On Approximation of New Opimization Methods for Assessing Network Vulnerability

Thang N. Dinh, Ying Xuan, My T. Thai
CISE Department
University of Florida
Gainesville, FL, 32601
Email: {tdinh, yxuan, mythai }@cise.ufl.edu

E.-K. Park
CSEE Department
University of Missouri at Kansas City
Kansas City, MO 64110
Email: ekpark@umkc.edu

Taieb Znati
CS Department
University of Pittsburgh
Pittsburgh, PA 15215
Email: znati@cs.pitt.edu

***Abstract*—Assessing network vulnerability before potential disruptive events such as natural disasters or malicious attacks is vital for network planning and risk management. It enables us to seek and safeguard against most destructive scenarios in which the overall network connectivity falls dramatically. In this paper, we study a new problem called β-disruptor. The goal is to identify nodes (links) in the network whose removal results in a large number of disconnected pair of nodes i.e. there is no functional path between them.**

We formulate the problem as a graph-based optimization problem and prove that detecting such a set of nodes (edges) in general case is NP-complete, thus it cannot be solved in polynomial time unless P = NP. We present an $O(\log n \log \log n)$ pseudo-approximation algorithm for detecting the set of nodes and an $O(\log^{1.5} n)$ pseudo-approximation algorithm for detecting the set of edges. Our proposed approximation algorithms can handle both homogeneous and heterogeneous networks with unidirectional links.

I. Introduction

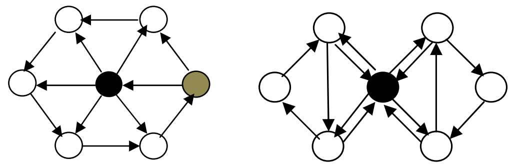
Connectivity plays a vital role in network performance and is fundamental to vulnerability assessment. Potential disruptive events, such as natural disasters or malicious attacks, which always destroy a set of interacting elements or connections, can dramatically compromise the connectivity and result in considerate decline of the network QoS, or even breakdown the whole network [1] [2] [3] [4]. Of this concern, pre-active evaluation over the network vulnerability with respect to connectivity, in order to defense such potential disruptions, is quite essential and beneficial to the design and maintenance of any infrastructure networks, for example, communication, commercial, and social networks.

Most studies over network vulnerability abstract the network as a graph $G = (V, E)$, which consists of a set of vertices V and a set of edges E representing the communication links. Due to the inhomogeneity of general graphs, it is often the case that removing some vertices and edges will decrease the network connectivity to a greater extent than removing other ones. Therefore, these vertices and edges are more critical to the overall graph connectivity, hence the corresponding elements and connections in the network reveal a higher risk in the front of potential disruptions.

There have been numerous efforts on proposing evaluation measures of the network vulnerability, as summarized in [1]. On one hand, several global graph measures, such as Cyclomatic number, Maximum network circuits, Alpha index, and Beta index, which investigate basic graph properties, i.e., number of vertices, edges and pairwise shortest paths, are adopted to evaluate the network vulnerability. However, these global measures can neither be rigorously mapped to the over network connectivity, nor reveal the set of most critical vertices and edges, thus are not suitable to assess the network vulnerability in terms of connectivity. On the other hand, researchers focused on local nodal centrality [5], such as degree centrality, betweenness centrality and closeness centrality, in order to differentiate the critical vertices from the others, and further evaluate the network by quantifying such vertices. Unfortunately, being unable to cast these local properties to global network connectivity, these measures fail to indicate accurate vulnerabilities and cannot reveal the global damage done on the network under attacks.

Instead of such detours, we model the objective network as a connected directed graph, and directly quantify the minimized set of vertices/edges whose removal incurs a certain level of network disruption, i.e., reduces the overall pairwise connectivity to some certain value as a measure for the objective graph, where the connectivity for each vertex pair is quantified as 1 if they are strongly connected and 0 if not. The motivation behind this is to explore the number of necessary disruptive events to incur a certain level of disruption in the objective network, i.e., the more vertices/edges required to be removed, the less vulnerable the network is; conversely, the fewer vertices/edges needed to removed, the easier this network is to be destroyed.

In the light of the above arguments, the following new optimization problem: *finding a minimized set of vertices/edges whose removal degrades the pairwise connectivity to a desired degree* has to be solved in order to assess the vulnerability of the objective network. Considering that disrupting these vertices and edges will considerately degrade the network performance, we refer to them as *β-disruptor* throughout this paper, where $0 \leq \beta < 1$ denotes the fraction of desired pairwise connectivity (which we will define later). Two new optimization problems *β-vertex disruptor* and *β-edge disruptor* will be studied and proved to be NP-complete. We addressed them with several pseudo-approximation algorithms with a provable performance bound, which thus ensure the



(a) network A (b) network B

Fig. 1. After the "central" vertex (in black) with maximum out-going degree is removed, A is still strongly connected while B is fragmented; however in fact, only removing one vertex (in grey) is enough to destroy A.

feasibility and accuracy of this evaluation measure.

The benefit of our new measure can be briefly illustrated in Fig.1, compared with the assessment using degree centrality. Notice that both networks A and B are with 7 vertices and originally strongly connected. According to the nodal degree centrality, removing the black vertex with maximum outgoing degree 5 in Fig.1-(a) leaves the network A still strongly connected with 5 vertices; and removing the black vertex with maximum outgoing degree 4 in Fig.1-(b) partitions the graph into two strongly connected components. In this sense, network A is somewhat stronger (less vulnerable) than B. However, our model can discover that, deleting only the grey vertex in A will be enough to bring the overall connectivity to 0; on the contrary, at least 3 vertices in B are required to be removed to make overall connectivity 0. Therefore, A is actually much more vulnerable. Apparently, our measure provides more accurate assessment.

Furthermore, our study over the multiple disruption levels (different values of β) presents a deeper meaning and greater potentials. Several recent studies in the context of wireless networks have aimed to discover the nodes/edges whose removal disconnects the network, regardless of how disconnected it is [6] [7] [8] [9]. Apparently, this is a weaker version of our *β-disruptor*, since no specification over the quantified network connectivity is concerned. However, it is not reasonable to limit the possible disruption to only disconnecting the graph, ignoring how fragmented it is. For instance, a scale-free network can tolerate high random failure rates [10], since the destructions to boundary vertices may not significantly decline the network connectivity even though the whole graph becomes disconnected. In addition, different disruption levels may require different sets of disruptor on which our model can differentiate whereas existing methods cannot. For example, the node centrality method always returns a set of nodes with non-increasing degrees regardless of the disruption level.

The main contributions of this paper are as follows:

- Introducing a novel quantitative function based on an optimization problem *β-disruptor* on general graphs, which consists of two versions *β-vertex disruptor* and *β-edge disruptor*;
- Proving the NP-completeness of the two problems above and further proving that no PTAS exists for *β-vertex disruptor*;
- Providing an $O(\log^{\frac{3}{2}} n)$ pseudo-approximation algorithm for *β-vertex disruptor*, and an $O(\log n \log\log n)$ pseudo-approximation algorithm for *β-edge disruptor*. These solutions can be applied to both homogeneous networks and heterogeneous networks with unidirectional links and non-uniform nodal properties.

The paper is organized as follows. We include the problem definition, models and notations in Section II. We provide the hardness results in Section III. The pseudo-approximation algorithms for *β-edge disruptor* and *β-vertex disruptor* are presented in Section IV and Section V respectively. Related work can be found in Section VI. Section VII summarizes the whole paper.

II. Model and Definitions

Besides the homogeneous network model consisting of uniform nodes and bidirectional links, the heterogeneous network model, where various interacting elements of different kinds are connected through unidirectional links with non-uniform expenses, finds numerous applications nowadays [11]–[13], but as well, exhibits multiple difficulties for optimization and maintenance. In the light of this, we abstract our general network model as a directed graph $G(V,E)$, where V refers to a set of nodes and E refers to a set of unidirectional links. The expense of each directed edge (u,v) between vertex u and v is quantified as a nonnegative value $c(u,v)$, for all the $|E| = m$ links among $|V| = n$ nodes. As mentioned above, our evaluation over the network vulnerability is based on the value of *overall pairwise connectivity* in the abstracted graph, which is defined as follows: given any vertex pair $(u,v) \in V \times V$ in the graph, we say that they are connected iff there exists paths between u and v in *both directions* in G, i.e., strongly connected to each other. The pairwise connectivity $p(u,v)$ is quantified as 1 if this pair is connected, 0 otherwise. Since the main purpose of network lies in connecting all the interacting elements, we study on the aggregate pairwise connectivity between all pairs, that is, the sum of quantified pairwise connectivity, which we denote as $\mathcal{P}(G) = \sum_{u,v \in V \times V} p(u,v)$ for graph G. Apparently $\mathcal{P}(G)$ is maximized at $\binom{n}{2}$ when G is a strongly connected graph.

Based on this, we study the following problems::

Definition 1: (*Edge disruptor*) Given $0 \leq \beta < 0$, a subset $S \subset E$ in $G = (V,E)$ is a *β-edge disruptor* if the overall pairwise connectivity in the $G[E \backslash S]$, obtained by removing S from G, is no more than $\beta\binom{n}{2}$. By minimizing the cost of such edges in S, we further have the *β-edge disruptor problem*, i.e., find a minimized β-edge disruptor in a strongly connected graph $G(V,E)$.

Recall that G is strongly connected iff for every vertex v in G, there is a directed path from v to all other vertices. A subgraph of G is called a *strongly connected component* (SCC) iff it is a maximal subgraph of G with all vertex pairs u,v within it connected by directed paths in both directions. Assume that a β-edge disruptor disrupts the connectivity in $G(V,E)$ by separating it into several smaller SCCs, say C_i

for $i = 1 \dots l$ i.e. $V = \biguplus_{i=1}^{l} C_i$. We have:

$$\mathcal{P}(G) = \sum_{i=1}^{l} \binom{|C_i|}{2} = \frac{1}{2}\left(\sum_{i=1}^{l} |C_i|^2 - |V|\right) = \frac{1}{2}\left(\frac{n^2}{l} - n\right) - \frac{1}{2}Var(C)$$

where $Var(C) = \sum_{i=1}^{l}(C_i - \overline{C})^2 = \sum_{i=1}^{l}(C_i - \frac{n}{l})^2$. Therefore, the two key factors affecting pairwise connectivity are the number of SCCs and the variance of their sizes. They provide us an alternative measure for evaluating the balance and fragmentation of the network.

Similarly, we define *β-vertex disruptor* and its corresponding optimization problem:

β-vertex disruptor problem: Given a strongly connected graph $G(V, E)$ and a fixed number $0 \le \beta \le 1$, find a subset $S \subseteq V$ with the minimum size such that the total pairwise connectivity in $G[V \setminus S]$, obtained by removing S from G, is no more than $\beta\binom{n}{2}$. Such a set S is called β-vertex disruptor.

III. Hardness Results

In this section we show that both the β-edge disruptor and β-vertex disruptor in directed graph are NP-complete that denies the existence of polynomial time exact algorithms unless P = NP. We state a stronger result that both problems are NP-complete even in undirected graph with unit cost edges.

Note that only in this section we consider the problem for undirected graph $G(V, E)$. All results in other sections are studied on directed graphs, thus solving both homogeneous and heterogeneous networks.

A. NP-completeness of β-edge disruptor

The proof involves a reduction from the balanced cut problem.

Definition 2: A cut $\langle S, V \backslash S\rangle$ corresponding to a subset $S \in V$ in G is the set of edges with exactly one endpoint in S. The cost of a cut is the sum of its edges' costs (or simply its cardinality in the case all edges have unit costs). We often denote $V \backslash S$ by $\bar{S}$.

Finding a min cut in the graph is polynomial solvable [14]. However, if one asks for a somewhat "balanced" cut of minimum size, the problem becomes intractable. A balanced cut is defined as following:

Definition 3: (Balanced cut) Let f be a function from the positive integers to the positive reals. An f-balanced cut of a graph $G(V, E)$ asks us to find a cut $\langle S, \bar{S}\rangle$ with the minimum size such that $|S|, |\bar{S}| \ge f(|V|)$.

Abusing notations, for $0 < c \le \frac{1}{2}$, we also use c-balanced cut to find the cut $\langle S, \bar{S}\rangle$ with the minimum size such that $\min\{|S|, |\bar{S}|\} \ge c|V|$. We will use in our proofs the following results on balanced cut shown in [15]:

Corollary 1: (Monotony) Let g be a function with the property

$$0 \le g(n) - g(n-1) \le 1$$

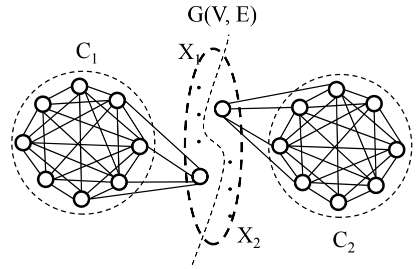


Fig. 2. Construction of $H(V_H, E_H)$ from $G(V, E)$

Then $f(n) \le g(n)$ for all n, implies f-balanced cut is polynomially reducible to g-balanced cut.

Corollary 2: (Upper bound) αn^{ϵ}-balanced cut is NP-complete for $\alpha, \epsilon > 0$.

It follows from Corollaries 1 and 2 that for every $f = O(\alpha n^{\epsilon})$ f-balanced cut is NP-complete. We are ready to prove the NP-completeness of β-edge disruptor:

Theorem 1: (β-edge disruptor NP-completeness) β-edge disruptor in undirected graph is NP-complete even if all edges have unit weights.

Proof: We prove the result for the special case when $\beta = \frac{1}{2}$. For other values of β the proof can go through with a slight modification of the reduction. We consider n to be a large enough number in our proof, say $n > 10^3$.

Consider the decision version of the problem that asks whether an undirected graph $G(V, E)$ contains a $\frac{1}{2}$-edge disruptor of a specified size:

$$\frac{1}{2}\text{-ED} = \{\langle G, K\rangle \mid G \text{ has a } \frac{1}{2}\text{-edge disruptor of size } K\}$$

To show that $\frac{1}{2}$-ED is in NP-complete we must show that it is in NP and that all NP-problems are polynomial time reducible to it. The first part is easy; given a candidate subset of edges, we can easily check in polynomial time if it is a β-edge disruptor of size K. To prove the second part, we show that f-balanced cut is polynomial time reducible to $\frac{1}{2}$-ED where $f = \lfloor\frac{n - \sqrt{2\lfloor\frac{n^2}{3}\rfloor + n}}{2}\rfloor$.

Let $G(V, E)$ be a graph in which one seeks to find a f-balanced cut of size k. Now construct the following graph $H(V_H, E_H)$: $V_H = V' \cup C_1 \cup C_2$ where $V' = \{\ v_i \mid i \in V\ \}$ and C_1, C_2 are two cliques of size $\lfloor\frac{n^2}{3}\rfloor$. The total number of nodes in H is, hence, $N = 2\lfloor\frac{n^2}{3}\rfloor + n$. Beside edges inside two cliques we put an edge (v_i, v_j) for each edge $(i, j) \in E$. In addition, connect each vertex v_i to $\lfloor\frac{n^2}{4}\rfloor + 1$ vertices in C_1 and $\lfloor\frac{n^2}{4}\rfloor + 1$ vertices in C_2 so that degree difference of nodes in the cliques are at most one. We illustrate the construction of $H(V_H, E_H)$ in Figure 2. We show that there is a f-balanced cut of size k in G iff H has an $\frac{1}{2}$-edge disruptor of size $K = n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k$ where $0 \le k \le \lfloor\frac{n^2}{4}\rfloor$. Note that the cost of any cut $\langle S, V \backslash S\rangle$ in G is at most $|S||V \backslash S| \le \lfloor\frac{(|S| + |V \backslash S|)^2}{4}\rfloor = \lfloor\frac{n^2}{4}\rfloor$.

In one hand, an f-balanced cut $\langle S, \bar{S}\rangle$ of size k in G induces a cut $\langle C_1 \cup S, C_2 \cup \bar{S}\rangle$ with size exactly $n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k$.

If we select the cut as the disruptor, the pairwise connectivity will be at most $\frac{1}{2}\binom{N}{2}$.

On the other hand, assume that H has an $\frac{1}{2}$-edge disruptor of size $K = n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k$. Because separating n nodes in a clique from the other nodes in that clique requires cutting at least $n(\lfloor\frac{n^2}{3}\rfloor - n) > n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k$ edges. Hence, there are at most n nodes separated from both the cliques. A direct consequence is that at least $(\lfloor\frac{n^2}{3}\rfloor - n)$ nodes remain connected in each clique after removing edges in the disruptor. Denote the sets of those nodes by C_1' and C_2' respectively. C_1' and C_2' cannot be connected otherwise the pairwise connectivity will exceed $\frac{1}{2}\binom{N}{2}$. Denote by X_1, X_2 the set of nodes in V' that are connected to C_1' and C_2' respectively. Since, C_1' and C_2' are disconnected we must have $X_1 \cap X_2 = \phi$.

We will modify the disruptor without increasing its size and the pairwise connectivity such that no nodes in the the cliques are split. For each $u \in C_1 \backslash C_1'$ remove from the disruptor all edges connecting u to C_1' and add to the disruptor all edges connecting u to X_2. This will move u back to the connected component that contains C_1' while reducing the size of the disruptor at least $(\lfloor\frac{n^2}{3}\rfloor - n) - n$. At the same time select an arbitrary node $v \in X_1$ and add to the disruptor all remaining v's adjacent edges. This increases the size of the disruptor at most $(\lfloor\frac{n^2}{4}\rfloor + 1) + n$ while making v isolated. By doing so we decrease the size of the disruptor by $(\lfloor\frac{n^2}{3}\rfloor - n) - n - ((\lfloor\frac{n^2}{4}\rfloor + 1) + n) > 0$. In addition, the pairwise connectivity will not increase when we connect u to C_1' at the same time with disconnecting v from C_1'. If no nodes left in X_1 we can select $v \in X_2$ as in that case $|C_2' \cup X_2| > |C_1' \cup X_1|$ that makes sure the pairwise connectivity will not increase. We repeat the same process for every node in $C_2 \backslash C_2'$ also and note that $|(C_1 \backslash C_1') \cup (C_2 \backslash C_2')| < n$ as proved in previous paragraph. At the end of the process $C_1' = C_1$ and $C_2' = C_2$.

We will prove that $X_1 \cup X_2 = V'$ i.e. $\langle X_1, X_2\rangle$ induces a cut in $V(G)$. Assume not, the cost to separate $C_1 \cup X_1$ from $C_2 \cup X_2$ will be at least $(\lfloor\frac{n^2}{4}\rfloor + 1)(|V' - X_1| + |V' - X_2|) = (\lfloor\frac{n^2}{4}\rfloor + 1)(2n - |X_1| - |X_2|) \geq (\lfloor\frac{n^2}{4}\rfloor + 1)(n+1) > n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k$ that is a contradiction.

Since $X_1 \cup X_2 = V'$ we have the disruptor induces a cut in G. To have the pairwise connectivity at most $\frac{1}{2}\binom{N}{2}$ both $(C_1 \cup X_1)$ and $(C_2 \cup X_2)$ must have size at least $\frac{N - \sqrt{N}}{2}$. If follows that X_1 and X_2 must have size at least $f(n) = \lfloor\frac{n - \sqrt{2\lfloor\frac{n^2}{3}\rfloor + n}}{2}\rfloor$. The cost of the cut induced by $\langle X_1, X_2\rangle$ in G will be $n\left(\lfloor\frac{n^2}{4}\rfloor + 1\right) + k - n(\lfloor\frac{n^2}{4}\rfloor + 1) = k$. ■

B. Hardness of β-vertex disruptor

Theorem 2: β-vertex disruptor in undirected graph is NP-complete.

Proof: We ignore the details and present here the sketch of the proof. We show that vertex cover is polynomial time reducible to β-vertex disruptor. Recall that a vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vetex of the set. The problem of finding a minimum vertex cover is an NP-hard optimization problem.

Let $G(V, E)$ be a graph in which one seeks to find a vertex cover of size k. Note that if we remove nodes in a vertex cover from the graph, the pairwise connectivity in the graph will be zero. Hence, by setting $\beta = 0$, G has a vertex cover of size k iff G has an β-vertex disruptor of size k.

One can also avoid using $\beta = 0$ by replacing each vertex in G by a clique of large enough sizes, say $O(n)$, that ensures no vertices in cliques will be selected in the β-vertex disruptor. ■

Given the NP-hardness of the β-vertex disruptor, the best possible result is a polynomial time approximation scheme (PTAS) that given a parameter $\epsilon > 0$, produces a $(1+\epsilon)$-approximation solution in polynomial time. Unfortunately, such a scheme does not exist for β-vertex disruptor, unless P = NP.

Theorem 3: Unless P = NP, β-vertex disruptor has no polynomial time approximation scheme.

Proof: In the case $\beta = 0$, the problem is equivalent to finding the minimum vertex cover in the graph. In [16], Dinur and Safra showed that approximating vertex cover within constant factor less than 1.36 is NP-hard. Hence, a PTAS scheme for β-vertex disruptor does not exist unless P = NP. ■

IV. Approximating β-edge Disruptor using Tree Decomposition

In this section, we present an $O(\log^{\frac{3}{2}} n)$ pseudo-approximation algorithm for the β-edge disruptor problem in the case when all edges have uniform cost i.e. $c(u, v) = 1\ \forall(u, v) \in E(G)$. Formally, our algorithm finds in an uniform directed graph G a β'-edge disruptor whose the cost is at most $O(\log^{\frac{3}{2}} n)\text{OPT}_{\beta-ED}$, where $\frac{\beta'}{4} < \beta < \beta'$ and $\text{OPT}_{\beta-ED}$ is the cost of an optimal β-edge disruptor.

As shown in Algorithm 1, the proposal algorithm consists of two main steps. First, we constructs a decomposition tree of G by recursively partition the graph into two halves with directed c-balanced cut. Second, we solve the problem on the obtained tree using a dynamic programming algorithm and transfer this solution to the original graph. These two main steps are explained in the next two sections.

A. Balanced Tree-Decomposition

A tree decomposition of a graph is a recursive partitioning of the node set into smaller and smaller pieces until each piece contains only one single node. We show the tree construction in Algorithm 1 (line 1 to 11). Our decomposition tree is a rooted binary tree whose leaves represent nodes in G. (Because our decomposition tree is a binary tree with n leaves, it will contains exactly $n-1$ non-leaf nodes. One can prove this with induction on number of nodes.).

Definition 4: Given a directed graph $G(V, E)$ and a subset of vertices $S \subset V$. We denote the set of edges outgoing from S by $\delta^+(S)$; the set of edges incoming to S by $\delta^-(S)$. A cut $(S, V\backslash S)$ in G is defined as $\delta^+(S)$. A c-balanced cut is a cut $(S, V\backslash S)$ s.t. $\min\{|S|, |V\backslash S|\} \geq c|V|$. The directed

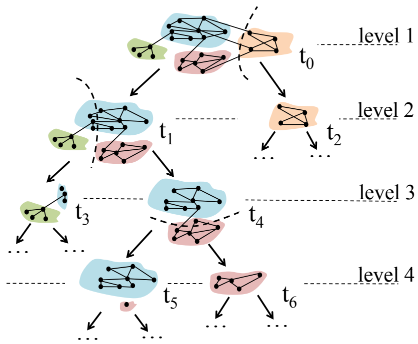


Fig. 3. A part of a decomposition tree. $F = \{t_2, t_3, t_5, t_6\}$ is a G-paritionable. The corresponding partition $\{V(t_2), V(t_3), V(t_5), V(t_6)\}$ in G can be obtained by using cuts at ancestors of nodes in F i.e. t_0, t_1, t_4.

c-balanced cut problem is to find a c-balanced cut of the minimum size.

Note that a cut $(S, V \backslash S)$ separate pairs $(u, v) \in S \times (V \backslash S)$ as paths from v to u cannot exist i.e. no SCC can contain vertex in both S and $V \backslash S$.

The decomposition procedure is as follows. We start with the tree T containing only one root node t_0. We associate the root node t_0 with the vertex set V of G i.e. $V(t_0) = V(G)$. For each node $t_i \in T$ whose $V(t_i)$ contains more than one vertex and $V(t_i)$ has not been partitioned, we partition the subgraph $G[V(t_i)]$ induced by $V(t_i)$ in G using a c-balanced cut algorithm. In detail, we use the directed c-balanced cut algorithm presented in [17] that finds in polynomial time a c'-balanced cut within a factor of $O(\sqrt{\log n})$ from the the optimal c-balanced cut for $c' = \alpha c$ and fixed constant α. The constant c is chosen to be $1 - \sqrt{\frac{\beta}{\beta'}}$. Create two child nodes t_{i1}, t_{i2} of t_i in T corresponding to two sets of vertices of $G[V(t_i)]$ separated by the cut. We associate with t_i a cut cost $cost(t_i)$ equal to the cost of the cut.

We define the root node t_0 to be on level 1. If the a node is on level l, all its children are defined to be on level $l + 1$. Note that all collections of vertices corresponding to nodes in a same level forms a partition in V i.e. they are pairwise disjoint sets of V.

One important parameter of the decomposition tree is the height i.e. the maximum level of nodes in T. Using balanced cuts guarantees a small height of the tree that in turn leads to a small approximation ratio. When separating $V(t_i)$ using the balanced cut, the size of the larger part is at most $(1 - c')|V(t_i)|$. Hence, we can prove by induction that if a node t_i is on level k, the size of the corresponding collection $V(t_i)$ is at most $|V| \times (1 - c')^{k-1}$. It follows that the tree's height is at most $O(-\log_{1-c'} n)$.

B. Algorithm

In this section, we present the second main step which uses the dynamic programming to search for the right set of nodes in T such that the cuts to partition those corresponding sets of vertices in G have the minimum cost and the obtained pairwise connectivity is at most $\beta' \binom{n}{2}$. The details of this step are shown in Algorithm 1 (lines 12 to 18).

Denote a set $F = \{t_{u_1}, t_{u_2}, \ldots, t_{u_k}\} \subset V_T$ where V_T is the set of vertices in T so that $V(t_{u_1}), V(t_{u_2}), \ldots, V(t_{u_k})$ is a partition of $V(G)$ i.e. $V(G) = \biguplus_{h=1}^{k} V_{u_h}$. We say such a subset F is G-partionable. Denote by $\mathcal{A}(t_i)$ the set of ancestors of t_i in T and $\mathcal{A}(F) = \bigcup_{t_i \in F} \mathcal{A}(t_i)$. It is clear that a F is G-partionable iff F satisfies:

1) $\forall t_i, t_j \in F : t_i \notin \mathcal{A}(t_j)$ and $t_j \notin \mathcal{A}(t_i)$
2) $\forall t_i \in V_T, t_i$ is a leaf: $\mathcal{A}(t_i) \cap F \neq \phi$

In case F is G-partitionable, we can separate $V(t_{u_1}), V(t_{u_2}), \ldots, V(t_{u_k})$ in G by performing the cuts corresponding to ancestors of node in F during the tree construction. For example in Figure 3, we show a decomposition tree with a G-partitionable set $\{t_2, t_3, t_5, t_6\}$. The corresponding partition $\{V(t_2), V(t_3), V(t_5), V(t_6)\}$ in G can be obtained by cutting $V(t_0), V(t_1), V(t_4)$ successively using balanced cuts in the tree construction. The cut cost, hence, will be $cost(t_0) + cost(t_1) + cost(t_4)$. In general, the total cost of all the cuts to separate $V(t_{u_1}), V(t_{u_2}), \ldots, V(t_{u_k})$ will be:

$$cost(F) = \sum_{t_u \in \mathcal{A}(F)} cost(t_u)$$

The pairwise connectivity in G then will be:

$$\mathcal{P}(F) = \sum_{t_u \in F} \mathcal{P}(G[V(t_u)])$$

Of course we want to find F so that $\mathcal{P}(F) \leq \beta' \binom{n}{2}$ i.e. the union of cuts to separate $V(t_{u_1}), V(t_{u_2}), \ldots, V(t_{u_k})$ forms a β'-edge disruptor in G. Because of the suboptimal structure in T, finding such a G-partitionable subset F in V_T with minimum $cost(F)$ can be done in $O(n^3)$ using dynamic programming as described netx.

Denote $cost(t_i, p)$ the minimum cut cost to make the pairwise connectivity in $G[V(t_i)]$ equal to p using only cuts corresponding to nodes in the subtree rooted at t_i. The minimum cost for a G-partionable subset F that induces a β'-edge disruptor of G is then $\min\{cost(t_0, p) \mid p \leq \beta' \binom{n}{2}\}$ where t_0 is the root node in T.

The value of $cost(t_i, p)$ can be calculated using following recursive formula:

$$cost(t_i, p) = \begin{cases} 0 & \text{if } \mathcal{P}(G[V(t_i)]) \leq p \\ \min\limits_{p_1 + p_2 = p} cost(t_{i1}, p_1) + cost(t_{i2}, p_2) + cost(t_i) & \text{otherwise} \end{cases}$$

where t_{i1}, t_{i2} are children of t_i.

In the first case, when $\mathcal{P}(G[V(t_i)]) \leq p$ we cut no edges in $G[V(t_i)]$ hence, $cost(t_i, p) = 0$. Otherwise, we try all possible combinations of pairwise connectivity p_1 in $V(t_{i1})$ and p_2 in $V(t_{i2})$ that sum up to p. The combination with the smallest cut cost is then selected.

Using the above recursive formula we can find a minimum cost G-partitionable subset F that induces a β'-edge disruptor in G. We now prove that the cost of the β'-edge disruptor induced from the minimum cost G-partitionable F found in the dynamic programming algorithm is no more than $O(\log^{\frac{3}{2}} n)\mathrm{Opt}_{\beta\text{-ED}}$, where $\mathrm{Opt}_{\beta\text{-ED}}$ denote the cost of the

Algorithm 1. β**-edge Disruptor**

Input: Uniform directed graph $G = (V, E)$ and $0 \le \beta < \beta' < 1$

Output: A β'-edge disruptor of G.

/* Construct the decomposition tree */
$c \leftarrow 1 - \sqrt{\frac{\beta}{\beta'}}$.
$T(V_T, E_T) \leftarrow (\{t_0\}, \phi)$, $V(t_0) \leftarrow V(G)$, $l(t_0) = 1$
while $\exists$ unvisited t_i with $|V(t_i)| \ge 2$ **do**
    Mark t_i visited, create new child nodes t_{i1}, t_{i2} of t_i.
    $l(t_{i1}), l(t_{i2}) \leftarrow l(t_i) + 1$.
    $V_T \leftarrow V_T \cup \{t_{i1}, t_{i2}\}$
    $E_T \leftarrow E_T \cup \{(t_i, t_{i1}), (t_i, t_{i2})\}$
    Separate $G[V(t_i)]$ into two using directed c-balanced cut.
    Assign two obtained partitions to $V(t_{i1}), V(t_{i2})$
    $cost(t_i) \leftarrow$ The cost of the balanced cut
end while
/* Find the minimum cost G-partitionable */
for $t_i \in T$ in reversed BFS order from root node t_0 **do**
    for $p \leftarrow 0$ **to** $\beta' \binom{n}{2}$
        if $\mathcal{P}(G[V(t_i)]) \le p$ **then**
            $cost(t_i, p) \leftarrow 0$
        else
            $cost(t_i, p) \leftarrow \min\{cost(t_{i1}, p_1) + cost(t_{i2}, p_2) + cost(t_i) \mid p_1 + p_2 = p\}$
Find F with $\mathcal{P}(F) = \min\{cost(t_0, p) \mid p \le \beta' \binom{n}{2}\}$
Return union of cuts used at $\mathcal{A}(F)$ during tree construction

Algorithm 2. *Find a good G-partitionable subset of T that induces a β'-edge disruptor in G*

1: $X_T \leftarrow \phi$
2: **for** $t_u \in T$ in BFS order from t_0
3: **if** $(\exists C_i \in X_\beta : |V(t_u) \cap C_i| \ge (1-c)|V(t_u)|)$
4: **and** $(\mathcal{A}(t_u) \cap X_T = \phi)$ **then**
5: $X_T \leftarrow X_T \cup \{t_u\}$

optimal β-edge disruptor in G. It follows directly from the following lemma:

Lemma 1: There exists a G-partitionable subset of T that induces a β'-edge disruptor whose cost is no more than $O\left(\log^{\frac{3}{2}} n\right)$ $\text{Opt}_{\beta\text{-ED}}$.

Proof: Denote D_β an optimal β-edge disruptor in G. Removing D_β from G we obtain a set of SCCs, say $\mathcal{C}_\beta = \{C_1, C_2, ..., C_k\}$.

We construct a G-partitionable subset X_T as in the Algorithm 2. We visit nodes in T in a top-down manner i.e. every parent must be visited before its children. This can be done by visiting nodes in Bread First Search (BFS) order from the root node t_0. For each node t_i if there exists some component $C_j \in \mathcal{C}_\beta$ that $V(t_i)$ contains more than $(1-c)|V(t_i)|$ nodes in C_j (all leaves in T satisfies this condition) and no ancestors of t_i have been ever selected into X_T, then we select t_i as a member of X_T.

One can verify that the obtained X_T satisfies two mentioned conditions of a G-partitionable subset. We are going to prove that $\mathcal{P}(X_T) \le \beta' \binom{n}{2}$:

$$\begin{aligned}
\mathcal{P}(X_T) &\le \sum_{t_i \in X_T} \binom{|V(t_i)|}{2} \\
&= \frac{1}{2} \sum_{C_j \in \mathcal{C}} \sum_{|V(t_i) \cap C_j| \ge (1-c)|V(t_i)|} |V(t_i)|^2 - \frac{n}{2} \\
&\le \frac{1}{2} \sum_{C_j \in \mathcal{C}} \left(\sum_{|V(t_i) \cap C_j| \ge \sqrt{\frac{\beta}{\beta'}} |V(t_i)|} |V(t_i)| \right)^2 - \frac{n}{2} \\
&\le \frac{1}{2} \sum_{C_j \in \mathcal{C}} \left(\sqrt{\frac{\beta'}{\beta}} |C_j| \right)^2 - \frac{n}{2} \\
&< \frac{\beta'}{\beta} \frac{1}{2} \left(\sum_{C_j \in \mathcal{C}} |C_j|^2 - n \right) \le \beta' \binom{n}{2}
\end{aligned}$$

Finally we show that $cost(X_T) \le O(\log^{\frac{3}{2}} n)\text{Opt}_{\beta\text{-ED}}$. Let denote by $h(T)$ the height of T and L_T^i the set of nodes at level i in T_G. We have:

$$cost(X_T) = \sum_{i=1}^{h(T)} \sum_{t_u \in \left(L_T^i \cap \mathcal{A}(X_T)\right)} cost(t_u) \quad (1)$$

If $t_u \in \mathcal{A}(X_T)$ then t_u is not selected to X_T. Hence, there exists $C_j \in \mathcal{C}$ so that $|V(t_u) \cap C_j| < (1-c)|V(t_u)|$ (otherwise t_u was selected into X_T as it satisfied the conditions in the line 3, Algorithm 2). To guarantee $c < 1 - c$ we constrain $c < 1/2$ i.e. $\beta > \frac{\beta'}{4}$.

The edges in the optimal β-edge disruptor D_β separate C_j from the other SCCs. Hence, D_β also separates $C_j \cap V(t_u)$ from the $V(t_u) \backslash C_j$ in $G[V(t_u)]$. Denote $sep(t_u, D_\beta)$ the set of edges in D_β separating $C_j \cap V(t_u)$ from the rest in $G[V(t_u)]$. Obviously, $sep(t_u, D_\beta)$ is a directed c-balanced cut of $G[V(t_u)]$. Since, the cut we used in the tree construction is only $O(\sqrt{\log n})$ times the optimal c-balanced cut. We have $cost(t_u) \le O(\sqrt{\log n})|sep(t_u, D_\beta)|$.

Recall that if two nodes t_u, t_v are on a same level then $V(t_u)$ and $V(t_v)$ are disjoint subsets. It follows that $sep(t_u, D_\beta)$ and $sep(t_v, D_\beta)$ are also disjoint sets. Therefore, we have:

$$\begin{aligned}
\sum_{t_u \in \left(L_T^i \cap \mathcal{A}(X_T)\right)} cost(t_u) &\le O(\sqrt{\log n}) \sum_{t_u \in \left(L_T^i \cap \mathcal{A}(X_T)\right)} |sep(t_u, D_\beta)| \\
&\le O(\sqrt{\log n}) \Big| \bigcup_{t_u \in \left(L_T^i \cap \mathcal{A}(X_T)\right)} sep(t_u, D_\beta) \Big| \\
&= O(\sqrt{\log n}) Opt_{\beta\text{-ED}}
\end{aligned}$$

Since $h(T)$ is at most $O(\log n)$, it follows from the Equation 1 that $cost(X_T) \le O(\log^{\frac{3}{2}} n) Opt_{\beta\text{-ED}}$. This yields the proof. ∎

Since there exists a G-partitionable subset of T that induces a β'-edge disruptor whose cost is no more than $O\left(\log^{\frac{3}{2}} n\right)$ $\text{Opt}_{\beta\text{-ED}}$ as shown in Lemma 1 and the dynamic programming is always able to find such a set F, the following Theorem follows immediately.

Theorem 4: Algorithm 1 achieves a pseudo-approximation ratio of $O(\log^{\frac{3}{2}} n)$ for the β-edges disruptor problem.

V. β-VERTEX DISRUPTOR

We present a polynomial time algorithm that finds a β'-vertex disruptor in the directed graph $G(V, E)$ whose the size is at most $O(\log n \log\log n)$ times the optimal β-vertex disruptor where $0 < \beta < \beta'^2$. The algorithm involves in two phases. In the first phase, we split each vertex $v \in V$ into two vertices v^+ and v^- while putting an edge from v^- to v^+ and show that removing v in G has same effects as removing edge $(v^+ \to v^-)$ in the new graph. In the second phase, we try to decompose the new graph into SCCs capping the sizes of the largest component while minimizing the number of removed edges. The constraints on the size of each component is kept relaxing until the set of cut edges induces a β'-vertex disruptor in the original graph G. We summarize our algorithm in Algorithm 3.

Given a directed graph $G(V, E)$ for which we want to find a small β'-vertex disruptor, we split each vertex in G into two new vertices to obtain a new directed graph $G^{'}(V^{'}, E')$ where

$$\begin{aligned} V' &= \{\, v^-, v^+ \mid v \in V\} \\ E' &= \{(v^- \to v^+) \mid v \in V\} \\ &\quad \cup \{(u^+ \to v^-) \mid (u \to v) \in E\} \end{aligned}$$

The new graph $G'(V', E')$ will have twice the number of vertices in G i.e. $|V'| = 2|V| = 2n$. An example for the first phase is shown in Figure 4.

We set the costs of all edges in $E'_V = \{(v^- \to v^+) \mid v \in V\}$ to 1 and other edges in E' to $+\infty$ so that only edges in E'_V can be selected in an edge disruptor set. In implementation, it is safe to set the costs of edges not in E'_V to $O(n)$ noting that by paying a cost of $2n$ we can effectively disconnect all edges in E'_V and make $\mathcal{P}(G') = 0$.

Consider a directed edge disruptor set $D'_e \subset E'$ that contains only edge in E'_V. We have a one-to-one correspondence between D'_e to a set $D_v = \{v \mid (v^- \to v^+) \in D'_e\}$ in $G(V, E)$ which is a vertex disruptor set in G. Since G and G' have different maximum pairwise connectivity, $\frac{(n-1)n}{2}$ for G and $\frac{(2n-1)2n}{2}$ for G', the fractions of pairwise connectivity remaining in G and G' after removing D_v and D'_e are, however, not simply related to each other.

In the second phase of Algorithm 3, we observe that if we remove edges and separate the graph into SCCs then there is a correlation between the pairwise connectivity in the remaining graph and the maximum size of SCC. The smaller the sizes of SCCs, the smaller pairwise connectivity in the graph. However, the smaller the maximum size of each SCC, the more edges are needed to be cut. Hence, we perform binary search to find a right upper bound for size of each SCC in G'. In the algorithm, the lower bound and upper bound of the size of each SCC are $\underline{\beta}|V'|$ and $\overline{\beta}|V'|$ respectively. At each step we try to find a minimum capacity edge set in $G'(V', E')$ whose removal partitions the graph into strongly connected components of size at most $\tilde{\beta}|V'|$, where $\tilde{\beta} = \lfloor \frac{\underline{\beta} + \overline{\beta}}{2\epsilon} \rfloor \times \epsilon$. We round the value of $\tilde{\beta}$ to the nearest multiple of ϵ so that the number of steps for the binary search is bounded by $\log \frac{1}{\epsilon}$. The problem of finding a minimum capacity edge set to decompose a graph of size n into strongly connected components of size at most ρn is known as ρ-separator problem. We use here the algorithm presented in [18] that for a fixed $\epsilon > 0$ finds a ρ-separator in directed graph G whose value is at most $O\left(\frac{1}{\epsilon^2}.\log n \log\log n\right)$ times $Opt_{(\rho-\epsilon)}$-separator where $Opt_{(\rho-\epsilon)}$-separator is the cost of the optimal $(\rho - \epsilon)$-separator. Finally, we convert the cut edges in G' to vertices in G to obtain the β'-vertex disruptor.

Lemma 2: Algorithm 3 always terminates with a β'-vertex disruptor.

Proof: We show that whenever $\tilde{\beta} \le \beta'$ then the corresponding D_v found in Algorithm 3 is a β'-vertex disruptor in G. Consider the edge disruptor D'_e in G' induced by D_v. We first show the mapping between SCCs in $G[V \backslash D_v]$ and SCCs in $G'[E' \backslash D'_e]$, the graph obtained by removing D'_e from G'. Partition the vertex set V of G into: (1) D_v: the set of removed nodes (2) V_{single}: the set of nodes that are not in any cylcle i.e. they are SCCs of size one (3)$V_{connected}$: union of remaining SCCs that sizes are at least two, say $V_{connected} = \biguplus_{i=1}^{l} C_i, |C_i| \ge 2$. Vertices in $V_{connected}$ belong to at least one cycle in G.

We have following corresponding SCCs in $G'[E' \backslash D'_e]$:

1) $v \in D_v \leftrightarrow$ SCCs $\{v^+\}$ and $\{v^-\}$. Since after removing $(v^- \to v^+)$ v^+ does not have incoming edges and v^- does not have outgoing edges.
2) $v \in V_{single} \leftrightarrow$ SCCs $\{v^+\}$ and $\{v^-\}$. Since v does not lie on any cycle in G. Assume v^+ belong to some SCC of size at least 2 i.e. v^+ lies on some cycle in G'. Because the only incoming edge to v^+ is from v^-. It follows that v^- is preceding $v+$ on that cycle. Let u^-, u^+ be the successive vertices of v^+ on that cycle. We have u and v belong to a same SCC in G which yields a contradiction. Similarly, v^- cannot lie on any cycle in G'.
3) SCC $C_i \subset V_{connected} \leftrightarrow$ SCC $C'_i = \{v^-, v^+ \mid v \in C_i\}$. This can be shown using a similar argument to that in the case $v \in V_{single}$.

Since D'_e is a $\tilde{\beta}$-separator, the sizes of SCCs in $G'[E' \backslash D'_e]$ are at most $\tilde{\beta}\, 2n$. It follows that the sizes of SCCs in $G[V \backslash D_v]$ are bounded by $\tilde{\beta}n$. Denote the set of SCCs in $G[V \backslash D_v]$ by $\mathcal{C}$ with the convention that vertices in D_v become singleton SCCs in $G[V \backslash D_v]$. Therefore, we have:

$$\begin{aligned} \mathcal{P}(G[V \backslash D_v]) = \sum_{C_i \in \mathcal{C}} \binom{|C_i|}{2} &= \frac{1}{2}\left(\sum_{C_i \in \mathcal{C}} |C_i|^2 - |V|\right) \\ &\le \frac{1}{2}\left(\sum_{C_i \in \mathcal{C}} \tilde{\beta}|V|)|C_i| - |V|\right) \\ = \frac{1}{2}\left(\tilde{\beta}|V|^2 - |V|\right) &\le \tilde{\beta}\binom{|V|}{2} < \beta'\binom{|V|}{2} \end{aligned}$$

This guarantees that the binary search always finds a β'-vertex disruptor and completes the proof. ∎

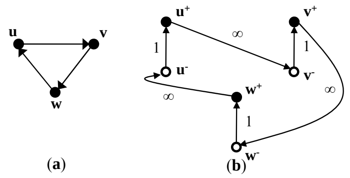


Fig. 4. Conversion from the node version in a directed graph (a) into the edge version in a directed graph (b)

Theorem 5: Algorithm 3 always finds a β'-vertex disruptor whose the size is at most $O(\log n \log\log n)$ times the optimal β-vertex disruptor for $\beta'^2 > \beta > 0$.

Proof: It follows from the Lemma 2 that Algorithm 3 terminates with a β'-vertex disruptor D_v. At some step the capacity of D_v equals to the capacity of $\tilde{\beta}$-separator D'_e in G' where $\tilde{\beta}$ is at least $\beta' - \epsilon$ according to Lemma 2 and the binary search scheme. The cost of the separator is at most $O(\log n \log\log n)$ times the $Opt_{(\tilde{\beta}-\epsilon)}$-separator using the algorithm in [18].

Consider an optimal $(\beta'^2 - 9\epsilon)$-vertex disruptor D'_v of G and its corresponding edge disruptor D'_e in G'. Denote the cost of that optimal vertex disruptor by $Opt_{(\beta'^2-9\epsilon)}$-VD. If there exists in $G[V\backslash D_v]$ a SCC C_i so that $|C_i| > (\beta' - 2\epsilon)n$ then $\mathcal{P}(G[V\backslash D_v]) > \frac{1}{2}((\beta'-2\epsilon)n-2)((\beta'-2\epsilon)n-1) > (\beta'^2 - 9\epsilon)\binom{n}{2}$ when $n > \frac{20(\beta'+1)}{\epsilon}$. Hence, every SCCs in $G'[V\backslash D'_v]$ have size at most $(\beta'-2\epsilon)(2n)$ i.e. D'_e is an $(\beta'-2\epsilon)$-separator in G'. It follows that $Opt_{(\beta'^2-9\epsilon)}$-VD $\geq Opt_{(\beta'-2\epsilon)}$-separator in G'.

Because $\tilde{\beta} - \epsilon \geq \beta' - 2\epsilon$, we have $Opt_{(\tilde{\beta}-\epsilon)}$-separator $\leq Opt_{(\beta'-2\epsilon)}$-separator $\leq Opt_{(\beta'^2-9\epsilon)}$-VD.

The size of the vertex disruptor $|D_v| = |D'_e|$ is at most $O(\log n \log\log n)$ times $Opt_{(\tilde{\beta}-\epsilon)}$-separator. Thus, the size of found β'-vertex disruptor D_v is at most $O(\log n \log\log n)$ times the optimal $(\beta'^2 - 9\epsilon)$-vertex disruptor. As we can choose arbitrary small ϵ, setting $\beta = \beta'^2 - 9\epsilon$ completes the proof. ∎

We further investigate the relation between β-edge disruptor and β-vertex disruptor on G and G' as shown next.

A. Approximating edge disruptor is at least as hard as approximating vertex disruptor

We show that an approximation algorithm for general edge disruptor in directed graph also yields an approximation algorithm for directed vertex disruptor with the same approximation ratio.

Lemma 3: A β-edge disruptor set in the directed graph G' induces the same cost β-vertex disruptor set in G.

Proof: We use D_v and D'_e for vertex disruptor in G and edge disruptor in G'.

Given $\mathcal{P}(G'[E'\backslash D'_e]) \leq \beta\binom{2n}{2}$ we need to prove that: $\mathcal{P}(G[V\backslash D_v]) \leq \beta\binom{n}{2}$ where $n = |V|$.

Assume $G[V\backslash D_v]$ has l SCCs of size at least 2, say $C_i, i = 1\ldots l$. The corresponding SCCs in $G'[E'\backslash D'_e]$wil be $C'_i, i = 1\ldots l$ where $|C'_i| = 2|C_i|$.

Since $\frac{\binom{2k}{2}}{\binom{2n}{2}} - \frac{\binom{k}{2}}{\binom{n}{2}} = \frac{k(n-k)}{(n-1)n(2n-1)} \geq 0,\ \forall 0 \leq k \leq n$. We have:

$$\frac{\mathcal{P}(G[V\backslash D_v])}{\binom{n}{2}} = \sum_{i=1}^{l} \frac{\binom{|C_i|}{2}}{\binom{n}{2}} \leq \sum_{i=1}^{l} \frac{\binom{|C'_i|}{2}}{\binom{2n}{2}} \leq \beta$$

Algorithm 3. β'**-vertex disruptor**

Input: Directed graph $G = (V, E)$ and fixed $0 < \beta' < 1$.

Output: A β'-vertex disruptor of G

$G'(V', E') \leftarrow (\phi, \phi)$
$\forall v \in V : V' \leftarrow V' \cup \{v^+, v^-\}$
$\forall v \in V : E' \leftarrow E' \cup \{(v^- \to v^+)\}, c(v^-, v^+) \leftarrow 1$
$\forall (u \to v) \in E : E' \leftarrow E' \cup \{u^+ \to v^-\}, c(u^+, v^-) \leftarrow \infty$
$\underline{\beta} \leftarrow 0, \overline{\beta} \leftarrow 1$
$D_V \leftarrow V(G)$
while $(\overline{\beta} - \underline{\beta} > \epsilon)$ **do**
  $\tilde{\beta} \leftarrow \lfloor \frac{\underline{\beta} + \overline{\beta}}{2\epsilon} \rfloor \times \epsilon$
  Find $D_e \subset E'$ to separate G' into strongly connected components of sizes at most $\tilde{\beta}|V'|$ using algorithm in [18]
  $D_v \leftarrow \{v \in V(G) \mid (v^+ \to v^-) \in D_e\}$
  if $\mathcal{P}(G[V\backslash D_v]) \leq \beta\binom{n}{2}$ **then**
    $\underline{\beta} = \tilde{\beta}$
    Remove nodes from D_v as long as $\mathcal{P}(G[V\backslash D_v]) \leq \beta\binom{n}{2}$
    if $|D_V| > |D_v|$ **then** $D_V = D_v$
  else
    $\overline{\beta} = \tilde{\beta}$
end while
Return D_V

∎

Lemma 4: A β-vertex disruptor set in the directed graph G' induces the same cost $(\beta + \epsilon)$-edge disruptor set in G for fixed $\epsilon > 0$.

Proof: We use the same notations in the proof of Lemma 3. Given $\mathcal{P}(G[V\backslash D_v]) \leq \beta\binom{2n}{2}$ we need to prove that: $\mathcal{P}(G'[E'\backslash D'_e]) \leq (\beta + \epsilon)\binom{n}{2}$. We have:

$$\begin{aligned}\frac{\mathcal{P}(G'[E'\backslash D'_e])}{\binom{n}{2}} &= \sum_{i=1}^{l} \frac{|C_i|(n-|C_i|)}{(n-1)n(2n-1)} + \frac{\mathcal{P}(G[V\backslash D_v])}{\binom{n}{2}} \\ &= \frac{\mathcal{P}(G[V\backslash D_v])}{\binom{n}{2}}\left(1 - \frac{1}{2n-1}\right) + \frac{1}{2n-1} \\ &< \beta + \epsilon\end{aligned}$$

when $n \geq \lfloor \frac{1+\epsilon}{2\epsilon} \rfloor + 1$. ∎

Lemmas 3 and 4 provide a path to approximate the β-vertex disruptor. The fixed ϵ constant in Lemma 4 is inevitable due to the natural of the problem.

VI. Related and Prior Works

The related works to be included in this section consist of three aspects: traditional vulnerability measures, existing critical vertex/edge detections, and related graph partition problems.

The classic vulnerability measures are two-fold: global graph measures and local nodal measures. The former ones are mainly functions of graph properties, e.g., the number of vertices/edges, operational O-D pairs, operational paths, minimum shortest paths [1] [2] [3] [4]. However, some of these attributes cannot be calculated in polynomial-time for dense networks. Plus these functions do not reveal the set of most critical vertices and edges, thus are not suitable to assess the network vulnerability in terms of connectivity. The local nodal measures are mainly based on the centrality of each vertex in the graph, which consist of degree centrality, betweenness, closeness, and eigenvector centrality [5]. However, these measures fail to indicate accurate vulnerabilities and cannot reveal the global damage done on the network under attacks.

Previous works involving with Critical Vertex/Edge defined it as the minimum number of vertices/edges whose removal disconnects the graph, *regardless how disconnected it is*. Several heuristics have been proposed with no approximation ratio guaranteed. In the context of wireless network, there are some DFS-based centralized algorithms [6] [7], pairwise-routing-based distributed algorithms [8] and localized algorithms [9]. Regardless of their high communication overheads, the definition of critical vertex/edge is only limited to elements disconnecting graphs, thus cannot apply this concept to the vulnerability evaluation.

Related to our β-disruptor problem is some works along the line of graph partitioning problems. Specifically, *linear programming relaxation* based on multicommodity flow which was provided by Leighton and Rao [19] obtained an $O(\log n)$ approximation algorithm for SPARSEST CUT and an $O(\log n)$ bicriteria approximation algorithm for c-BALANCED CUT; Arora, Rao, Vazirani [20] improved these results to $O(\sqrt{n})$ using *semidefinite relaxation* with triangle inequality constraints. In addition, *divide and conquer* technique via spreading metrics [21] [18] and *hierarchical decomposition tree* techniques [22] [23] [24] also become major tools for graph approximation algorithms. Meanwhile, many inapproximability results shown in the sequence of papers [25] [26] and [27] related the minimum bisection to the *unique games conjecture*. A variant of the β-vertex disruptor in undirected network is presented in [28] in which the authors seek for a set of k nodes whose removal maximally decreases the connectivity in the graph. However, no approximation results have been developed in the literature for all variants of the problems presented in this paper.

VII. Conclusion

We introduced a novel model and measurement for evaluating the vulnerability of general networks even in the case of networks with unidirectional links. We also formulated new optimization problems corresponding to the model, studied their hardness, and proposed two pseudo-approximation algorithms with provable performance bounds. Being the first quantitative measure directing at global network connectivity, our method provides an accurate vulnerability assessment toward different types of network.

For the future work, we are aiming to bridge the gap in the proposed algorithm of the β-vertex disruptor problem for any value of $\beta \leq \beta'$ and further tightening the ratio, which we conjecture that using the semidefinite relaxation in [20], we may be able to reduce the ratio to $O(\sqrt{n})$.

References

[1] Tony H. Grubesic, Timothy C. Matisziw, Alan T. Murray, and Diane Snediker. Comparative approaches for assessing network vulnerability. *International Regional Science Review*, 31(1):88–112, January 2008.

[2] R. Church, M. Scaparra, and R. Middleton. Identifying critical infrastructure: the median and covering facility interdiction problems. *Ann Assoc Am Geogr*, 94(3):491–502, 2004.

[3] A. Murray, T. Matisziw, and T. Grubesic. Multimethodological approaches to network vulnerability analysis. *Growth Change*, 2008.

[4] E. Jenelius, T. Petersen, and L. Mattsson. Importance and exposure in road network vulnerability analysis. *Transport Res Part A*, pages 40:537–C560, 2005.

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

[6] M. Duque-Anton, F. Bruyaux, and P. Semal. Measuring the survivability of a network: Connectivity and rest-connectivity. *European Transactions on Telecommunications*, 11:149–159, 2000.

[7] D. Goyal and J. Caffery. Partitioning avoidance in mobile ad hoc networks using network survivability concepts. *7th IEEE Symposium on Computers and Communications*, page 553, 2002.

[8] M. Hauspie, J. Carle, and D. Simplot. Partition detection in mobile ad hoc networks using multiple disjoint paths set. *Workshop of Objects, Models and Multimedia technology*, 2003.

[9] M. Jorgic, I. Stojmenovic, M. Hauspie, and D. Simplot-Ryl. Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks. *3rd IFIP MED-HOC-NET Workshop*, 2004.

[10] A. Barabasi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281, 2000.

[11] Y. J. Suh, D. J. Kim, W. S. Lim, and J. Y. Baek. Method for supporting quality of service in heterogeneous networks. 2009.

[12] T. Lehman, J. Sobieski, and B. Jabbari. Dragon: a framework for service provisioning in heterogeneous grid networks. *IEEE Communication Magazines*, 2006.

[13] V. Mhatre and C. Rosenberg. Homogeneous vs heterogeneous clustered sensor networks: a comparative study. *IEEE International Conference on Communication*, 2004.

[14] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of ACM*, 44(4):585–591, 1997.

[15] D. Wagner and F. Wagner. Between min cut and graph bisection. In *MFCS '93: Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, pages 744–750, London, UK, 1993. Springer-Verlag.

[16] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:2005, 2004.

[17] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. O(log n) approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581, New York, NY, USA, 2005. ACM.

[18] G. Even, J. S. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of ACM*, 47(4):585–616, 2000.

[19] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of ACM*, 46(6):787–832, 1999.

[20] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 222–231, New York, NY, USA, 2004. ACM.

[21] G. Even, J. S. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. In *SIAM Journal on Computing*, pages 639–648, 1999.

[22] A. Coja-Oghlan, A. Goerdt, A. Lanka, and F. Schädlich. Techniques from combinatorial approximation algorithms yield efficient algorithms for random 2k-sat. *Theor. Comput. Sci.*, 329(1-3):1–45, 2004.

[23] K. Andreev and H. Räcke. Balanced graph partitioning. *Theorical Compupter System*, 39(6):929–939, 2006.

[24] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 255–264, New York, NY, USA, 2008. ACM.

[25] T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is np-hard. *Information Processing Letters*, 42(3):153 – 159, 1992.

[26] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006.

[27] S. Khot and N. K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l1. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 53–62, Washington, DC, USA, 2005. IEEE Computer Society.

[28] A. Arulselvan, Clayton W. Commander, L. Elefteriadou, and Panos M. Pardalos. Detecting critical nodes in sparse graphs. *Comput. Oper. Res.*, 36(7):2193–2200, 2009.