Technical Report: An Exact Approach and Bounding Techniques for Network Vulnerability Assessment

Thang N. Dinh and My T. Thai

Computer & Information Science & Engineering
University of Florida, Gainesville, FL, 32611,
Email: {tdinh, mythai}@cise.ufl.edu

Abstract. We study the problem of deleting a minimum number of nodes or links in a network to disrupt the network connectivity to a certain level. Formally, we find a minimum set of edges/vertices, called β-edge(vertex) disruptor whose removal lessens the number of connected pairs in the network to at most $\beta\binom{n}{2}$, where $0 \leq \beta \leq 1$ is a constant and n is the number of nodes in the network.

We provide an spectral method to give an efficiently lower bound for the size of a β-edge disruptor and an upper bound via a $O(\sqrt{\log n})$ bicriteria approximation algorithm. We also introduce a mathematical programming approach to find optimal β-vertex disruptor for networks of hundreds of nodes. The extension of the exact solution to β-edge disruptor and other variants of the problem is straightforward.

1 Introduction

Identifying nodes and edges that removal dramatically reduces the connectivity in networks is of great importance. The study of critical infrastructure in network will help plan the allocation of resources during the evacuation, reestablish critical routes in the aftermath of a disaster, as well as predict the network responses.

In this paper, we study the β-disruptor problem, formulated in [1], in which the goal is to locate a minimum set of edges (nodes) to remove so that the pairwise connectivity falls down to a certain level. The β-disruptor problem takes 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.

Finding β-disruptor is to find the most cost-effective way to consolidate or destroy the network and it turns out to be a quite challenging problem. For any computable function $\alpha(n)$, the problem of deleting k nodes to minimize remaining connectivity cannot be approximated within a factor $\alpha(n)$, unless P=NP. The proof involves a simple reduction from the independent set problem. Bissias [2] was first to study the problem of removing ω edges from the network to minimize the sum of squares of components' sizes. The critical node detection problem that study the questions how to remove k edges/nodes to maximize network disruption were studied in [3]. In that paper, the authors gives NP-completeness for the problem and find exact solution in sparse graphs using integer programming. For asynchronous networks (directed graphs), we present in [1] the NP-hardness of β-disruptor problems and $O(\log^{1.5}|V|)$ bicriteria approximation algorithms for both vertex and edge versions.

Our contributions. The main contributions of this paper are as follows:

- We provide spectral bounds to effectively estimate the size of the optimal edge disruptor. Instead of solving the NP-hard Quadratic Constrained Quadratic Programming with integral constraints, we provide a dynamic programming solutions for computing the spectral bound. Computational experiments with the spectral bound were done earlier in [2,4,5] shows that the spectral bound is often within a small constant times the optimal solution and is useful for analyzing networks' characteristics.
- A simple bicriteria approximation algorithm that finds a β-edge disruptor with the cost at most $\frac{1}{\beta-\beta'}O(\sqrt{\log|V|})$ the cost of the optimal β'-edge disruptor for any $0 \le \beta' < \beta$.
- Devise a new mixed-integer programming and *compressed metric* technique to find the exact solution in networks with hundreds of nodes. The technique can be applied for variations of disruptor problems and numerous of connectivity-related problems such as multicut, graph partitioning, etc.

2 Model and Definitions

Let $G = (V, E)$ be a undirected graph given by the weighted adjacency matrix $A(G) = [c_{uv}]$. Denote by $D(G)$ the diagonal matrix of G with vertex degree on the diagonal, i.e. $d_{vv} = d_v$, and $d_{uv} = 0$ if $u \neq v$. The difference

$$L(G) = D(G) - A(G)$$

is called *Laplacian matrix* of G.

The matrix $L(G)$ is real symmetric, so it has $n = |V|$ real eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ and $\lambda_1 = 0$ with a corresponding eigenvector $\mathbb{1} = (1, 1, \ldots, 1)^T$.

We study on the aggregate pairwise connectivity between all pairs, that is the number of connected pairs and denoted as $\mathcal{P}(G)$. Apparently $\mathcal{P}(G)$ is maximized at $\binom{n}{2}$ when G is a connected graph.

For subsets X and Y of V, denote $\langle X, Y\rangle$ the set of edges that link a node in X to a node in Y, and $\phi(X, Y)$ the total weight of the edges in $\langle X, Y\rangle$. For any cut $\langle S, \bar{S}\rangle$, where $\bar{S} = V \setminus S$, the *sparsity* (edge expansion) of the cut is $\alpha(S) = \frac{\phi(S,\bar{S})}{|S||V\setminus S|}$.

Definition 1 (β-edge disruptor). *Given $0 \le \beta \le 1$, 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 $\beta\binom{n}{2}$. In the β-edge disruptor problem, we find a minimum cost β-edge disruptor in a connected graph $G(V, E)$.*

The vertex version of the problem is called *β-vertex disruptor.*

3 Bounding Disruptors with Network Spectrum

Theorem 1. *Let $\phi(\mathrm{OPT}_\beta)$ be the total capacity of the optimal β-edge disruptor OPT_β of G, then*

$$\frac{1-\beta}{2}\lambda_2(G)(n-1) \le \phi(\mathrm{OPT}_\beta) \le \frac{1-\beta}{2}\lambda_n(G)(n-1).$$

Proof. Assume that an optimal edge disruptor separates the graph into connected components $C_1, C_2, \dots, C_k$. We must have $\sum_i |C_i|^2 = 2\sum_i \binom{|C_i|}{2} - n \le 2\beta\binom{n}{2} - n$.

Number of edges in the optimal disruptor.

$$\phi(\mathrm{OPT}_\beta) = \frac{1}{2}\sum_{i=1}^{k}\phi(C_i, \overline{C_i}) = \frac{1}{2}\sum_{i=1}^{k}\alpha(C_i)|C_i|(n-|C_i|). \tag{1}$$

The value of $\alpha(C_i)$ is bounded between the second smallest eigenvalue and the largest eigenvalue of $L(G)$ [6] that is $\lambda_2 \le n\ \alpha(S) \le \lambda_n$. Therefore

$$\phi(\mathrm{OPT}_\beta) \ge \frac{1}{2}\sum_{i=1}^{k}\frac{1}{n}\lambda_2|C_i|(n-|C_i|) = \frac{1}{2n}\lambda_2\left(n\sum_{i=1}^{k}|C_i| - \sum_{i=1}^{k}|C_i|^2\right)$$
$$= \frac{1}{n}\lambda_2\left(\binom{n}{2} - \sum_{i=1}^{k}\binom{|C_i|}{2}\right) \ge \frac{1}{n}\lambda_2(1-\beta)\binom{n}{2} = \frac{1-\beta}{2}\lambda_2(n-1).$$

Similarly, we also obtain the other part of the inequality. □

3.1 Bounding Disruptor Size with Network Spectrum

We begin with an elegant result in graph partitioning by Donath and Hoffman.

Lemma 1. *[4] Let a k-partition of a graph be a division of the vertices into k disjoint subsets containing $m_1 \ge m_2 \ge \dots \ge m_k$ vertices. Let E_{cut} be the set of edges whose two vertices belong to different subsets. Let $\lambda_1 \le \lambda_2 \le \dots \le \lambda_k$, be the k smallest eigenvalues of the Laplacian matrix plus any diagonal matrix U such that the sum of all the elements of U is zero. Then*

$$|E_{\mathrm{cut}}| \ge \frac{1}{2}\sum_{i=1}^{k}m_i\lambda_i.$$

Assume that we always have $k = n$ partitions with the allowance of zero-size subsets. Bissias et al. [2] based on Lemma 1 give a lower bound for the damage caused by removing at most k edges from the network. This can be done by relaxing the integrality constraints of m_i then solving a quadratic programming that checks for all eligible configurations of $\{m_i\}_i$.

We extend the method for β-edge disruptor problem in several aspects. First, we do not relax integrality constraints on m_i, thus, tighten the bound. Second, we avoid tackling the formulated Quadratic Constrained Quadratic Programming (QCQP) directly and be able to devise an effective dynamic programming solution. Notice that in general solving QCQP is an NP-hard problem even when variables are not restricted to be integers.

$$\text{minimize} \quad \frac{1}{2}\sum_{i=1}^{n} m_i \lambda_i \tag{2}$$

$$\text{subject to} \quad \mathbb{1}^T m = n \tag{3}$$

$$\sum_{i=1}^{n} \binom{m_i}{2} = \beta \binom{n}{2} \tag{4}$$

$$m_i \in [0..n], \quad i \in [1..n]. \tag{5}$$

Theorem 2. *Let Q_β be the optimal objective of the* QCQP *(2-5) and* OPT_β *be the minimum β-edge disruptor of graph $G = (V, E)$. Then, $Q_\beta \le \phi(\mathrm{OPT}_\beta)$ for all $0 \le \beta \le 1$. Moreover, the equality holds at $\beta = 0$ and $\beta = 1$*

Proof. Obviously, the sizes of subsets in the optimal β-edge disruptor satisfy all constraints (3-5). Hence, $Q_\beta \le \phi(\mathrm{OPT}_\beta)$ for all $0 \le \beta \le 1$.

In case $\beta = 0$, all subsets are of size one. Hence, $Q_1 = \frac{1}{2}\sum_{i=1}^{n} \lambda_i = \frac{1}{2}\mathrm{Trace}(L + U) = \frac{1}{2}(2|E| + \mathrm{Trace}\, U) = |E|$. The only way to cut all pairs in the network is to cut all edges. In other words, $Q_0 = |\mathrm{OPT}_0| = |E|$.

In case $\beta = 1$, in order to achieve the maximum connectivity $\binom{n}{2}$, there must be a single partition in the network and the optimal disruptor cutting no edges. Hence, $m_1 = n$ and $m_i = 0\ \forall i > 1$. Since $\lambda_1 = 0$, it follows $Q_1 = 0 = |E_{\mathrm{cut}}|$. □

To devise a dynamic programming solution for the bound, we define

$$\mathcal{L}_k(l, p) = \min_{m_1 + m_2 + \ldots + m_k = l} \left\{ \sum_{i=1}^{k} \lambda_i m_i : \sum_{i=1}^{k} \binom{m_i}{2} \le p \right\},$$

the minimum spectral bound obtained by first k subsets whose sizes sum up to l and the total pairwise connectivity is at most p. The solution of the QCQP shall be given by $\mathcal{L}_n(n, \beta\binom{n}{2})$.

The following lemma allows us to pay attention to only partitions whose sizes of subsets m satisfying $m_1 \ge m_2 \ge \ldots \ge m_n$.

Lemma 2. *There exists an optimal solution m of* QCQP*(2-5) such that $m_1 \ge m_2 \ge \ldots \ge m_n$.*

Proof. Let $m^* = \{m_1^*, m_2^*, \ldots, m_n^*\}$ be an optimal solution of QCQP(2-5). Denote $\mathrm{inv}(m^*)$ the number of inversions of $m*$ i.e. such pairs of indices (i, j) that $i < j$ and $m_i^* > m_j^*$. If $\mathrm{inv}(m^*) = 0$, then $m_1^* \ge m_2^*, \ge \ldots \ge m_n^*$, otherwise there exists a pair $i < j$ and $m_i^* > m_j^*$. Construct m' by swapping m_i^* and m_j^* inside

m^*. Then, m' is a feasible solution of QCQP(2-5) and the objective increases an amount $m_i^*\lambda_j + m_j^*\lambda_i - (m_i^*\lambda_i + m_j^*\lambda_j) = (m_i^* - m_j^*)(\lambda_j - \lambda_i) \geq 0$. Thus, we obtain a new optimal solution with less the number of inversions. Repeat the process at most $\binom{n}{2}$, that is the maximum number of inversions in m^*, we finally obtain an optimal solution with no inversions. That optimal solution shall satisfy the lemma's condition. □

The choice of the smallest subset. In order to compute $\mathcal{L}_p(l,k)$, we identify the possible values of m_k. Since $m_1 \geq m_2 \geq \ldots \geq m_k$, m_k can only obtain values between 0 and $\lfloor \frac{l}{k} \rfloor$. Let $m' = (m_1 - m_k, m_2 - m_k, \ldots, m_{k-1} - m_k)$ that are subsets' sizes in some network partition with following properties

1. The partition consists of $k' = k - 1$ subsets.
2. The total size of all subsets is $l' = l - km_k$.
3. The total pairwise connectivity is

$$\sum_{i=1}^{k} \binom{m_i - m_k}{2} = \sum_{i=1}^{k} \left[\binom{m_i}{2} + \binom{m_k + 1}{2} - m_i m_k \right]$$
$$= \sum_{i=1}^{k} \binom{m_i}{2} + k\binom{m_k + 1}{2} - m_k \sum_{i=1}^{k} m_i \leq p - m_k l + k\binom{m_k + 1}{2}$$

Therefore, the pairwise connectivity of m' is a function of k, l, m_k and pairwise connectivity of m.

We compute value of $\mathcal{L}_p(l,k)$ in increasing order of p and l but in decreasing order of k. The recursive formula for $\mathcal{L}_p(l,k)$ is as follow.

$$\mathcal{L}_k(l,p) = \begin{cases} +\infty, & \text{if } p < p_{\min}(l,k) \\ \lambda_1 l = 0, & \text{if } p \geq p_{\max}(l,k) \\ \min\limits_{m_k = 1..\lfloor \frac{l}{k} \rfloor} \left\{ \mathcal{L}_{k-1}\left(l - km_k, p - m_k l + k\binom{m_k+1}{2}\right) + m_k \sum_{i=1}^{k} \lambda_k \right\}, & \text{otherwise} \end{cases} \quad (6)$$

where $p_{\min}(l,k) = \binom{\lceil l/k \rceil}{2}(l \bmod k) + \binom{\lfloor l/k \rfloor}{2}(k - l \bmod k)$ and $p_{\max}(l,k) = \binom{l}{2}$ that are the minimum and maximum pairwise connectivity of a graph with l vertices and k connected components, respectively.

Theorem 3. *Optimal solutions of* QCQP *(2-5) can be found in $O(n^4 \log n)$ time.*

Proof. It is straightforward to prove optimal substructure of the problem and that recursive formula (6) leads to an optimal solution. Since calculation of $\mathcal{L}_k(l,p)$ takes $\frac{l}{k}$ time, the total time complexity is bounded by

$$\sum_{p=1}^{\binom{n}{2}} \sum_{l=1}^{n} \sum_{k=1}^{l} \frac{1}{k} l = \sum_{p=1}^{\binom{n}{2}} \sum_{l=1}^{n} O(l \log l) = \sum_{p=1}^{\binom{n}{2}} O(n^2 \log n) = O(n^4 \log n).$$

In practice, the computing time is much less than $O(n^4 \log n)$ with classical memorization technique [7]. □

3.2 Bicriteria Upper Bound via Iterative Connective Cuts

We analyze a greedy approach in which we delete at each step a subset of most "cost-effective" edges that is the one with minimum ratio between the total cost and number of separated pairs.

We show that by iteratively approximating the sparsest cut, we obtain a $O(\sqrt{\log n})$ bicriteria approximation algorithm for β-edge disruptor, where $n = |V(G)|$. We first generalize the notion of cut to handle disconnected graphs.

Definition 2 (Connective cut). *For any disjoint subsets $S, T \subset V$, $\langle S, T\rangle$ define a connective cut in G if and only if the following conditions hold.*

1. *$\alpha(S \cup T) = 0$ i.e. there are no edges coming out from $S \cup T$.*
2. *$G[S]$ and $G[T]$, the subgraphs induced by S, T in G are connected subgraphs.*

Denote $\alpha_c(S, T) = \frac{\phi(S,T)}{|S||T|}$ the *sparsity* of connective cut $\langle S, T\rangle$. In the SPARSEST CUT and the SPARSEST CONNECTIVE CUT problems, we wish to determine, respectively, the cut/connective cut with the smallest edge expansions

$$\begin{aligned}\alpha(G) &= \min\{\alpha(S) : S \subset V\},\\ \alpha_c(G) &= \min\{\alpha_c(S, T) : \text{ connective cut } \langle S, T\rangle\}.\end{aligned}$$

Proposition 1. *A connective cut $\langle S, T\rangle$ disconnects exactly $|S||T|$ pairs in G.*

Proposition 2. *If G is connected, then $\alpha(G) = \alpha_c(G)$.*

Lemma 3. *Given a polynomial time $f(n)$ approximation algorithm for* SPARSEST CUT*, we can design a polynomial time to approximate* SPARSEST CONNECTIVE CUT *problem with the same approximation ratio $f(n)$, where $n = |V(G)|$.*

Proof. If G is disconnected, we can approximate the sparsest connective cut within each connected component in G and select the connective cut with the smallest edge expansion. Thus, we reduce the problem to the case G is connected. If G is connected, for any cut $\langle S, \bar{S}\rangle$ which induces $k \geq 3$ connected components, there must be one component that can be separated from the rest with a connective cut of sparsity at most $\alpha(S)$. Thus, designing sparsest-cut based algorithm to approximate the sparsest connective cut is straightforward. □

Theorem 4. *[8] There is a $\tilde{O}(n^2))$ algorithm that finds a cut $\langle S, \bar{S}\rangle$ in $G = (V, E)$ of sparsity at most $O(\sqrt{\log n}\ \alpha(G))$, where $\tilde{O}(\cdot)$ notation is used to suppress polylogarithmic factors.*

Corollary 1. *Given a graph $G = (V, E)$, there is a $\tilde{O}(n^2))$ algorithm that finds a connective cut $\langle S, T\rangle$ in $G = (V, E)$ of sparsity at most $\sqrt{\log n}\ \alpha_c(S)$.*

Lemma 4. *Disrupting K pairs in G requires cutting at least $\alpha_c(G)K$ edges.*

Proof. Assume the network is connected. Otherwise we can apply the proof on each connected components separately and yield the same result.

Let $\mathcal{C}_K$ be a minimum set of edges that reduce the connectivity in G by K. Assume that C separates the graph into connected components $C_1, C_2, \ldots, C_k$. Use the same approach in Theorem 1, we have the size of the minimum solution

$$\sum_{(u,v)\in\mathcal{C}_K} c_{uv} = \frac{1}{2}\sum_{i=1}^{k}\alpha(C_i)|C_i|(n-|C_i|) \leq \frac{1}{2}\sum_{i=1}^{k}\alpha(G)|C_i|(n-|C_i|) \leq \alpha(G)K.$$

Since G is connected, we have $\alpha(G)K = \alpha_c(G)K$. □

Proposition 3. *If $\beta' < \beta$, then $\phi(\mathrm{OPT}_{\beta'}) \geq \phi(\mathrm{OPT}_{\beta})$.*

We present a $O(\sqrt{\log n})$ bicriteria approximation algorithm for β-edge disruptor in Algorithm 1.

Algorithm 1. Find the β-disruptor $G = (V, E)$

$\mathcal{D} = \emptyset, PWC = \binom{n}{2}$
while $PWC > \beta\binom{n}{2}$ do
  Find a connective cut (S, T) as in Lemma 3
  $\mathcal{D} = \mathcal{D} \cup \langle S, T\rangle$
  $PWC = PWC - |S||T|$
end while
Output $\mathcal{D}$

Theorem 5. *For any $0 \leq \beta' < \beta$, Algorithm 1 returns a β-edge disruptor of capacity at most $O(\sqrt{\log n})\phi(\mathrm{OPT}_{\beta'})$, where $\mathrm{OPT}_{\beta'}$ is the optimal β-edge disruptor. Moreover, the running time is at most $\tilde{O}(n^2)$.*

Proof. At a particular step when $PWC > \beta\binom{n}{2}$, $\mathrm{OPT}_{\beta'}$ can disrupt at least $(\beta - \beta')\binom{n}{2}$ more pairs in $G[E \setminus \mathcal{D}]$. By Lemma (4), $\phi(\mathrm{OPT}_{\beta'}) \geq \alpha_c(G[E \setminus \mathcal{D}])(\beta - \beta')\binom{n}{2}$. Hence, $\alpha_c(G[E \setminus \mathcal{D}]) \leq \frac{\phi(\mathrm{OPT}_{\beta'})}{(\beta-\beta')\binom{n}{2}}$.

The algorithm adds a connective cut $\langle S, T\rangle$ into $\mathcal{D}$. The average cost to disrupt a pair is

$$\alpha_c(S,T) = \frac{\phi(S,T)}{|S||T|} \leq O(\sqrt{\log n})\alpha_c(G[E \setminus \mathcal{D}]) \leq O(\sqrt{\log n})\frac{\phi(\mathrm{OPT}_{\beta'})}{(\beta-\beta')\binom{n}{2}}.$$

Since Algorithm 1 can disrupt at most $\binom{n}{2}$ pairs, the total cost .i.e. the total capacity of $\mathcal{D}$ is bounded by

$$\binom{n}{2}O(\sqrt{\log n})\frac{\phi(\mathrm{OPT}_{\beta'})}{(\beta-\beta')\binom{n}{2}} = \frac{1}{\beta-\beta'}O(\sqrt{\log n})\phi(\mathrm{OPT}_{\beta'}).$$

Time complexity. In each step, only a connected component is divided into two while the others are "untouched". We only need to find the sparsest cut in those new components. Let $T(n)$ be the time complexity of our algorithm. Assume the algorithm cut the graph into two components of size k and $n-k$. Then, $T(n) \leq \tilde{O}(n^2) + T(k) + T(n-k)$. It is easy to show that $T(n) = \tilde{O}(n^2)$. □

4 Exact Algorithm

Linear Integer Programming (IP) is perhaps the most general method for formulating difficult combinatorial problems. However, integer variables make optimization problems far more difficult to solve. Solution time and memory may rise exponentially as more integer variables added. In addition, solving integer programming requires solving a sequence of Linear Programming (LP) relaxation, consists of optimizing without requiring the variables to be an integer.

IP formulation for k-CND, k-CED two closely related problems to β-disruptor, were presented in [3]. Both of them require $O(n^2)$ integer variables and $(n-2)\binom{n}{2}$ linear constraints, where n is the number of nodes in the network. The running time of the IP is mostly unbearable when $n > 80$. The paper reported the results for networks of 150 nodes; we however note that all of large instances in the paper associated with extreme low connectivity levels ($\beta < 0.01$) that make the problem really easy to solve.

In this section, we devise a new Mixed-Integer Programming (MIP) formulation for β-disruptor problem that consists only n integer variables. Furthermore, we combine the row-generation technique in [9] and the combinatorial structure of the problem to devise in subsection 4.2 a new technique called *compressed metric*. The proposed technique considerably cuts down the number of constraints from $O(n^3)$ to $O(n^2)$. Our approach is capable of finding optimal disruptor for large networks of several hundred nodes. For example in the largest instance, we managed to find the optimal disruptor of a 1000 nodes power-law network on our desktop computer.

4.1 Mixed-Integer Linear Programming

For simplicity, we consider only undirected network with unit node-weights, however, with subtle modifications the MIP can also handle both the directed and general node-weights cases. The technique can also be widely applied for problems involving connectivity e.g. β-edge disruptor, k-CND, k-CED, vertex/edge separator, and so on.

The MIP formulation for the β-vertex disruptor problem is presented below. We assume that nodes in the network $G = (V, E)$ are numbered from 1 to n.

$$\text{minimize} \quad \sum_{i=1}^{n} s_i \tag{7}$$

$$\text{subject to} \quad s_i \le c_{i,j}, \qquad i \ne j, \tag{8}$$

$$c_{i,j} \le s_i + s_j, \qquad (i,j) \in E, \tag{9}$$

$$c_{i,j} + c_{j,k} \ge c_{i,k}, \qquad (i,j,k) \in \mathcal{T} \subseteq [1..n]^3, \tag{10}$$

$$\sum_{i<j} c_{i,j} \ge (1-\beta)\binom{n}{2}, \tag{11}$$

$$c_{i,j} = c_{j,i}, \qquad i,j \in [1..n], \tag{12}$$

$$c_{i,j} \in [0,1], \qquad i,j \in [1..n], \tag{13}$$

$$s_i \in \{0,1\} \qquad i \in [1..n], \tag{14}$$

where $s_i = 1$ if i is selected into the disruptor and $s_i = 0$ otherwise. Thus, the disruptor set is given by $\mathcal{D}_{\text{MIP}} = \{ i \mid s_i = 1\}$. The variable $c_{i,j}$ corresponds to the disconnectivity between i and j i.e. $c_{i,j} > 0$ if there is no path connecting i with j, and $c_{i,j} = 0$ otherwise.

The objective function guarantee that a minimum number of nodes will be removed from the network. The constraint (10) implies that if there is a path between i, j ($c_{i,j} = 0$) and there is a path between j, k ($c_{j,k} = 0$), then there is also a path between i, k ($c_{i,k}$ must be 0). Moreover, the constraint (9) guards against the base case that if i, j are neighbors and neither i or j is removed ($s_i = s_j = 0$), then i, j remain connected ($c_{i,j} = 0$). Finally, the constraint (11) ensures the connectivity level in the network is at most a fraction β of that at the beginning.

We first show that the integral requirements on $c_{i,j}$ are actually unnecessary.

Lemma 5. *For every fractional solution of MIP* (7-14)*, there is a feasible solution of the MIP with the same objective value in which all variables are integral.*

Proof. Round all $c_{i,j} > 0$ to 1. This will not violate constraints (8) and (11). In case of (9), the integrality of s_i, s_j will make sure $s_i + s_j \geq 1$, if $c_{i,j} > 0$ is rounded up to 1. Assume the rounding violates constraints (10) for some triple $(i, j, k) \in \mathcal{T}$. This happens iff $c_{i,k} = 1$ and $c_{i,j} = c_{j,k} = 0$. Hence, before rounding, $c_{i,k} > 0$ and $c_{i,j} = c_{j,k} = 0$ that contradicts the constraint $c_{i,j} + c_{j,k} \geq c_{i,k}$. It follows that rounding gives a feasible integral solution to the MIP. □

Lemma 6. *For any $\mathcal{T} \subseteq [1..n]^3$, the optimal solution of* MIP(7-14) *gives a lower bound for the size of the minimum β-vertex disruptor in G.*

Proof. Let $\mathcal{D}^*$ be the minimum capacity β-vertex disruptor of G. By setting $s_i = 0\ \forall i \in \mathcal{D}^*$ and $c_{i,j} = 0$ for all i, j in a same connected component of $G_{[V \setminus \mathcal{D}]}$, the graph obtained by removing vertices $\mathcal{D}$ from G, and $c_{i,j} = 1$ if not, we yield a feasible solution for MIP (7-14). Hence, $|\mathcal{D}^*| \geq |\mathcal{D}_{\text{MIP}}|$, the optimal solution of the MIP. □

Lemma 7. *$\mathcal{D}_{\text{MIP}}$ is a β-vertex disruptor, if $c_{i,j} = 0$ for all connected pairs (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$.*

Proof. The proof is straightforward. Assume that $c_{i,j} = 0$ for all connected pairs (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. Only disconnected pairs contribute to constraint (11). Since $c_{i,j} \leq 1 \quad \forall i, j \in [1..n]$, the number of disconnected pairs must be at least $(1 - \beta)\binom{n}{2}$. Hence, there will be at most $\beta\binom{n}{2}$ connected pairs in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. □

Lemma 8. *When $\mathcal{T} = \{(i, j, k) \in V^3 : i \neq j \neq k\}$, the subset $\mathcal{D}_{\text{MIP}} = \{ i \mid s_i = 1\}$ obtained by solving* MIP(7-14) *is a minimum β-vertex disruptor of G.*

Proof. By Lemma (6), $\mathcal{D}_{\text{MIP}}$ has size at most that of the minimum β-vertex disruptor. The rest is to show that $\mathcal{D}_{\text{MIP}}$ is a β-vertex disruptor.

By Lemma (7), we need to show $c_{i,j} = 0$ for every connected pairs (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. Note that c is a pseudo-metric, i.e., the function $c(i, j) = c_{i,j}$ satisfy:

1. $c(i,j) \geq 0$ (non-negativity)
2. $c(i,i) = 0$ (and possibly $c(i,j) = 0$ for some distinct values $i \neq j$)
3. $c(i,j) = c(j,i)$ (symmetry)
4. $c(i,j) \leq c(i,k) + c(k,j)$ (subadditivity/triangle inequality).

Let (i,j) be a connected pair in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. There is a path $v_0 = i, v_1, \ldots, v_t = j$ connecting i and j and none of nodes in the path were removed i.e. $s_{v_j} = 0 \; \forall j = 0..t$. Since the constraint (9) holds for all edges on the path, we have $c_{v_j,v_{j+1}} \leq s_{v_j} + s_{v_{j+1}} = 0$. Apply the triangle inequalities, we have

$$c_{i,j} \leq c_{v_0,v_1} + c_{v_1,v_t} \leq c_{v_0,v_1} + c_{v_1,v_2} + c_{v_2,v_t} \leq \ldots \leq c_{v_1,v_2} + \ldots + c_{v_{t-1},v_t} = 0. \square$$

Although selecting $\mathcal{T} = \{(i,j,k) \in V^3 : i \neq j \neq k\}$ gives us the desired solution for the disruptor problem, it adds $\Omega(n^3)$ constraints to the MIP. A large number of constrains put stresses on both the time and memory requirement to optimize the mathematical formula. For example when the network has 1,000 nodes the number of triangle inequalities in the MIP (7-14) explodes to 500 million that can hardly be solved even with sophisticated algorithm and modern supercomputer.

4.2 Compressed Metric

Instead of including all triangle inequalities into the MIP to make c a pseudo-metric, we use only a small subset of triangle inequalities (hence, the name *compressed metric*). The goal is to find a concise set of constraints that captures the essential structure of connectivity inside the network while ensuring the subset $\mathcal{D}_{\text{MIP}}$ is a valid β-vertex disruptor.

We shall use the triple (i,j,k) to refer to the constraint $c_{i,j} + c_{j,k} \geq c_{i,k}$. The following lemma establishes the foundation for our technique.

Lemma 9. *If for all connected pair i,j in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$ there exists $(i,j,k) \in \mathcal{T}$ where k is a node on some shortest path between i and j, then the* MIP *returns a minimum size β-vertex disruptor.*

Proof. Assume that $\mathcal{T}$ contains for each connected pair (i,j) a triples (i,j,k) where k is on some shortest path between i,j in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. We shall prove that for all connected pair i,j in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$, $c_{i,j} = 0$. Hence, having more than $\beta\binom{n}{2}$ connected pairs in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$ will violate constraint (10).

We use the induction on the length t of the shortest path between i and j to prove that "*For all $t > 0$, if there exists a path between i and j of length at most t in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$, then $c_{i,j} = 0$.*"

The basis. The statement holds for $t = 1$ according to the constraint (9).

The inductive step. Assume that the statement holds for $t = t'$, we show that the statement is also true for $t = t' + 1$. Let i,j be some pairs connected with a path of length at most $t' + 1$. Since there exists k on the shortest path from i to j in $G_{[V \setminus \mathcal{D}]}$, we have $c_{i,k} + c_{k,j} \geq c_{i,j}$. In addition, the shortest paths from i to k and from k to j are all at most of length t'. Using the induction hypothesis we have $c_{i,k} = c_{k,j} = 0$. It follows that $c_{i,j} \leq c_{i,k} + c_{k,j} = 0$.

Thus, the statement holds for all $t > 0$. $\square$

Remark 1. We can replace the condition that k must be on some shortest path between i and j with a weaker condition that distances (in hops) from k to i and from k to j are strictly smaller than the distance from i to j.

Remark 2. With reduced set of constraints $\mathcal{T}$, c might be no longer a pseudo-metric. However, $\mathcal{T}$ is tight enough to force $\mathcal{D}_{\text{MIP}}$ a valid β-vertex disruptor.

Based on Lemma (9), we describe in Algorithm 2 a *Branch &Cut with Compressed Metric (B&C CM)* to find the minimum capacity β-vertex disruptor.

Algorithm 2. Branch&cut with compressed metric(B&C CM)

1: $\mathcal{T} = \emptyset$
2: **for each** $(i, j) \in V \times V$ **do**
3:     Find a node k on a shortest path from i to j
4:     $\mathcal{T} = \mathcal{T} + \{(i, j, k)\}$
5: $iteration = 0$
6: **Repeat**
7:     $iteration = iteration + 1$
9:     Solve the MIP using branch and cut method
10:     Let $\mathcal{D} = \{\ i \mid s_i = 1\}$.
11:     **for each** connected pair (i, j) in $G_{[V \setminus \mathcal{D}]}$ **do**
12:         **if** $(c_{i,j} > 0)$ **then**
13:             Find a shortest path $P_{i,j}$ between i, j in $G_{[V \setminus \mathcal{D}]}$
14:             **if** (degree(i) < degree(j)) **then**
15:                 select k' as the neighbor of i on $P_{i,j}$
16:             **else** select k' as the neighbor of j on $P_{i,j}$
17:             $\mathcal{T} = \mathcal{T} + \{(i, j, k')\}$
18: **Until** ($\mathcal{D}$ is a β-vertex disruptor)
19: Output $\mathcal{D}$.

Initially, we select for each connected pair (i, j), a node k on some shortest path from i to j. If $\mathcal{D}$ obtained by solving the MIP is not a β-vertex disruptor, by Lemma (7), there must be a connected pair (i, j) in $G_{[V \setminus \mathcal{D}]}$ with $c_{i,j} > 0$. The main reason $c_{i,j} > 0$ for a connected pair (i, j) is that all nodes in $\{\ k \mid (i, j, k) \in \mathcal{T}\}$ are in different connected components with i, j. We fix the problem by including into $\mathcal{T}$ a triple (i, j, k') where k' is on some shortest path between i, j in $G_{[V \setminus \mathcal{D}]}$ and solve the MIP again after examining all pairs. The algorithm terminates when $\mathcal{D}$ is a β-vertex disruptor. Selecting of k' as in lines 14 to 16 in Algorithm 2, we assure that the number of triples involving i, j is at most $\min\{degree(i), degree(j)\}$.

Remark 3. We can select k' as vertices on the minimum vertex cut separating i and j [10]. However, selecting such k' migh be time-consuming.

Lemma 10. *The algorithm output the optimal disruptor in at most δ_G iterations, where δ_G is the maximum degree in the network.*

Lemma 11. *The number of added constraints in each iteration is at most $\binom{n}{2}$. Overall, the total number of constraints is at most $O(mn)$.*

4.3 Comparisons and Discussions

Vertex	Edge	β	Removed vertex	Time (seconds) BC	Time (seconds) B&C CM	Constraint BC	Constraint B&C CM	BC&CM Round
50	141	60.0%	4	63	8	60, 167	4, 861	8
150	286	1.0%	18	19, 788	2	1, 665, 362	31, 887	2
-	-	5.0%	15	18, 070	7	-	32, 161	4
-	-	8.0%	12	n/a	73	-	33, 242	3
-	-	10.0%	11	n/a	1, 363	-	39, 615	5
-	-	20.0%	9	n/a	1, 737	-	39, 313	3
-	-	40.0%	7	n/a	2, 149	-	42, 830	4
-	-	60.0%	5	n/a	1, 610	-	38, 458	3
-	-	90.0%	2	26, 277	147	-	34, 321	2
200	387	60.0%	8	n/a	64, 860	3, 960, 488	72, 980	6
600	1, 166	0.5%	69	n/a	48, 918	107, 641, 467	516, 656	3
1000	1, 959	0.5%	198	n/a	747	499, 340, 027	1, 437, 326	1

Table 1: Comparisons of MIPs on power-law networks

In our actual implementation, at the initial step we include for each pair (i, j) two triples (i, j, k_1) and (i, j, k_2), where k_1, k_2 belong to two node-disjoint paths between i and j to reduce the number of iterations. We solve the MIP using branch and cut algorithm equipped in GUROBI 3.0 on a Intel Xeon 2.93 Ghz with 12 GB memory.

Table 1 shows statistics for the original branch and cut (B&C) and our new branch and cut with compressed metric (B&C CM) algorithm on power-law networks [11] of various sizes. For the B&C CM, we take the measures of the size of the MIP formula at the final iteration. We report algorithm for each disruption level β, the number of removed vertices in the optimal solution, the number of Rows (constraints), Nonzeros (nonzero coefficients), and solving time.

Our algorithm utilizing compressed metric technique is substantially faster and more memory-efficient than the original Branch & cut equipped in GUROBI MIP solver. The speed up for 50 nodes network is about eight times. For the network of 150 nodes, B&C CM often takes less than 30 minutes, while B&C runs out of memory or does not terminate within two days (noted with *n/a*).

References

1. T. N. Dinh, Y. Xuan, M. T. Thai, E. K. Park, and T. Znati. On approximation of new optimization methods for assessing network vulnerability. INFOCOM, 2010.
2. G. D. Bissias, B. N. Levine, and A. Rosenberg. Bounding damage from link destruction, with application to the internet. *SIGMETRICS Perform. Eval. Rev.*, 2007.
3. 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.

4. W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17, September 1973.
5. J. Cullum, W. E. Donath, and P. Wolfe. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. In *Nondifferentiable Optimization*, Mathematical Programming Studies. Springer Berlin Heidelberg, 1975.
6. Mohar and Poljak. Eigenvalue in combinatorial optimization. 1992.
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition edition, 2009.
8. S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC*, pages 222–231, New York, NY, USA, 2004. ACM.
9. M. Grtschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96, 1989. 10.1007/BF01589097.
10. H. Nagamochi and T. Ibaraki. *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press, New York, NY, USA, 2008.
11. A. Barabasi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A*, 2000.