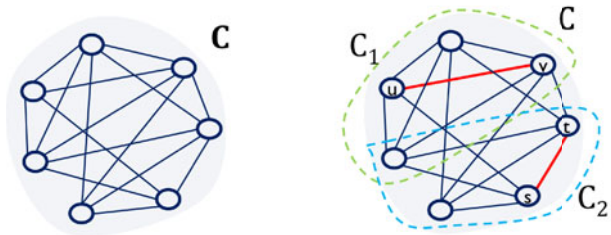
Fig. 1 illustrates this idea of cores and generating edges on a hypothetical community C. C_1 and C_2 are two cores of C since $\Psi(C_i) \geq \tau(C_i)$ for both $i = 1, 2$. The cores are marked with dotted lines in Fig. 1b. The edges (u, v) and (s, t) are the generating edges of C_1 and C_2 respectively as can be seen marked in red in Fig. 1b. $\{(u, v), (s, t)\}$ comprises the MGES of C.

The cores generated by edges in a MGES of a community C of G are tightly connected and they all together compose C. As a result, if we delete an endpoint of every edge in a MGES, C will be broken into smaller modules with the number of modules is at least the number of edges in a MGES (Lemma 2). Since our goal is to break the current community structure X into as many new communities as possible, the removal of crucial nodes defined by edges in a MGES will be a good heuristic for this purpose. But first and foremost, we need to characterize all MGESs in the current community structure X based only on the input network G. Lemma 3 realizes the location of the generating edge(s) of a local core in a community C: they have to be adjacent to nodes with the highest degree in C. Based on this result, we present in Algorithm 2 a procedure that can correctly find the MGES of a given community C (Theorem 3).

Lemma 2. *Let L^* be a MGES of a community C. The removal of an endpoint in every edge of L^* will break C into at least $|L^*|$ subcommunities.*

Proof. See Appendix B, available in the online supplemental material. □

Lemma 3. *Let C be a subset of V, $U = \{u \in C | d_u^C$ is maximum in $C\}$ and $NE(U) = \{(u, v) | u \in U$ or $v \in U$ but not both$\}$. Then, $|NE(U) \cap L^*| \geq 1$.*



(a) A community C (b) Generating Edges of two cores C_1 and C_2 of C

Fig. 1. Example of minimum generating edge set identified by genEdge.

Proof. After each reconstruction in step 3 of Algorithm 2, let u be the node with the highest indegree in C. After step 2, all negative edges are deleted since they do not contribute to the actual generating set L^*. As such, edges incident to u are not negative. This in turn implies that they are candidates for generating edges. Now, iterate through all edges incident to u and choose the one that generates the biggest-sized core. This edge will be in the list L^*. □

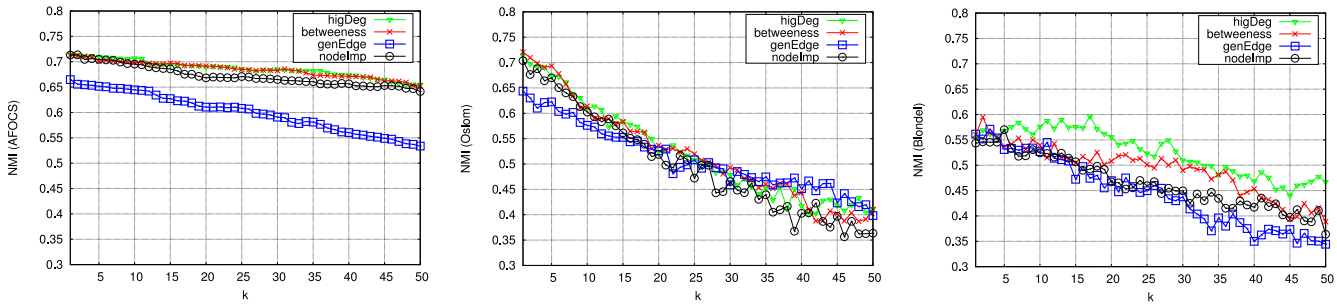
Theorem 3. *Let d_C be the maximum in-degree of a node in C. Algorithm 2 takes $O(d_C|C|)$ time in the worst case scenario and returns an optimal solution for MGES problem.*

Proof. Since every time Lemma 3 makes sure that at least one edge should be added to L^* and the procedure terminates when no edges are left, the algorithm will terminate. Moreover, it is verifiable that Algorithm 2 takes time at most the number of edges in C, which is $O(d_C|C|)$. Also, due to the intense internal density of a core, every time an edge is added into L^*, that edge actually generates the largest core possible. The proof follows from this fact, Lemma 3 and the exhaustive property of Algorithm 2. □

Algorithm 2. An Optimal Algorithm for Finding the MGES

Input: Network $G = (V, E)$ and a community $C \in X$;
Output: Minimum generating edge set L^* of C;
1: Mark all nodes as "unassigned" and $L^* = \emptyset$.
2: Remove all negative edges in C. If any edge(s) survives, they are candidate for generating edges in their corresponding communities, include them to L^*, go to step 3. Else, go to step 4.
3: Reconstruct local cores based on generating edges found in step 2. Mark all nodes in those communities as "assigned". Discard generating edges in L^* that fall into any newly constructed communities. Return if all edges are assigned.
4: Find the set U as in Lemma 3. Find the edge in $NE(U)$ that can generate a local community having the largest size. Include this edge to L^* and mark all nodes in the new local community as "assigned". Ties are broken randomly. Return if all edges are assigned.
5: If there are still unassigned nodes, say the set $I \subseteq C$, construct $G' = G[(I \cup N(I)) \cap C]$. Go back to step 2.

With the optimal solution of MGES taken into account, we next suggest a heuristic for selecting important nodes following the guidelines suggested in Section 3. In particular, our proposed heuristic, *genEdge*, as described in Algorithm 3,



(a) AFOCS (b) OSLOM (c) Blondel

Fig. 2. Performance of node selection strategies on synthesized networks with $n = 2{,}500$ nodes.

selects nodes in a greedy manner, starting from communities that have large-size MGESs. Moreover, in the MGES of each community C, we give priority to nodes that are incident to more generating edges since their removal will break C into more subcommunities.

Algorithm 3. *genEdge* - A Node Selection Strategy for CVA Based on Generating Edges

Input: Network $G = (V, E)$, $X = \mathcal{A}(G)$;
Output: A set $S \subseteq V$ of k nodes;
1: Use Algorithm 2 to find $L^*_{X_i}$ for all communities X_i's in X.
2: Sort all communities X_i's in X by their sizes of MGESs.
3: Sort all nodes in G by the number of generating edges that they are incident to in X_i. If there is a tie, sort them by their degrees in G.
4: Return top k nodes from step 3.

Although we have considered internal density as the objective function for identifying communities, our proposed method can easily be extended for other community detection schemes as well. For instance, the definition of the *generating edge* can be adapted to consider other well-known objective functions for identifying the local cores of a community. The rest of the subroutines remains unchanged and can be applied as they are once the minimum generating edge set is identified using the adapted definition of generating edges.

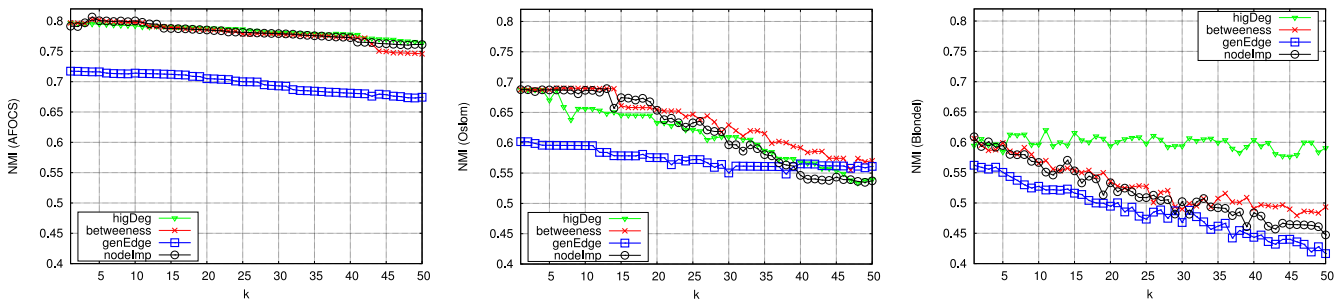
5 Experimental Evaluation

In this section, we evaluate the performance of *genEdge* in identifying the most critical nodes removal of which results in maximum change of the community structure as measured by NMI. On top of that, we not only evaluate how the community structures are changed drastically in real world social network traces, but also we show the impact it puts effectively on the performance of social based routing and forwarding schemes in real world DTNs. In order to certify the performance of our approach on tackling CVA, we compare the results obtained by the following methods: high degree centrality (higDeg) selects top k nodes in G with the highest degrees, betweenness centrality (betweeness) selects top k nodes in G with the highest betweennesses (where the betweenness of a node u is the number of shortest paths in G that passes through u), generating edges (genEdge) - our strategy described in Algorithm 3, and finally, node importance (nodeImp) [12] selects top k nodes by their importance to the community structure. It has been shown in literature that these selection strategies are well representative approaches for identifying important/crucial nodes in a network [12]. We first examine the effect of the underlying community detection methods by comparing results obtained by *AFOCS* [10], *Blondel* [13] and *OSLOM* [14] algorithms to the embedded ground truths. In particular, we set X to be the ground truth community structure and when S is removed from the network, $NMI(X, Y)$ is reported, where $Y = AFOCS(G[V \backslash S])$, $Y = Blondel(G[V \backslash S])$ and $Y = OSLOM(G[V \backslash S])$, respectively. These methods have been empirically certified in the literature to be the best algorithms for finding non-overlapping and overlapping community structure [8]. Verifying our strategy on synthesized networks with known community structure not only certifies its performance but also provides us the confidence to its behavior when applied to real-word traces.

5.1 Performance of genEdge

Setup. We use the well-known LFR overlapping benchmark [8] to generate test networks. The number of nodes are $n = 2{,}500$ and $5{,}000$, the mixing parameter $\mu = 0.15$, the community sizes $c_{min} = 10$ and $c_{max} = 50$ for $n = 2{,}500$ and $c_{min} = 30$ and $c_{max} = 100$ for $n = 5{,}000$. On these small networks, the number of removed nodes k is varied from 1 to 50. At every time k nodes are removed from the network, the network community structure is reidentified and compared to the original embedded one (or the ground truth). The overlapping threshold β in *AFOCS* is set at 0.7. All tests are averaged on 1,000 runs for consistency.

Results. We first evaluate the performance of the node selection strategies in terms of NMI score. Because the ground truth communities are given a priori, a comparison through NMI scores among these strategies as well as among detection algorithms is therefore valid, and the lower NMI score a strategy obtains the more effective it seems to be. In addition, the higher the remaining NMI values a detection algorithm obtains after node removal, the more resilient to node vulnerability it appears to be. The quality of node selection for $n = 2{,}500$ and $n = 5{,}000$ are reported in Figs. 2 and 3, respectively for different methods. In general, NMI values tend to drop down quickly as more nodes are removed from the network when $n = 2{,}500$; however, they do degrade much slowly in networks with $n = 5{,}000$. The first observation revealed in those figures is that our approach appears to achieve the best (lowest) NMI scores on almost all test cases. On average, in networks with 2,500 nodes, genEdge is 22 percent better than both higDeg and betweenness, and is 12 percent better than nodeImp.



(a) AFOCS (b) OSLOM (c) Blondel

Fig. 3. Performance of node selection strategies on synthesized networks with $n = 5{,}000$ nodes.

We observe nearly same trend in networks with $n = 5{,}000$ nodes in which genEdge outperforms other methods.

The second observation we obtain from Figs. 2 and 3 is that the top-of-the-list nodes seems to be essential to the network community structure. The removal of the topmost node only from the network brings the NMI scores to as low as 0.70 (0.80) for *AFOCS* in Figs. 2a, 3a, and to 0.7 for *OSLOM* algorithm in Figs. 2b, 3b, and to 0.58(0.6) for *Blondel* algorithm in Figs. 2c, 3c in networks with 2,500 (5,000) nodes. Furthermore, the top 30 nodes are also observed to be vital to the network community structure since their removal brings the NMI scores down to 0.5 in $n = 2{,}500$ nodes (Figs. 2b, 2c) - the threshold where the community structure become stochastic and fuzzy to recognize. In networks with $n = 5{,}000$ nodes, the removal of 50 nodes only brings the NMI measure down to 0.6, however, the generally decreasing trend suggests that excluding 90-100 nodes from the network will degrade the NMI measure to the stochastic threshold.

Finally, the last observation inferred from Figs. 2 and 3 is that, various community detection algorithms behave differently under disparate node removal strategies. Among the three community detection algorithms, *AFOCS* algorithm obtains the highest remaining NMI values when the same number of nodes is removed from the networks. Overall, *genEdge* consistently outperforms all other methods in bringing the NMI measure to lower values for all the community detection algorithms compared to other node selection methods. In other words, *genEdge* was able to detect the crucial nodes that are important for the community structures irrespective of the detection algorithm.

5.1.1 Comparison with Optimal Algorithm

CVA is a NP-complete problem as proven in Section 4.1 which means there is no efficient algorithm to find the optimal solution especially for large-scale data sets. Computational complexity increases exponentially with the problem instance size. Consequently, to get an idea of the comparative performance of the node identification methods used in this paper, we generated a network of $n = 100$ nodes using LFR benchmark setting the mixing parameter $\mu = 0.15$, the community sizes $c_{min} = 5$ and $c_{max} = 10$. We ran the brute force exhaustive algorithm on that synthesized network and removed $k = 1, \ldots, 5$ nodes to compare other methods with this oracle. However, even for k as small as 5, this algorithm took exponentially longer time as reported in Table 2. For larger value of k, we could not obtain any result for the optimal one even after 6 days of running the program. On the other hand, interestingly, all the node selection methods perform quite well in this small network in terms of minimizing the NMI. In particular, genEdge achieves very close performance with compared to the optimal one as shown in Fig. 4. For $k = 1$ and $k = 5$, genEdge reduces the NMI to 0.85 and 0.67 which are within 96.3 and 89 percent, respectively, of the optimal one. This experiment suggests, once again, genEdge achieves comparative performance without incurring any significant computational complexity. Due to space constraint, we only report the results for the *Blondel* community detection algorithm while stressing that the trend is similar for other community detection algorithms. These experiments were performed on an Intel(R) Xeon(R) W350 CPU with 24 GB-memory running the Windows operating system.

TABLE 2
Computational Complexity of Different Methods

k	higDeg	betweeness	genEdge	nodeImp	optimal
1	3	8	4	3	86
2	6	15	8	6	239
3	8	21	13	10	4,994
4	11	27	16	12	209,669
5	13	32	20	15	518,400

All time in seconds.

5.2 Performance of genEdge on Real Social Networks

In this section, we show the empirical results of our node selection strategy on real-world social traces including the

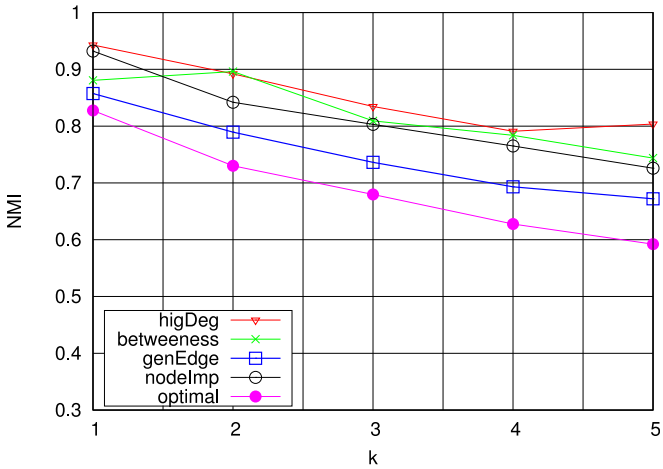


Fig. 4. Performance of node selection strategies on synthesized network with $n = 100$ nodes.

TABLE 3
Statistics of social traces

Data	n	m	Average Degree	Maximum Community Size
Reality	100	3,100	62	35
Facebook	63,731	1.5 M	23.50	33,425
Foursquare	44,832	1.1 M	49.13	30,381

Reality mining cellular dataset [15], Facebook [16] and Foursquare [17] social networks. The overview of these datasets is summarized in Table 3. Here, we only report the results we obtain for *AFOCS* community detection algorithm as the results from *Blondel* and *OSLOM* are almost similar in trend.

5.2.1 Setup

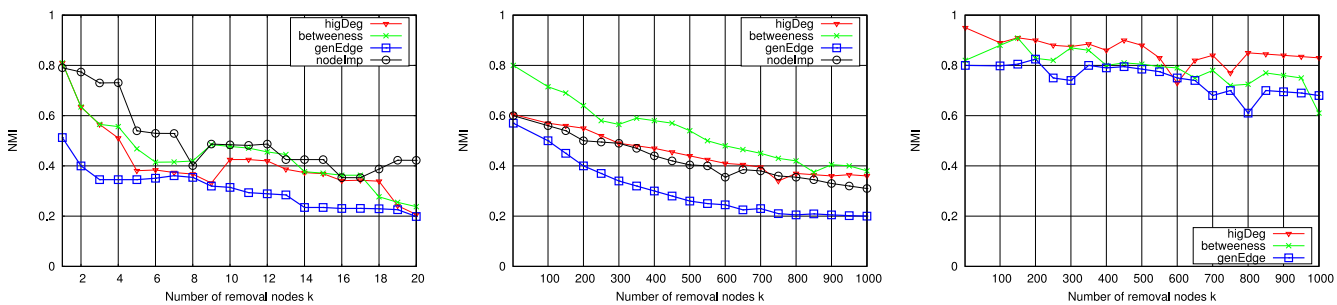
Reality Mining dataset is provided by the MIT Media Lab. This dataset contains communication, proximity, location, call, and activity information from 100 students at MIT over the course of the 2004-2005 academic year. Each node signifies a student and an edge between two nodes denotes the communication between the two students representing the nodes. *Facebook* dataset contains friendship information (i.e., who is friend with whom and wall posts) among New Orleans regional network on Facebook, spanning from September 2006 to January 2009. To collect the information, the authors created several Facebook accounts, connected each to the regional network, started crawling from a single user and visited all friends in a breath-first-search fashion. This network contains 63,731 nodes each representing a Facebook user and the edge between two nodes signifies their friendship. A node removal from Facebook can be interpreted as the closure of that account which can be triggered by the user through deactivating or deleting it or by the authorities for regulatory and compliance issues. It is interesting to see how the removal of Facebook user can impact the Facebook communities.

Foursquare dataset contains location and activities of 44,832 users on Foursquare social network on May 2011 - July 2011. To collect the data, we created several Foursquare accounts, joined the network, started crawling from a single user and visited all friends also in a breadth-first-search fashion. An edge between two nodes denotes their friendship in the network. Similar to Facebook network, a node removal can be viewed as the closing of that account by the user himself or by the respective authorities.

5.2.2 Results

On Reality Mining dataset, we set $k = 1...20$ and report result in Fig. 5a. It reveals from this figure that community structure in this dataset is extremely vulnerable to node attacks since the removal of only 2 nodes, found by $genEdge$ is enough to make the new community structure significantly different from the original one as it brings down the NMI values to 0.4. In comparison with other node selection methods, $genEdge$ still performs excellently and is about 14 - 17 percent better than the others. We note that the first node identified by $genEdge$ is indeed crucial to the community structure of this network since it immediately brings down NMI score to 0.5 while the other does not seem to discover this important node. Furthermore, when too many nodes are removed from the network, the network does not seem to contain communities any more or the community structure becomes extremely fuzzy as NMI values converge down to around 0.2. This is understandable since this dataset is of small size with a very high average node degree.

On larger networks of Facebook and Foursquare, we set k from 50 nodes to 1,000 nodes (only 2.1 and 1.5 percent number of nodes of Foursquare and Facebook networks, respectively) with a 50-node increment at a time. The numerical results are reported in Figs. 5b and 5c, respectively. In general, NMI values of all methods degrade quickly on Foursquare network, and tend to decrease slower on Facebook network. As more nodes are excluded from the network, $genEdge$ still achieves the best performance on both networks with significantly lower NMI values than the other methods. Specifically, on Foursquare with high average degree and internal community density, the removal of nodes incident to the most generating edges in $genEdge$ significantly leads to the separation of network community structure as NMI scores drop down to 0.2 for $genEdge$. On Facebook network, the similarity between the original and new community structure seems to retain fairly high even when all 1,000 nodes are removed, whereas the new structure of Foursquare network is at the edge of stochastic threshold as can be seen from the small NMI measure of around 0.2. This implies that community structure in Foursquare network is also extremely vulnerable to node removal, while the mature Facebook network does not seem to suffer much. One possible reason for this is since Facebook contains a giant community with low average degree, it therefore requires much more effort in order to break that giant community apart.



(a) Reality (b) Foursquare (c) Facebook

Fig. 5. NMI scores on reality mining data, foursquare and facebook networks obtained by AFOCS.

The better performance of *genEdge* along with the structural vulnerability of network communities encourages us to further investigate how the node removal would impact the performance of social-aware routing and forwarding methods in real-world DTN traces, which we accomplish in the next section.

5.3 Impact of genEdge on Community-Based DTN Routing Protocols

In this section, we determine the impact of important node removal on the routing and forwarding of real world DTN traces and present the empirical results on them including the Reality Mining data [15] and the Haggle project [18]. Identified critical nodes in DTNs henceforth can be used to develop more secure and reliable networks in terms of efficient opportunistic routing and forwarding. For instance, the awareness of this vulnerability can help in designing a forwarding algorithm that does not overload those crucial devices by flooding the limited resources (e.g., queue capacity), or in designing an effective backup plan when some of them may fail at the same time.

In order to test how the performance of different forwarding and routing algorithms is impacted due to structural vulnerability of communities through CVA, we pick popular social-based routing strategies: Bubble-Rap [4] (a community-based), SimBetTS [19] (a community-centrality based) and Epidemic [20] (the baseline flooding) as forwarding and routing algorithms.

Epidemic routing ensures delivery of messages among mobile devices in DTNs using a random pair-wise exchange of messages. In this approach, messages are stored locally and, are forwarded or replicated to the encountered nodes whenever an opportunity occurs. It serves as an excellent baseline for comparison of routing and forwarding methods in DTNs.

SimBetTS is a routing method where the routing problem has been cast as an information flow problem in a social network. It forwards messages only to the nodes with a higher likelihood of meeting the destination. This scheme relies on the centrality measures of individual devices which are based on a node's past social interactions. It achieves comparable delivery performance with Epidemic Routing, but with significantly reduced overhead.

Bubble-rap scheme focuses on two specific aspects of a network: community and centrality. Within a community, some devices are more important, and interact with more devices than others (i.e., have high centrality); these devices are called hubs. Exploiting this kind of community information to select forwarding paths and relays to the destination is the main idea behind this approach.

To detect network communities, we choose *AFOCS*, *OSLOM* and *Blondel* algorithms due to their superior performance [8]. However, we only depict the results for *AFOCS* and *OSLOM* in this section due to space constraint. Nevertheless, the outcomes from *Blondel* adhere to the general trend we observe for the other two community detection methods. In order to observe behavior of different node selection approaches, similar to what we did in synthesized networks, we compare *genEdge* to the following methods: high degree centrality, betweenness centrality, and node importance (nodeImp). The references for all methods can be found in Table 4.

TABLE 4
Experimental Datasets and Methods

Data	Community detection Alg.	Routing Alg.	Node selection Alg.
Haggle [18]	AFOCS [10]	Epidemic [20]	high Degree
Reality [15]	OSLOM [14]	Bubble-Rap [4]	betweenness
–	Blondel [13]	SimBetTS [19]	nodeImp [12]
–	–	–	genEdge

5.3.1 Setup

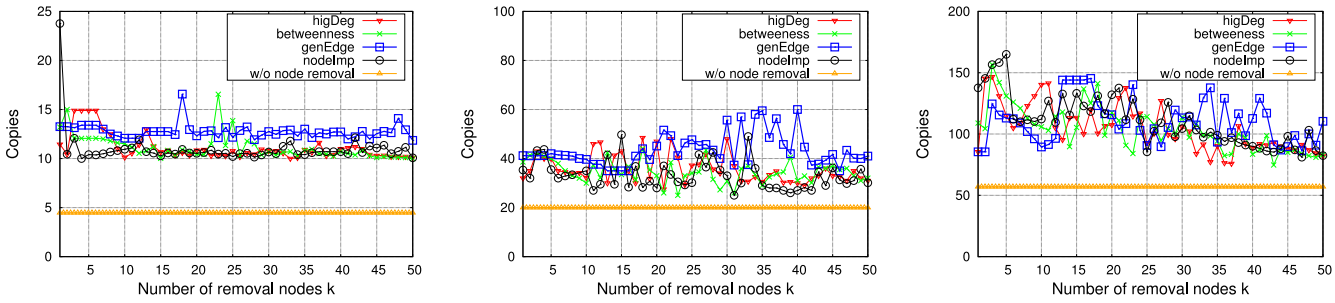
The number of excluded nodes k is from $1 \ldots 20$ for Reality dataset and is from $1 \ldots 50$ for Haggle dataset. In each experiment, 500 message sending requests are randomly generated and distributed in different time points. To control the forwarding process, we use hop-limit, time-to-live, and max-copies parameters. A message cannot be forwarded more than hop-limit hops in the network or exist in the process longer than time-to-live; it will be automatically discarded once the limits are reached. Moreover, the maximum number of same messages a device can forward to the others is restricted by max-copies. Results are averaged over several runs of the experiments for consistency. Description of datasets can be found in the provided references.

For each set of datasets, we report the average number of duplicated messages (or the overhead) that the system generates as well as the average ratio of message delivery. The orange straight line in each plot represents the original value attained by the forwarding and routing simulation without any node removed. Due to page limit, we exclude the graphs of the delivery latency of all routing algorithms in this section. Nonetheless, similar to the increased number copies and reduced delivery ration, the delay latency also increases as important nodes are removed from the network.

5.3.2 Results

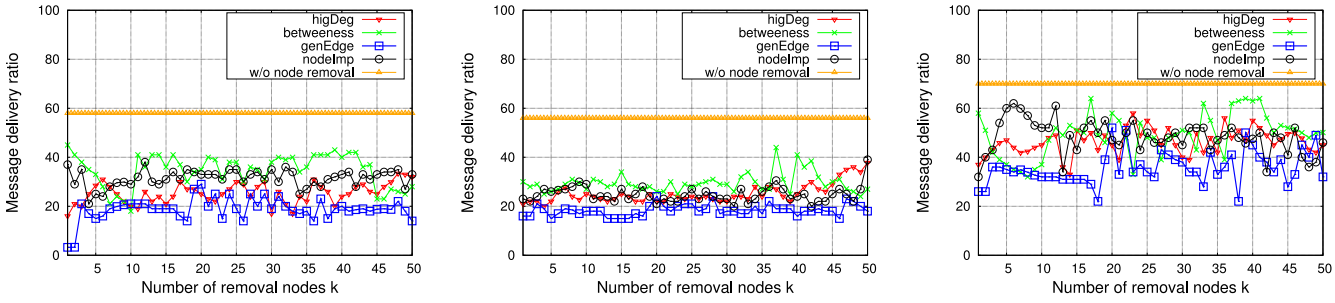
While we expect the performance of SimBetTS to deteriorate upon the removal of important nodes, we anticipate the performance of Bubble-Rap to deteriorate more quickly, because of the reliance of the protocol on the community structure since Bubble-Rap resorts on the knowledge of the community structure to route the messages.

The results on Haggle dataset are presented in Figs. 6, 7, 8, and 9. As revealed through these figures, the nodes or devices identified by genEdge method have great impact on the performance of forwarding and routing algorithms in this dataset. In general, the successful message delivery ratios attained by genEdge are the lowest among all of the node selection methods. In particular, the average delivery ratios of genEdge is about 10-15 percent lower than higDeg, betweenness and nodeImp with *AFOCS* community detection algorithm, and is about 5-7 percent lower than the other methods when *OSLOM* is the underlying community detection technique. Moreover, in comparison with the delivery ratio attained by the original system, the delivery ratios attained by genEdge is significantly lower than those obtained by the unaltered system. These results indicate that those nodes and devices selected by these methods are not only crucial to the network community structure but also play a vital role in maintaining the reliability of the network performance.



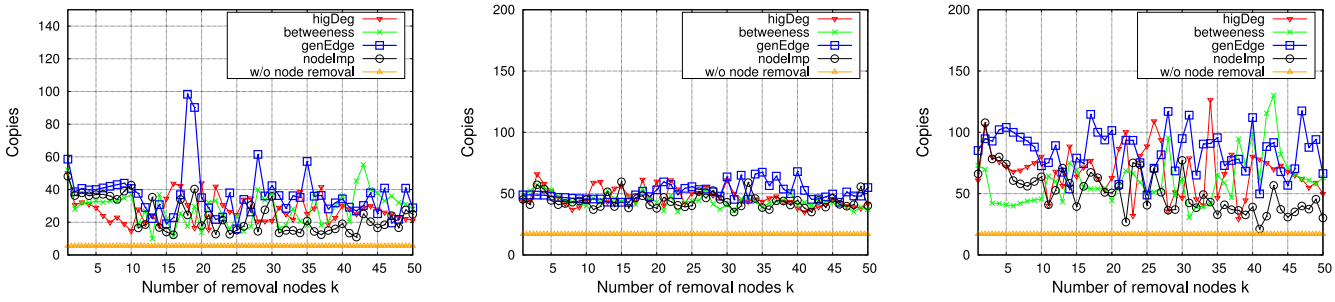
(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 6. Average number of duplicate messages on Haggle dataset with *AFOCS*.



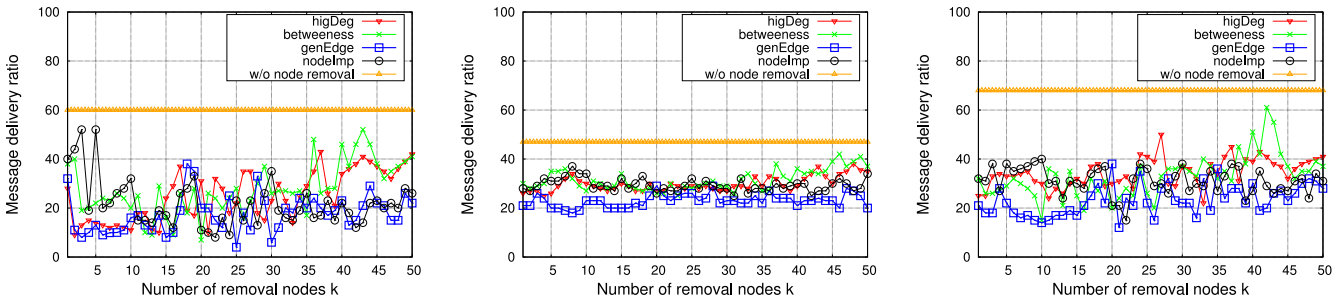
(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 7. Average ratio of message delivery on haggle dataset with *AFOCS*.



(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 8. Average number of duplicate messages on haggle dataset with *OSLOM*.

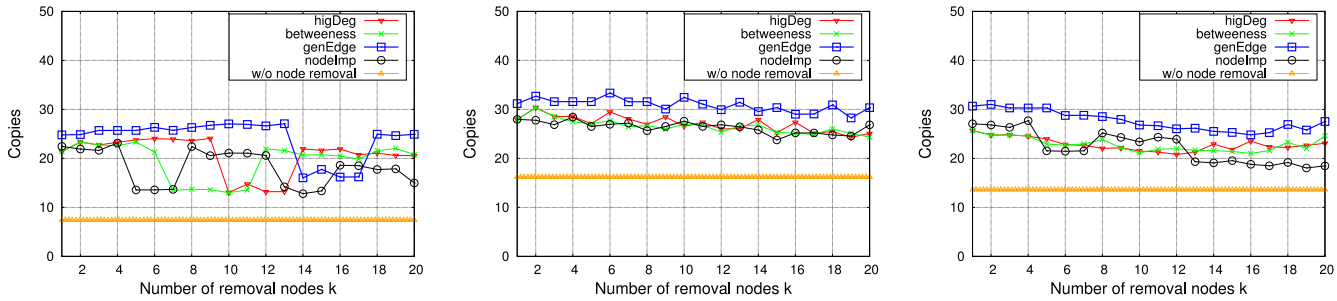


(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 9. Average ratio of message delivery on haggle dataset with *OSLOM*.

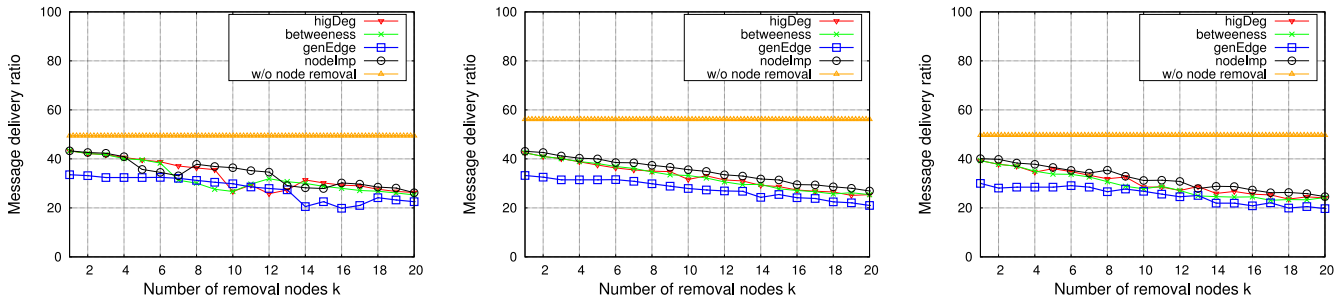
In terms of the number of duplicated messages, the exclusion of nodes selected by genEdge introduces a significant number of duplicated messages into the system. In a general trend, the number of duplicate messages introduced by genEdge is the highest one among all forwarding and routing strategies, and is much more than the uninterrupted system in cases of both *AFOCS* and *OSLOM*. In particular, the number of duplicated messages introduced by genEdge is about $1.05 \sim 1.3$ times more than higDeg, betweenness and nodeImp on average in all test cases. This can be explained further by, again, the node selection strategy of genEdge when breaking the core of each community in the network, thus better separating the forwarding route from the source to the destination.

In a fairly large DTN as Haggle (around 5,000 participants), we observed that the average delivery ratio and the



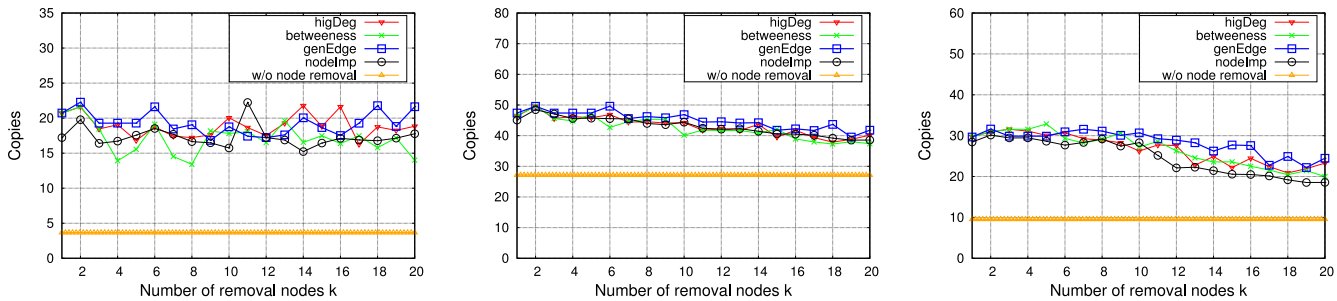
(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 10. Average number of duplicate messages on reality dataset with *AFOCS*.



(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 11. Average ratio of message delivery on reality dataset with *AFOCS*.



(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 12. Average number of duplicate messages on reality dataset with *OSLOM*.

number of duplicated messages do not strictly increase when more and more nodes are excluded from the networks. This can be explained by the merging of network communities: when crucial nodes leave the networks, their communities can be broken into smaller subcommunities (due to lesser internal interactions) and might be merged to other communities. This combination process could possibly bridge some pairs of sources and destinations, thus increasing the chances of message delivery.

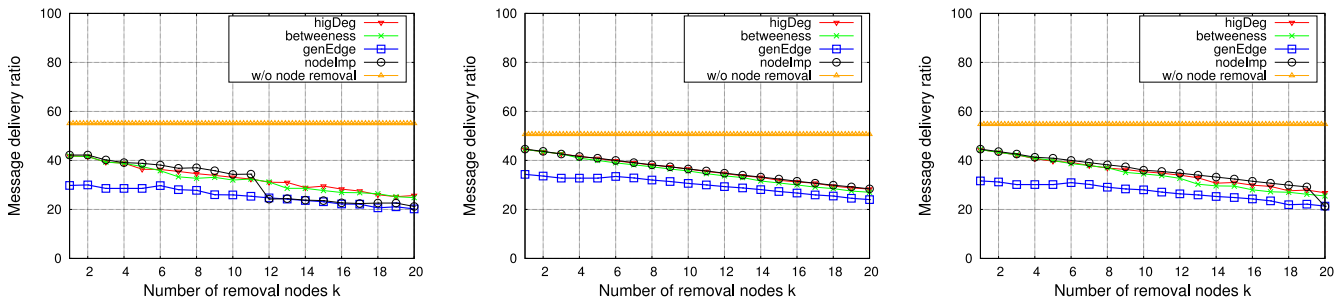
The results on Reality dataset are depicted in Figs. 10, 11, 12, and 13, the empirical results again confirm the superiority of genEdge over other node selection methods. In a big picture, the results obtained by genEdge are the best ones among four node selection methods. This also shows the vulnerability of community-based routing schemes specially when the critical nodes identified by genEdge are removed. In this medium-size dataset (100 participants), the decreasing trend of the message delivery can be visualized clearly. The reason is due to the strength of each community in this network: since participants are students in the same institution, they retain in their communities and do not appear to change their community much. As a result, the more nodes are removed the lower the number of delivered messages.

Each of the empirical results as shown in all of the figures also confirms that, community-based routing and forwarding in opportunistic networks are vulnerable to node removals regardless of the scheme through which these nodes are chosen and can adversely impact the whole network performance if not safeguarded properly. Knowledge of these critical nodes would enable devising resilient protocols by protecting vulnerable devices which in turn would help these networks from malfunctioning in case of targeted attacks.

6 Related Work

Our work involves two major areas of networking research: community structure analysis and structural network vulnerability assessment.

The literature on community structure and its detection can be found in an excellent survey [8]. Assessing the



(a) Bubble-Rap (b) SimBetTS (c) Epidemic

Fig. 13. Average ratio of message delivery on reality dataset with *OSLOM*.

vulnerability of network community structure, however, has so far been an untrodden area. The first attempt on this research direction [21] has suggested potential conditions for the transformation of the network community structure, and has proposed multiple heuristic algorithms based on the modularity contribution of communities in the network. These approaches, while are efficient in analyzing disjoint structures, face nonnegligible limitations when applied to real networks displaying overlapping community structures. As a result, the need for an effective algorithm that can assess the vulnerability of the general network structures is of desire. Although the authors in [22] discuss the overlapping community structure vulnerability, they do not explore the vulnerability of community based routing in opportunistic networks limiting the scope only to online social networks.

Alim et al. [23] introduce the concept of *broken community* and analyze the vulnerability of communities in the context of arbitrary community detection algorithms. However, their goal is focused on identifying critical edges which will break maximum number of communities. Aside from that, a large body of work has been devoted to find the node roles within a community by a link-based technique together with a modification of node degree [24], by using the spectrum of the graph [12]. However, none of these approaches discusses how the community structure would change in the removal or failure of those important nodes, especially in terms of NMI measure.

On the assessment of network vulnerability, existing studies mainly focus on assessing the centrality measurements [25], including degree, betweenness and closeness centralities, average shortest path length [26]. A good survey about network vulnerability assessment can be found in the work of Grubesic et al. [25]. However, there is an even more crucial risk that could dramatically affect the normal network functionality that has not been addressed so far: *the transformation or restructure of the network community structure*. Due to its vital role in the network, any significant restructure or transformation of the community structure, resulted from important node removal, can potentially change the entire network organization and consequently lead to a malfunction or unpredictable disruption of the whole network function.

7 Conclusion

In this work, we have studied the structural vulnerability of social-aware routing and forwarding schemes in opportunistic networks. In order to assess system fragility from community structure point of view, we have proposed the CVA problem, analyzed the minimization of NMI measure and provided key insights into the selection of nodes that are crucial to the community structure. We have suggested an approximation algorithm for the case $k = 1$ and also presented genEdge, a heuristic for CVA problem when $k > 1$, based on the concept of minimum generating edge set. To certify the effectiveness of the suggested algorithms, we have tested them on synthesized networks with known community structures and the performance of genEdge on these networks sets out the corner stone of deploying it on real-work social and DTN traces. We have shown that community-based forwarding and routing methods in DTNs are really sensitive to the change of network communities, the nonparticipation of only some important devices is significant enough to degrade the performance of the entire network.

Acknowledgments

This work is supported in part by the National Science Foundation CAREER Award grant number 0953284 and the fellowship grant from Information and Communication Technology Division of Bangladesh.

References

[1] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 387–401, Jan.-Mar. 2013.

[2] E. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Commun.*, vol. 6, no. 2, pp. 46–55, Apr. 1999.

[3] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proc. 5th Annu. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2007, pp. 65–70.

[4] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[5] W. Gao, Q. Li, B. Zhao, and G. Cao, "Social-aware multicast in disruption-tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1553–1566, Oct. 2012.

[6] E. Bulut and B. Szymanski, "Friendship based routing in delay tolerant mobile social networks," in *Proc. IEEE Global Telecommun. Conf.*, 2010, pp. 1–5.

[7] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Statist. Mech.: Theory Exp.*, vol. 2005, no. 09, p. P09008, 2005.

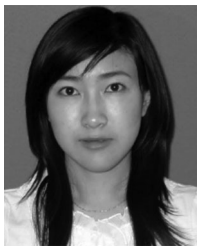
[8] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, no. 5, p. 056117, Nov. 2009.

[9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

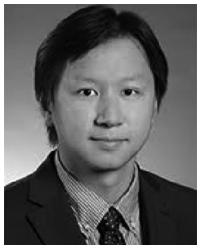
[10] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, "Overlapping communities in dynamic networks: Their detection and mobile applications," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 85–96.

[11] N. Apollonio and B. Simeone, "The maximum vertex coverage problem on bipartite graphs," *Discrete Appl. Math.*, vol. 165, pp. 37–48, 2014.
[12] Y. Wang, Z. Di, and Y. Fan, "Identifying and characterizing nodes important to community structure using the spectrum of the graph," *PLoS ONE*, vol. 6, no. 11, p. e27418, 11 2011.
[13] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech.: Theory Exp.*, vol. 2008, no. 10, p. P10008, 2008.
[14] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, p. e18961, 04 2011.
[15] N. Eagle and A. (Sandy) Pentland, "Reality mining: Sensing complex social systems," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 255–268, Mar. 2006.
[16] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proc. 2nd ACM SIGCOMM Workshop Soc. Netw.*, 2009, pp. 37–42.
[17] F. Data. (2012). Collected data [Online]. Available: http://cise.ufl.edu/∼alim/original_fsq.txt.
[18] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. (2009, May). CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29) [Online]. Available: http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006.
[19] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant manets," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, May 2009.
[20] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw.*, 2005, pp. 244–251.
[21] N. P. Nguyen, M. A. Alim, Y. Shen, and M. T. Thai, "Assessing network vulnerability in a community structure point of view," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining*, 2013, pp. 231–235.
[22] M. A. Alim, N. P. Nguyen, T. N. Dinh, and M. T. Thai, "Structural vulnerability analysis of overlapping communities in complex networks," in *Proc. 2014 IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)-Vol. 01*, IEEE Computer Society, 2014, pp. 5–12.
[23] M. A. Alim, A. Kuhnle, and M. T. Thai, "Are communities as strong as we think?" in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining*, 2014, pp. 314–319.
[24] J. Scripps, P.-N. Tan, and A.-H. Esfahanian, "Node roles and community structure in networks," in *Proc. 9th WebKDD 1st SNA-KDD 2007 Workshop Web Mining Soc. Netw. Anal.*, 2007, pp. 26–35.
[25] T. H. Grubesic, T. C. Matisziw, A. T. Murray, and D. Snediker, "Comparative approaches for assessing network vulnerability," *Int. Regional Sci. Rev.*, vol. 31, pp. 88–112, 2008.
[26] R. Albert, I. Albert, and G. L. Nakarado, "Structural vulnerability of the north american power grid," *Phys. Rev. E*, vol. 69, no. 2, p. 025103, Feb. 2004.

Md Abdul Alim received the bachelor of science degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Bangladesh in 2007. He is currently working toward the PhD degree in the Computer and Information Science and Engineering Department, University of Florida under the supervision of Dr. My T. Thai. His research interests include network vulnerability and community structure analysis in complex networks, social-aware device-to-device communication, influence propagation in online social networks and approximation algorithms and its application in combinatorial optimization.



Xiang Li received the master of science degree from the Academy of Mathematics Systems and Science, Chinese Academy of Sciences, Beijing in 2012 and the master of science in industrial and systems engineering from the University of Florida, where she has been working toward the PhD degree in the CISE Department from Fall 2014. Her current research interests include online social networks, network vulnerability, and security in Smart Grid.



Nam P. Nguyen received the BS degree from Vietnam National University and the MS degrees from Ohio University in 2007 and 2009, respectively, and the PhD degree in computer science from the University of Florida in 2013. He is currently an assistant professor in the Computer and Information Sciences Department at Towson University. His research interests focus on network structure analysis, BigData, social-aware information mining, cyber security, and practical mobile computing applications.

My T. Thai (M06) received the PhD degree in computer science from the University of Minnesota, in 2005. She is a professor in the Computer and Information Science and Engineering Department, University of Florida. Her current research interests include algorithms and optimization on network science and engineering. She has engaged in many professional activities, such as being the PC chair of EEE IWCMC 12, IEEE ISSPIT 12, and COCOON 2010. She is a founding EiC of *Computational Social Networks journal*, an associate editor of *Journal of Combinatorial Optimization*, *IEEE Transactions on Parallel and Distributed Systems*, and a series editor of Springer Briefs in Optimization. She has received many research awards including a UF Provosts Excellence Award for assistant professors, a DoD YIP, and an US National Science Foundation (NSF) CAREER Award. She is a member of the IEEE.

Abdelsalam Helal is a professor in the CISE Department at the University of Florida, and the director of its Mobile and Pervasive Computing Laboratory. He has recently been awarded a Finland Distinguished Professorship - FiDiPro (2011-2014) and a senior visiting fellow at the Institute of Advanced Studies at the University of Bologna, Italy. His active areas of research focus on mobile, pervasive and ubiquitous systems and their applications. He is a cofounder of the *IEEE Pervasive Computing magazine* and has served on its editorial board since 2002. He recently served as an IEEE Pervasive Computing's associate editor-in-chief, and currently serves as the editor-in-chief of *IEEE Computer*, the Computer Society's flagship, and premier publication. He founded two startups: Phoneomena, Inc. (2002-2007) and Pervasa, Inc., (2006-2011) and is an inventor or coinventor of nine published US patents. He is a fellow of the IEEE.