On New Approaches of Assessing Network Vulnerability: Hardness and Approximation

Thang N. Dinh, Ying Xuan, My T. Thai, *Member, IEEE*, Panos M. Pardalos, *Member, IEEE*, Taieb Znati, *Member, IEEE*

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. Existing vulnerability assessments mainly focus on investigating the inhomogeneous properties of graph elements, node degree for example, however, these measures and the corresponding heuristic solutions can provide neither an accurate evaluation over general network topologies, nor performance guarantees to large scale networks. To this end, in this paper, we investigate a measure called *pairwise connectivity* and formulate this vulnerability assessment problem as a new graph-theoretical optimization problem called β-disruptor, which aims to discover the set of *critical* node/edges, whose removal results in the sharpest decline of the global pairwise connectivity. Our results consist of the NP-Completeness and inapproximability proof of this problem, an $O(\log n \log\log n)$ pseudo-approximation algorithm for detecting the set of *critical* nodes and an $O(\log^{1.5} n)$ pseudo-approximation algorithm for detecting the set of *critical* edges. Finally, we perform extensive simulation to compare our algorithms with the optimal solution found by solving Integer Programming.

Index Terms—Network vulnerability, Pairwise connectivity, Hardness, Approximation algorithm.

1 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], [5], [6]. 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 [7], 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

• T. N. Dinh, Y. Xuan, M. T. Thai are with the Dept. of Comp. & Info. Sci. & Eng., University of Florida, Gainesville, FL, 32611.
E-mail: {tdinh, yxuan, mythai}@cise.ufl.edu.
• Panos M. Pardalos is with Industrial and System Engineering Dept., University of Florida at Gainesville, FL, 32611.
Email: pardalos@ufl.edu.
• T. Znati is with Computer Science Dept., University of Pittsburgh, Pittsburgh, PA 15215.
Email: znati@cs.pitt.edu.

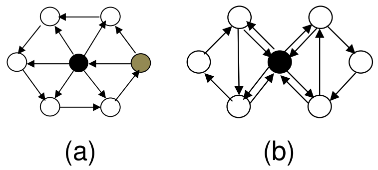

Fig. 1. After the "central" vertex (in black) with maximum out-going degree is removed, network (a) is still strongly connected while (b) is fragmented; however in fact, only removing one vertex (in grey) is enough to destroy network (a).

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.

Consequently, we convert the vulnerability assessment into a graph-theoretical optimization problem: *finding a minimized set of vertices/edges whose removal degrades the pairwise connectivity to a desired degree*. 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 provable performance bounds, which thus ensure the 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 have 7 vertices and are 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 decrease 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 [8][9][10]. 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 [11], 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:

- Providing a novel underlying framework toward the vulnerability assessment by investigating the pairwise connectivity and formulating it as 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*;
- Presenting 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 continue this section with Related works, Definition, Models and Notations. We provide the hardness results in Section 2. The pseudo-approximation algorithms for β*-edge disruptor* and β*-vertex disruptor* are presented in Section 3 and Section 4 respectively. Section 5 presents the simulation results comparing the performance of the proposed approximation algorithms to that of the optimal solution found by solving Integer Programming. Finally, Section 6 summarizes the whole paper.

1.1 Related and Prior Works

The classic vulnerability measurements are mainly based on the centrality of each vertex in the graph, which consist of degree centrality, betweenness, closeness, and eigenvector centrality [7]. However, these measures fail to indicate accurate vulnerabilities and cannot reveal the global damage done on the network under attacks.

On the other hand, the global graph measures 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]. However, some of these attributes cannot be calculated in polynomial-time for dense networks. In essence, 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. Several similar concepts with our pairwise connectivity have been recently investigated in [12], [13], [14], where the terms *average pairwise connectivity*, *pairwise connected ratio* and *cohesion* were used. However, none of them were able to formulate the calculation of this measure as an optimization problem and provide the hardness proof along with performance guaranteed approximation algorithms. Moreover, the problem β-disruptor studied in this paper take into account the roles of all edges and vertices in the global network connectivity, thus provides a more essential research and thorough analysis over the underlying vulnerability framework established.

As a subproblem of this vulnerability assessment problem, Critical Vertex/Edge, which are defined as the minimum number of vertices/edges whose removal disconnects the graph, are also studied and solved using extensive heuristics, however, without performance guarantee. Some work of this kind in the context of wireless network are [8][9][10], nevertheless, these works consider only whether or not the graph is disconnected and ignore how fragmental the graph becomes. They are insufficient to evaluate the graph vulnerability.

1.2 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 [15], [16], [17], 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 $m = |E|$ links among $n = |V|$ 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 have:

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

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 $0 \leq \beta \leq 1$, 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.

2 Hardness Results

In this section we show that both the β-edge disruptor and β-vertex disruptor in directed graph are NP-complete which thus have no 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.

2.1 NP-completeness of β-edge disruptor

We use a reduction from the balanced cut problem.

Definition 2: A cut $\langle S, V \setminus S \rangle$ 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 $V \setminus S$ by $\bar{S}$.

Finding a min cut in the graph is polynomial solvable [18]. 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) An f-balanced cut of a graph $G(V, E)$, where $f : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$, asks us to find a cut $\langle S, \bar{S} \rangle$ with the minimum size such that $|S|, |\bar{S}| \geq f(|V|)$.

Abusing notations, for $0 < c \leq \frac{1}{2}$, we also use c-balanced cut to find the cut $\langle S, \bar{S} \rangle$ with the minimum size such that $\min\{|S|, |\bar{S}|\} \geq c|V|$. We will use the following results on balanced cut shown in [19]:

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

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

Then $f(n) \leq g(n)$ 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 $f = \Omega(\alpha n^{\epsilon})$ f-balanced cut is NP-complete. We are ready to prove the NP-completeness of β-edge disruptor:

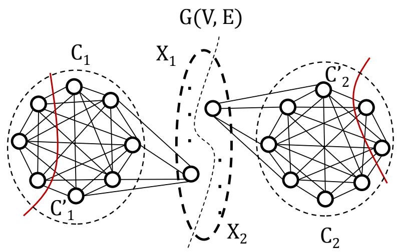

Fig. 2. Construction of $H(V_H, E_H)$ from $G(V, E)$

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 shall assume that n, the number of nodes is a sufficient large number (for our proof $n > 10^3$).

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. Construct the following graph $H(V_H, E_H)$: $V_H = V \cup C_1 \cup C_2$ where C_1, C_2 are two cliques of size $\lfloor \frac{n^2}{3} \rfloor$. Denote by $N = |V_H| = 2\lfloor \frac{n^2}{3} \rfloor + n$ the total number of nodes in H. In addition to edges in G, C_1, and C_2, connect each vertex $v \in V$ 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 $\langle S, V \setminus S\rangle$ in G is at most $|S||V \setminus S| \le \lfloor \frac{(|S|+|V\setminus S|)^2}{4} \rfloor = \lfloor \frac{n^2}{4} \rfloor$.

On 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 $K = n\left(\lfloor \frac{n^2}{4} \rfloor + 1\right) + k$. Remove the edges in the disruptor to reduce the pairwise connectivity to at most $\frac{1}{2}\binom{N}{2}$. Since cutting n nodes in C_1 or C_2 from the cliques requires removing at least $n(\lfloor \frac{n^2}{3} \rfloor - n) > n\left(\lfloor \frac{n^2}{4} \rfloor + 1\right) + k$ edges, let $C_1' \subseteq C_1$ and $C_2' \subseteq C_2$ be giant connected subsets that induce connected subgraphs in C_1 and C_2. These subsets must satisfy $|C_1'| + |C_2'| > |C_1| + |C_2| - n$. Denote by X_1, X_2 the subsets of nodes in V that are connected to C_1' and C_2' respectively. We have $X_1 \cap X_2 = \emptyset$ otherwise C_1' and C_2' will be connected; then, the pairwise connectivity will exceed $\frac{1}{2}\binom{N}{2}$.

We will modify the disruptor without increasing its size and the pairwise connectivity such that no nodes in the the cliques are cut off i.e. we alter the disruptor until $C_1' = C_1$ and $C_2' = C_2$. For each $u \in C_1 \setminus C_1'$ 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 attach u to 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 as we connect u to C_1' and at the same time disconnect v from C_1'.

If $X_1 = \emptyset$, 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 $C_2 \setminus C_2'$. Since $|(C_1 \setminus C_1') \cup (C_2 \setminus C_2')| < n$, the whole process finishes in less than n steps and results in $C_1' = C_1$ 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 G. Assume not, the cost to separate $C_1 \cup X_1$ from $C_2 \cup X_2$ will be at least $(\lfloor \frac{n^2}{4} \rfloor + 1)(|V - X_1| + |V - X_2|) = (\lfloor \frac{n^2}{4} \rfloor + 1)(2n - |X_1| - |X_2|) \ge (\lfloor \frac{n^2}{4} \rfloor + 1)(n+1) > n\left(\lfloor \frac{n^2}{4} \rfloor + 1\right) + k$ that is a contradiction.

Since $X_1 \cup X_2 = V$ we have that 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 $n\left(\lfloor \frac{n^2}{4} \rfloor + 1\right) + k - n(\lfloor \frac{n^2}{4} \rfloor + 1) = k$. □

2.2 Hardness of β-vertex disruptor

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

Proof: We present a polynomial-time reduction from Vertex Cover (VC), an NP-hard problem [20]:

Instance: Given a graph G and a positive integer k.

Question: Does G have a VC of size at most k?

to a decision version of β-vertex disruptor when $\beta = 0$

Instance: Given a graph G and a positive integer k

Question: Does G have a β-vertex disruptor of size at most k when $\beta = 0$?

Pairwise connectivity equals zeros if and only if the complement set of the disruptor is an independent set or in other words the disruptor must be a VC. □

Theorem 3: Unless P = NP, β-vertex disruptor cannot be approximated within a factor of 1.36.

Proof: We use the same reduction in Theorem 2. Assume that we can approximate β-vertex disruptor within a factor less than 1.36 when $\beta = 0$. In [21], Dinur and Safra showed that approximating VC within constant factor less than 1.36 is NP-hard. Since we have an one-to-one mapping between the set of vertex disruptors when $\beta = 0$ and the set of VCs, it follows that we can approximate VC within a factor less than 1.36 (contradiction). □

3 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 a uniform directed graph G a β'-edge disruptor whose the cost is at most $O(\log^{\frac{3}{2}} n)\mathrm{OPT}_{\beta-ED}$, where $\frac{\beta'}{4} < \beta < \beta'$ and $\mathrm{OPT}_{\beta-ED}$ 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 partitioning 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.

3.1 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 contain 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 $(S, V\setminus S)$ in G is defined as $\delta^+(S)$. A c-balanced cut is a cut $(S, V\setminus S)$ s.t. $\min\{|S|, |V\setminus S|\} \geq c|V|$. The directed c-balanced cut problem is to find the minimum c-balanced cut.

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

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 [22] that finds in polynomial time a c'-balanced cut within a factor of $O(\sqrt{\log n})$ from 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 c-balanced cut.

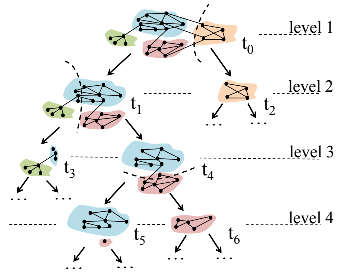

Fig. 3. A part of a decomposition tree. $F = \{t_2, t_3, t_5, t_6\}$ is a G-partitionable. The corresponding partition $\{V(t_2), V(t_3), V(t_5), V(t_6)\}$ in G can be obtained by using cuts at ancestors of nodes in F i.e. t_0, t_1, t_4.

We define the root node t_0 to be on level 1. If a node is on level l, all its children are defined to be on level $l+1$. Note that collections of subsets of vertices in G that correspond to nodes in a same level of T induces a partition in G.

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 $O(-\log_{1-c'} n) = O(\log n)$.

3.2 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 that induces an cost-efficient partition in G whose 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-partitionable. 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-partitionable if and only if F satisfies:

Algorithm 1. β**-edge Disruptor**

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

Output: A β'-edge disruptor of G.

/* Construct the decomposition tree */

1. $c \leftarrow 1 - \sqrt{\frac{\beta}{\beta'}}$.
2. $T(V_T, E_T) \leftarrow (\{t_0\}, \phi)$, $V(t_0) \leftarrow V(G)$, $l(t_0) = 1$
3. **while** $\exists$ unvisited t_i with $|V(t_i)| \ge 2$ **do**
4. Mark t_i visited, create new child nodes t_{i1}, t_{i2} of t_i.
5. $V_T \leftarrow V_T \cup \{t_{i1}, t_{i2}\}$
6. $E_T \leftarrow E_T \cup \{(t_i, t_{i1}), (t_i, t_{i2})\}$
7. Separate $G_{[V(t_i)]}$ using directed c-balanced cut.
8. Associate $V(t_{i1}), V(t_{i2})$ with two separated components.
9. $cost(t_i) \leftarrow$ The cost of the balanced cut

/* Find the minimum cost G-partitionable */

10. Traverse T in post-order, **for** each $t_i \in T$ **do**
11. **for** $p \leftarrow 0$ **to** $\beta'\binom{n}{2}$
12. **if** $\mathcal{P}(G_{[V(t_i)]}) \le p$ **then** $cost(t_i, p) \leftarrow 0$
13. **else** $cost(t_i, p) \leftarrow \min\{cost(t_{i1}, p_1) +$ $cost(t_{i2}, p_2) + cost(t_i) \mid p_1 + p_2 = p\}$
14. Find $F^{\text{opt}}_{\beta'}$ associating with $T^{\text{opt}}_{\beta'} = \min_{p \le \beta'\binom{n}{2}} \{cost(t_0, p)\}$
15. Return union of c-balanced cuts at $t_i \in \mathcal{A}(F^{\text{opt}}_{\beta'})$.

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:

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

We wish to find F so that $\mathcal{P}(F) \le \beta'\binom{n}{2}$ 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.

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-partitionable subset F that induces a β'-edge disruptor of G is then

$$T^{\text{opt}}_{\beta'} = \min_{p \le \beta'\binom{n}{2}} \{cost(t_0, p)\}$$

where t_0 is the root node in T.

We can easily derive the recursive formula:

$$cost(t_i, p) = \begin{cases} 0 & \text{if } \mathcal{P}(G_{[V(t_i)]}) \le p \\ \min\limits_{\pi \le p} cost(t_{i1}, \pi) + cost(t_{i2}, p - \pi) + cost(t_i) & \text{if not} \end{cases}$$

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

In the first case, when $\mathcal{P}(G_{[V(t_i)]}) \le p$ we cut no edges in $G_{[V(t_i)]}$ hence, $cost(t_i, p) = 0$. Otherwise, we try all possible combinations of pairwise connectivity π in $V(t_{i1})$ and $p - \pi$ in $V(t_{i2})$. The combination with the smallest cut cost is then selected.

We now prove that $T^{\text{opt}}_{\beta'} \le O(\log^{\frac{3}{2}} n)\text{Opt}_{\beta\text{-ED}}$, where $\text{Opt}_{\beta\text{-ED}}$ denotes the cost of the optimal β-edge disruptor in G.

Lemma 1: There exists a G-partitionable subset of T that induces a β'-edge disruptor whose cost is at most $O\left(\log^{\frac{3}{2}} n\right) \text{Opt}_{\beta\text{-ED}}$.

Proof: Let D_β be an optimal β-edge disruptor in G of size $\text{Opt}_{\beta\text{-ED}}$ and $\mathcal{C}_\beta = \{C_1, C_2, ..., C_k\}$ be the set of SCCs, after removing D_β from G.

We construct a G-partitionable subset X_T as in the Algorithm 2. We traverse tree T in preorder i.e. every parent will be visited before its children. For each node t_i, we select t_i into X_T if there exists some component $C_j \in \mathcal{C}_\beta$ that $|V(t_i) \cap C_j| \ge (1-c)|V(t_i)|$ and no ancestors of t_i have been selected into X_T. We can verify that X_T satisfies two mentioned conditions of a G-partitionable subset. For each $C_j \in \mathcal{C}_\beta$, define

$$N(C_j) = \{t_i \in T : |V(t_i) \cap C_j| \ge (1-c)|V(t_i)|\}.$$

Since $V(t_i), t_i \in T$ are disjoint subsets. We have

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

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^i_T the set of nodes at the ith level in T_G. We have:

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

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

Initialization: $X_T \leftarrow \phi$; `Preorder-Selection(`t_0`)`.
`Preorder-Selection(`t_u`)`
1: **if** $(\exists C_j \in \mathcal{C}_\beta : |V(t_u) \cap C_j| \geq (1-c)|V(t_u)|)$ **then**
2: $\quad X_T \leftarrow X_T \cup \{t_u\}$
3: **else** let t_{u1}, t_{u2} be children of t_u,
4: $\quad$ `Preorder-Selection(`t_{u1}`)`
5: $\quad$ `Preorder-Selection(`t_{u2}`)`
6: **end if**

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 need $c < 1/2$ i.e. $\beta > \frac{\beta'}{4}$.

Since the edges in D_β separate C_j from the other SCCs, they also separates $C_j \cap V(t_u)$ from $V(t_u) \setminus C_j$ in $G_{[V(t_u)]}$. Denote by $\delta(t_u, D_\beta)$ the set of edges in D_β separating $C_j \cap V(t_u)$ from $V(t_u) \setminus C_j$ in $G_{[V(t_u)]}$. Obviously, $\delta(t_u, D_\beta)$ 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 $cost(t_u) \leq O(\sqrt{\log n})|\delta(t_u, D_\beta)|$.

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 $\delta(t_u, D_\beta)$ and $\delta(t_v, D_\beta)$ are also disjoint sets. Therefore, the cut cost at the ith level

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

Since the number of levels $h(T) = O(\log n)$, by Eq. 1 we have $cost(X_T) \leq O(\log^{\frac{3}{2}} n)\mathrm{Opt}_{\beta\text{-ED}}$. □

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

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

Time complexity: Construction of the decomposition tree takes $O(n^{9.5})$. The major portion of time is for solving an semidefinite programming with $\Omega(n^3)$ constraints. Finding the optimal solution using Dynamic Programming takes $O(n^3)$. Hence, the overall time complexity is $O(n^{9.5})$.

4 β-VERTEX DISRUPTOR

We present a polynomial time algorithm (Algorithm 3) 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 the 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. We relax the constraints on the size of each component until the set of cut edges induces a β'-vertex disruptor in the original graph G.

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.

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 exactly equal to each other.

In the second phase of Algorithm 3, when separating a graph into SCCs, 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 to be cut. 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 denoted by $\underline{\beta}|V'|$ 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

Algorithm 3. β'**-vertex disruptor**
Input: Directed graph $G = (V, E)$ and fixed $0 < \beta' < 1$.
Output: A β'-vertex disruptor of G

$G'(V', E') \leftarrow (\phi, \phi)$
$\forall v \in V : V' \leftarrow V' \cup \{v^+, v^-\}$
$\forall v \in V : E' \leftarrow E' \cup \{(v^- \rightarrow v^+)\}, c(v^-, v^+) \leftarrow 1$
$\forall (u \rightarrow v) \in E : E' \leftarrow E' \cup \{u^+ \rightarrow v^-\}, c(u^+, v^-) \leftarrow \infty$
$\underline{\beta} \leftarrow 0, \overline{\beta} \leftarrow 1$
$D_V \leftarrow V(G)$
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 [23]
  $D_v \leftarrow \{v \in V(G) \mid (v^+ \rightarrow v^-) \in D_e\}$
  if $\mathcal{P}(G_{[V \setminus D_v]}) \leq \beta\binom{n}{2}$ **then**
    $\underline{\beta} = \tilde{\beta}$
    Remove nodes from D_v as long as $\mathcal{P}(G_{[V \setminus D_v]}) \leq \beta\binom{n}{2}$
    if $|D_V| > |D_v|$ **then** $D_V = D_v$
  else $\overline{\beta} = \tilde{\beta}$
end while
Return D_V

presented in [23] 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 $\text{Opt}_{(\rho-\epsilon)}$-separator where $\text{Opt}_{(\rho-\epsilon)}$-separator is the cost of the optimal $(\rho - \epsilon)$-separator. Finally, we derive the cut vertices in G from the cut edges in G' to obtain the β'-vertex disruptor.

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

Proof: We show that whenever $\tilde{\beta} \leq \beta'$ 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 $G_{[V \setminus D_v]}$ and SCCs in $G'[E' \setminus D'_e]$, 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 $V_{connected} = \biguplus_{i=1}^{l} C_i, |C_i| \geq 2$. Vertices in $V_{connected}$ belong to at least one cycle in G.

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

1) $v \in D_v \leftrightarrow$ SCCs $\{v^+\}$ and $\{v^-\}$. Since after removing $(v^- \rightarrow v^+)$ 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 $G'[E' \setminus D'_e]$ are at most $\tilde{\beta}\ 2n$. It follows that the sizes of SCCs in $G_{[V \setminus D_v]}$ are bounded by $\tilde{\beta}n$. Denote the set of SCCs in $G_{[V \setminus D_v]}$ by $\mathcal{C}$ with the convention that vertices in D_v become singleton SCC in $G_{[V \setminus D_v]}$. Therefore, we have:

$$\mathcal{P}(G_{[V \setminus D_v]}) = \sum_{C_i \in \mathcal{C}} \binom{|C_i|}{2} = \frac{1}{2}\left(\sum_{C_i \in \mathcal{C}} |C_i|^2 - |V|\right)$$
$$\leq \frac{1}{2}\left(\sum_{C_i \in \mathcal{C}} \tilde{\beta}|V|)|C_i| - |V|\right)$$
$$= \frac{1}{2}\left(\tilde{\beta}|V|^2 - |V|\right) \leq \tilde{\beta}\binom{|V|}{2} < \beta'\binom{|V|}{2}$$

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

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 $\text{Opt}_{(\tilde{\beta}-\epsilon)}$-separator using the algorithm in [23].

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 $\text{Opt}_{(\beta'^2-9\epsilon)}$-VD. If there exists in $G_{[V \setminus D_v]}$ a SCC C_i so that $|C_i| > (\beta' - 2\epsilon)n$ then $\mathcal{P}(G_{[V \setminus D_v]}) > \frac{1}{2}((\beta' - 2\epsilon)n - 2)((\beta' - 2\epsilon)n - 1) > (\beta'^2 - 9\epsilon)\binom{n}{2}$ when $n > \frac{20(\beta'+1)}{\epsilon}$. Hence, every SCC in $G'_{[V \setminus D'_v]}$ have size at most $(\beta' - 2\epsilon)(2n)$ i.e. D'_e is an $(\beta' - 2\epsilon)$-separator in G'. It follows that $\text{Opt}_{(\beta'^2-9\epsilon)}$-VD $\geq \text{Opt}_{(\beta'-2\epsilon)}$-separator in G'.

Since $\tilde{\beta} - \epsilon \geq \beta' - 2\epsilon$, we have $\text{Opt}_{(\tilde{\beta}-\epsilon)}$-separator $\leq \text{Opt}_{(\beta'-2\epsilon)}$-separator $\leq \text{Opt}_{(\beta'^2-9\epsilon)}$-VD.

The size of the vertex disruptor $|D_v| = |D'_e|$ is at most $O(\log n \log\log n)$ times $\text{Opt}_{(\tilde{\beta}-\epsilon)}$-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. □

Time complexity: Finding the separator costs $O(n^9)$ [23]. Hence, the total time complexity is $O(\log \frac{1}{\epsilon} n^9)$. However, in our experiments, the algorithm takes much less than its worst-case running time.

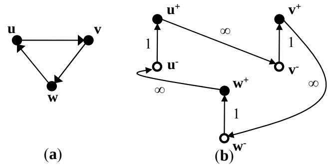

Fig. 4. Conversion from the node version in a directed graph (a) into the edge version in a directed graph (b)

4.1 Approximating edge disruptor is at least as hard as approximating vertex disruptor

We show that an approximation algorithm for general directed edge disruptor yields an approximation algorithm for directed vertex disruptor with (almost) 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 $\mathcal{P}(G'[E'\backslash D'_e]) \le \beta\binom{2n}{2}$ we need to prove that: $\mathcal{P}(G_{[V\backslash D_v]}) \le \beta\binom{n}{2}$ where $n = |V|$.

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

Since $\frac{\binom{2k}{2}}{\binom{2n}{2}} - \frac{\binom{k}{2}}{\binom{n}{2}} = \frac{k(n-k)}{(n-1)n(2n-1)} \ge 0$, for all $0 \le k \le n$. We have

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

□

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

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

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

when $n \ge \lfloor \frac{1+\epsilon}{2\epsilon} \rfloor + 1$. □

Theorem 6: Given a factor $f(n)$ polynomial time approximation algorithm for β-edge disruptor, there exists a factor $(1+\epsilon)f(n)$ polynomial time approximation algorithm for β-vertex disruptor where $\epsilon > 0$ is an arbitrary small constant.

Proof: Let G be a directed graph with uniform vertex costs in which we wish to find a β-vertex disruptor. Construct G' as described at the beginning of this Section.

Apply the given approximation algorithm to find in G' a β-edge disruptor, denoted by D'_e, with the cost at most $f(n) \cdot \mathrm{Opt}_{\beta-\mathrm{ED}}(G')$, where $\mathrm{Opt}_{\beta-\mathrm{ED}}(G')$ is the cost of a minimum β-edge disruptor in G'. From Lemma 3, D'_e induces in G a β-vertex disruptor D_v of the same cost. We shall prove that

$$\mathrm{Opt}_{\beta-\mathrm{ED}}(G') \le \mathrm{Opt}_{\beta-\mathrm{VD}}(G) + \gamma_0,$$

where $\mathrm{Opt}_{\beta-\mathrm{VD}}(G)$ is the cost of a minimum β-vertex disruptor in G and γ_0 is some positive constant. It follows that the cost of D_v will be at most

$$f(n) \cdot (\mathrm{Opt}_{\beta-\mathrm{VD}}(G) + \lambda_0) \le (1+\epsilon)f(n)\mathrm{Opt}_{\beta-\mathrm{VD}}(G)$$

Here, we assume that $\mathrm{Opt}_{\beta-\mathrm{VD}}(G) > \frac{\gamma_0}{\epsilon}$ otherwise we can find $\mathrm{Opt}_{\beta-\mathrm{VD}}(G)$ in time $O(n^{\frac{\gamma_0}{\epsilon}+2})$.

From an optimal β-vertex disruptor of G, construct its corresponding edge disruptor D^*_e in G'. If $\mathcal{P}(G'[E \setminus D^*_e] \le \beta\binom{2n}{2}$ then $\mathrm{Opt}_{\beta-\mathrm{ED}}(G') \le \mathrm{Opt}_{\beta-\mathrm{VD}}(G)$ and we yield the proof. Thus, we consider the case $\mathcal{P}(G'[E \setminus D^*_e] > \beta\binom{2n}{2}$.

Among SCCs of $G'[E \setminus D^*_e]$, there must be a SCC of size at least $\beta 2n$ or else $G'[E \setminus D^*_e] \le \beta^{-1}\binom{\beta 2n}{2} \le \beta\binom{2n}{2}$ (contradiction). Remove $\gamma_0 = \left\lceil \frac{1}{\beta} \right\rceil$ vertices from that SCC. The pairwise connectivity in $G'[E \setminus D^*_e]$ will decrease at least $(\beta 2n - \frac{1}{\beta})\frac{1}{\beta} = 2n - \frac{1}{\beta^2} \ge n$ for sufficient large n. From Eq. 2 in Lemma 4, the pairwise connectivity after removing vertices will be less than

$$(\beta + \frac{1}{2n-1})\binom{2n}{2} - n \le \beta\binom{2n}{2}$$

Therefore, after removing at most γ_0 vertices from D^*_e, we get a β-edge disruptor. Hence,

$$\mathrm{Opt}_{\beta-\mathrm{ED}}(G') \le \mathrm{Opt}_{\beta-\mathrm{VD}}(G) + \gamma_0.$$

□

5 EXPERIMENTAL STUDY

We perform experiments to find out the gap between the solution of the pseudo approximation algorithm (Algorithm 3) and an optimal solution found by solving an Integer programming formulation. We generate two types of network: random networks following Erdos-Rényi model and power-law networks following Barabási-Albert model. For each type of network, we generate different instances with number of nodes ranging from 30 to 100. Edge densities of generated networks are around 10%. The machine used for the experiments was an 8 cores 2.2 Ghz equipped with 64 GB memory.

Size of disruptors found by Algorithm 3 and the size of optimal disruptors are presented in Tables 1 and 2. Despite a large theoretical gap of the pseudo approximation algorithm, the algorithm produces near-optimal solutions and returning optimal solutions in more than half places (marked with bold numbers).

Especially, our algorithm performs extremely well on power-law networks. It misses the optimal solution in only one place when the number of vertices is 90. Between a random network and a power-law

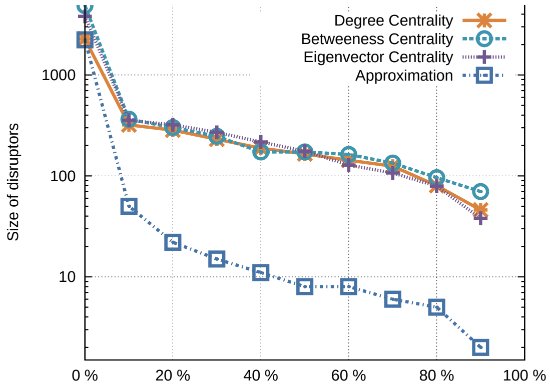

Fig. 5. Disruptors found by different methods in the Western States Power Grid of the United States at different levels of disruption.

network of roughly same sizes, the size of disruptor in the power-law network is significantly smaller (approximately 50%) than that in the random network, showing extremely high degree of vulnerability of power-law network to attacks [24].

TABLE 1
Size of disruptor on Erdos-Rényi networks at 60% connectivity.

Vertex	30	40	50	60	70	80	90	100
Edge	43	78	122	177	241	316	400	495
Optimal	2	4	7	9	11	12	16	18
Approx	3	**4**	8	**9**	**11**	13	**16**	19

TABLE 2
Size of disruptor on Barabási–Albert networks at 60% connectivity.

Vertex	30	40	50	60	70	80	90	100
Edge	54	131	189	208	245	262	354	445
Optimal	1	3	5	6	6	5	7	9
Approx	**1**	**3**	**5**	**6**	**6**	**5**	10	**9**

The running time for solving the Integer programming increases from few minutes to 10 hours for the largest test cases, while in the longest run, the pseudo-approximation algorithm takes only 29 seconds.

5.1 Case study: Western States Power Grid

We study a network of 4941 nodes and 6594 edges representing the topology of the Western States Power Grid of the United States. The network is shown to be high clustering with small characteristics path lengths [25]; hence the network is rather vulnerable to targeted attacks.

It is intractable to find the optimal disruptor using Integer Programming for such a large network. Our approximation algorithm uses row-generation technique to reduce excessive amount of constraints and runs on a clusters of 20 nodes, each node is equipped with an 8 cores 2.2 Ghz CPU and 64 GB memory.

We compare the attack schemes that target nodes based on their centrality with our pseudo approximation algorithm to show that those methods might not be suitable to reveal network vulnerability in term of overall network connectivity. Compared methods include

1) *Degree Centrality*: The algorithm sequentially remove node with the maximum degree until the pairwise connectivity in the graph less than $\beta\binom{n}{2}$.
2) *Betweenness Centrality*: We repeatedly remove the node with maximum betweenness centrality, until the pairwise connectivity in the graph less than $\beta\binom{n}{2}$. Recall that the betweenness $Bt(v)$ for node v is: $Bt(v) = \sum_{\substack{s \neq v \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}$ where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a node v.
3) *Eigenvector Centrality*: Nodes are removed in descending order of their Eigenvector centrality (Pagerank) values with the default damping factor of 85% as in [26].

We show in Figure 5 vulnerability reported by different methods at various levels of disruption. The network is surprisingly vulnerable to targeted attacks. For example to reduce 40% connectivity in the network (60% connectivity remain) we only need to destroy 0.16% stations. Bringing down the connectivity to the same level, the average number of nodes to remove for random networks and power-law networks are 13% and 3% respectively. Even destroying only 1% of stations can dramatically disrupt 90% connectivity in the network.

None of other methods can reveal correctly the vulnerability of the power grid. Their disruptor sizes are 6 to 20 times larger than those of our approximation algorithm. Thus, using alternative assessment methods rather than the ones we proposed might lead to a dangerous mirage that the network is strongly stable.

Because of high clustering property, nodes that lie among clusters in the networks will often have high betweenness values. Intuitively, we expected the betweenness method to easily identify those nodes and perform well in the experiment. Surprisingly, the performance of betweenness method turns out to be even worse than that of degree centrality.

6 Conclusion

We established a novel model to assess the vulnerability by investigating *how many nodes/edges are required to be deleted in order to bring down the network pairwise connectivity to a desired extent*. After formulating this problem as an optimization problem called β-disruptor, we presented several hardness results including the NP-Completeness and inapproximability, along with two pseudo-approximation algorithms with provable performance bounds. The accuracy of our framework compared with existing measurements are validated through a series of experiments on both simulated and real networks. Providing an underlying framework toward the vulnerability assessment over general network topology and performance guaranteed solutions, our method exhibits huge benefits and potentials for various practical network situations.

REFERENCES

[1] Tony H. Grubesic, Timothy C. Matisziw, Alan T. Murray, and Diane Snediker. Comparative approaches for assessing network vulnerability. *Inter. Regional Sci. Review*, 31, 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] A. Sen, S. Murthy, and S. Banerjee. Region-based connectivity - a new paradigm for design of fault-tolerant networks. In *HPSR*, 2009.

[5] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. In *INFOCOM*, 2009.

[6] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, Inc., New York, NY, USA, 1987.

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

[8] D. Goyal and J. Caffery. Partitioning avoidance in mobile ad hoc networks using network survivability concepts. *ISCC*, page 553, 2002.

[9] 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.

[10] 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.

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

[12] Fangting Sun and Mark A. Shayman. On pairwise connectivity of wireless multihop networks. *International Journal of Security and Networks*, 2(1/2):37–49, 2007.

[13] A. Arulselvan, Clayton W. Commander, L. Elefteriadou, and Panos M. Pardalos. Detecting critical nodes in sparse graphs. *Computers & Operations Research*, 36(7):2193–2200, 2009.

[14] Stephen P. Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

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

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

[17] V. Mhatre and C. Rosenberg. Homogeneous vs heterogeneous clustered sensor networks: a comparative study. *IEEE ICC*, 2004.

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

[19] D. Wagner and F. Wagner. Between min cut and graph bisection. In *MFCS*, pages 744–750, London, UK, 1993. Springer-Verlag.

[20] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

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

[22] 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*, pages 573–581, New York, NY, USA, 2005. ACM.

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

[24] R. Albert, H. Jeong, and A. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[25] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Thang N. Dinh received the BA degree in Information Technology from Vietnam National University, Hanoi, Vietnam in 2007. He is currently a PhD student at the Department of Computer and Information Science and Engineering, University of Florida, under the supervision of Dr. My T. Thai. His research focuses on designing combinatorial optimization methods for dynamic complex networks and mobile ad hoc network including network vulnerability, dynamic community structure, and fast information propagation.

Ying Xuan received the BE degree in computer engineering from the University of Science and Technology of China, Anhui, China, in 2006. He is now a PhD candidate at the Department of Computer and Information Science and Engineering, University of Florida, under the supervision of Dr. My T. Thai. His research topics include applied group testing theory, social networking and network vulnerability.

My T. Thai received her PhD degree in computer science from the University of Minnesota, Twin Cities, in 2006. She is an associate professor in the Department of Computer and Information Sciences and Engineering at the University of Florida. Her current research interests include algorithms and optimization on network science and engineering. She also serves as an associate editor for the Journal of Combinatorial Optimization (JOCO) and Optimization Letters and a conference chair of COCOON 2010 and several workshops in an area of network science. She is a recipient of DoD Young Investigator Awards and NSF CAREER awards. She is a member of the IEEE.

Panos M. Pardalos is a Distinguished Professor of Industrial and Systems Engineering at the University of Florida. He is the director of the Center for Applied Optimization. Dr. Pardalos obtained a PhD degree from the University of Minnesota in Computer and Information Sciences. Dr. Pardalos is the editor-in-chief of the "Journal of Global Optimization," and of the journals "Optimization Letters," "Computational Management Science," and "Energy Systems."

Taieb Znati Taieb Znati received the MS degree in computer science from Purdue University in 1984, and the PhD degree in computer science from Michigan State University in 1988. He is a professor in the Department of Computer Science, with a joint appointment in Telecommunications in the Department of Information Science, University of Pittsburgh. He currently serves as the director of the Computer and Network Systems (CNS) Division at the National Science Foundation (NSF). From 2000 to 2004, he served as a senior program director for networking research at NSF. He also served as the committee chairperson of the Information Technology Research (ITR) Program and an NSF-wide research initiative in information technology. His current research interests are on network science and engineering, with the focus on the design of scalable, robust, and reliable network architectures and protocols for wired and wireless communication networks. He is a recipient of several research grants from government agencies and from industry. He is frequently invited to present keynotes in networking and distributed conferences both in the United States and abroad. He is a member of the IEEE.