

Precise Structural Vulnerability Assessment via Mathematical Programming

Thang N. Dinh and My T. Thai
Computer & Information Science & Engineering
University of Florida, Gainesville, FL, 32611
Email: {tdinh,mythai}@cise.ufl.edu

Abstract—Network vulnerability assessment is an indispensable component of attack risk reduction and proactive response. However, traditional assessment methods simply assume the attacks only target at nodes with high degree or betweenness centrality, thus fail to capture the worst-case scenarios under simultaneous failures. In our previous work, we formulated assessing network vulnerability as optimization problems, so-called β -edge disruptor and β -vertex disruptor, to identify the minimum cost critical infrastructures that removal expose the network to a certain disruption level. Here, the disruption is measured as the fraction of node pairs with no paths between them in the residual network.

In this paper, we present an exact analytical solution for the vulnerability assessment problems i.e. an exact branch-and-cut algorithm to solve the integer programming formulation of the β -vertex disruptor. The two intriguing aspects of the algorithms are an efficient Mixed Integer Programming (MIP) formulation, called *sparse metric*, and *vertex-cut inequalities*, a specialized cutting plane procedure that tightens the bound on the optimal solutions. Experiments on both synthetic and real-world networks suggest that our algorithm yields a significant improvement on a large variety of network instances, raising the size of the largest instance solved from several dozen to several hundred nodes. Our techniques can be easily extended to many graph partitioning and connectivity optimization problems.

I. INTRODUCTION

Infrastructures in many networks such as telecommunication, transportation networks, gas and power grid systems are highly interdependent and sensitive to both random failures and intentional attacks. In fact, the failures of a small number of nodes may lead to a complete fragmentation of a network system and severely disrupt the network connectivity. Real-world examples include the unplanned destruction of fiber cables by dragging anchors [?], [?], malicious cyber-attack (e.g. a distributed-denial-of-service attacks) [?], [?], [?] to Internet Autonomous Systems and terrorist attacks targeting infrastructures in electrical power grids and highway systems [?], [?], [?]. Therefore, it is essential to assess the network vulnerability to those fatal failure schemes before they happen.

There have been numerous efforts on proposing evaluation measures of the network vulnerability, as summarized in [?], [?]. However, these measures can neither be rigorously mapped to the overall 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. To facilitate the search for critical infrastructures in networks regarding network connectivity, a new assessment method has

been proposed in [?] in form of an optimization problem, so-called β -vertex disruptor. The network vulnerability was measured through the minimum number of nodes that removal incurs a certain level of disruption in the objective network. Extensive experiments [?] on both synthetic and real networks showed that the new assessment method outperforms the traditional ones and successfully identifies small subsets of critical nodes that failures lead to the network wide fragmentation. In addition, the flexibility in selecting the level of disruption assessing vulnerability at multiple disruption levels, providing a complete network vulnerability spectrum.

To solve the β -vertex disruptor problem, which was shown to be an NP-hard problem, the authors [?] proposed a pseudo-approximation algorithm that can guarantee the performance of the solution to be within a factor $O(\log n \log \log n)$ of the optimal solutions. Despite that the algorithm is of theoretical interests, it has high time complexity and is difficult to be implemented efficiently. Besides designing algorithms with performance guarantees, the β -vertex disruptor problem can be formulated using integer programming (IP) [?] and solved for the exact solutions by branch-and-cut methods, which consist of a combination of a *cutting plane method* with a *branch-and-bound* algorithm. The same approach has been applied for the k critical nodes (edges) detection problems [?], [?] (abbreviated with k -CND and k -CED) that seek for a set of k nodes (edges) that removal maximizes the disruption in the residual network. Unfortunately, even for small network instances all proposed formulations become very large integer programming problems that consume excessive amount of memory and time to converge. For example, the largest reported instance with 150 nodes consists more than one million constraints. Moreover, solving those integer programming problems relied solely on the general version of the branch-and-cut algorithm implemented in the optimization packages that are not tailored to those specific problems.

In this paper, we overcome difficulties in previous mathematical approaches by introducing a new Mixed Integer Programming (MIP) formulation for the β -vertex disruptor problem that contains significantly less constraints, and deriving strong cutting planes that improve the linear programming relaxation of the MIP to further accelerate the search. The cutting planes are derived from a new class of *valid inequalities*, those are satisfied by all the feasible solutions of the MIP, so-called vertex-cut (VC) inequalities. Furthermore, we

use a greedy primal rounding heuristics to generate upper bounds at every node of the branch-and-bound search tree. The experimental results show the significant improvements on small to medium sized network instances, raising the size of the largest problem solved to thousands of nodes.

Our sparse MIP formulation and cutting planes based on VC inequalities can be applied for numerous of graph clustering and partitioning problems such as minimum weighted clustering [?], capacitated clustering [?], correlation clustering [?], maximizing modularity to detect the community structure [?] and so on.

Organization. We start with problem definitions and original integer programming formulation of β -vertex disruptor in Section II. The components of our branch-and-cut algorithm are presented in Section III. First, the sparse metric, an efficient MIP formulation for β -disruptor is introduced together with their polyhedra characteristics. Then, we propose a new set of valid inequalities (VC inequalities) and an exact separation routine to produce strong cutting planes. We complete the section with a primal rounding heuristic to produce good quality solution that reduces the number of explored branches in the search tree. Computational experiments are presented in Section IV. Finally, we conclude the paper in Section V.

II. PRELIMINARIES

We model the network as an undirected graph $G = (V, E)$ of n nodes numbered from 1 to n ; the degree of node $1 \leq i \leq n$ is denoted by $d(i)$. The pairwise connectivity of G , denoted by $\mathcal{P}(G)$ is the number of node pairs with at least one path between them. For example, if G is connected, then $\mathcal{P}(G) = \binom{n}{2}$.

Given a positive constant $0 \leq \beta \leq 1$, a subset of vertices $S \subset V$ in G is a β -vertex disruptor if the subgraph $G_{[V \setminus S]}$, induced by $V \setminus S$ in G , has pairwise connectivity at most $\beta \binom{n}{2}$. The β -edge disruptor problem asks to find a β -vertex disruptor of the minimum size.

The problem can be generalized so that each node $u \in V$ has a cost $w(u)$ of removing and we wish to find a β -vertex disruptor of the minimum cost. This generalization is straightforward and shall be ignored to simplify the presentation.

The IP formula for β -vertex disruptor (IP_{vd}) is as follow

$$\text{minimize } \sum_{i=1}^n s_i \quad (1)$$

$$\text{subject to } d_{ij} \leq s_i + s_j, \quad (i, j) \in E, \quad (2)$$

$$d_{ij} + d_{jk} \geq d_{ik}, \quad \forall i \neq j \neq k \quad (3)$$

$$\sum_{i < j} d_{ij} \geq (1 - \beta) \binom{n}{2}, \quad (4)$$

$$s_i \leq d_{ij}, \quad i \neq j \quad (5)$$

$$s_i, d_{ij} \in \{0, 1\}, \quad i, j \in [1..n] \quad (6)$$

We use variable d_{ij} to represent the “distance” between a pair of nodes i and j in the residual network i.e.

$$d_{ij} = \begin{cases} 0 & \text{if } i \text{ and } j \text{ are in the same connected component} \\ 1 & \text{otherwise.} \end{cases}$$

An extra variable s_i is used for each node $i \in V$, where

$$s_i = \begin{cases} 0 & \text{if node } i \text{ is not removed} \\ 1 & \text{if } i \text{ is removed (selected into the disruptor.)} \end{cases}$$

The objective minimizes the total number of removed nodes i.e. the size of the vertex disruptor. Note that $d_{ij} = d_{ji} \quad \forall (i, j) \in V \times V$. Constraint (3) is the well-known *triangle inequality* which implies that if i and j are connected, and j and k are connected, then i and k must be connected. Constraint (4) limits the pairwise connectivity in G to be at most $\beta \binom{n}{2}$.

Constraint (2) implies the base case that if i and j are neighbors and neither i or j is removed ($s_i = s_j = 0$), then i and j remain connected i.e. $d_{ij} = 0$. Constraint (5) states the fact that a removed node will not connect to any other nodes ($s_i = 1 \rightarrow d_{ij} = 1$).

There are several drawbacks with the IP formula of the β -vertex disruptor problem (IP_{vd}) (and also formulations of k -CND and k -CED [?]). A large number of integral variables, $\Theta(n^2)$, makes the selection of branching difficult and significantly increases the depth and size of the search tree. In addition, excessive number of constraints, $\Theta(n^3)$, even for small sized instances leads to a large linear programming relaxation that consumes an extremely large amount of memory and computing time.

III. BRANCH-AND-CUT ALGORITHM

Branch-and-cut methods have proven to be a very successful approach for solving a wide variety of integer programming problems. In contrast with meta-heuristics, they can guarantee optimality. They combine a *branch-and-bound* algorithm with a *cutting plane method* that is used to improve the solution of the linear programming relaxations.

This section presents components of our branch-and-cut algorithm. We begin with a new lightweight mixed integer programming formulation for β -vertex disruptor in Subsection III-A. In the next subsection, we introduce a new class of strong cutting planes and the separation procedure to find such cutting planes. The primal heuristics that provides upper bounds for pruning during the search process is presented in Subsection III-C.

A. Sparse Metric

We first devise a new Mixed-Integer Programming (MIP) formulation for the β -vertex disruptor problem that consists of only n integer variables and much smaller number of constraints. Since the only role of triangle inequalities is to guarantee d_{ij} to be a pseudo-metric (as defined later in the proof of Theorem 1), we introduce a compact subset of inequalities, so-called *sparse metric*, that guarantees the same pseudo-metric property. When the network is sparse i.e. $|E| \propto |V|$, the number of constraints reduces substantially from $\Theta(n^3)$ to $\Theta(n^2)$.

Our new MIP formulation for the β -vertex disruptor problem (MIP_{vd}) is similar to IP_{vd} except in places of constraints

(3) and (6) as presented below.

$$d_{ij} + d_{jk} \geq d_{ik}, \quad k \in N_{\min}(i, j) \quad (7)$$

$$d_{ij} \in [0, 1], \quad i, j \in [1..n], \quad (8)$$

where $N_{\min}(i, j)$ is the set of neighbors of i excluding j if $d(i) < d(j)$, and $N_{\min}(i, j)$ is the set of neighbors of j excluding i , otherwise. We also drop the integral requirements on d_{ij} i.e. replace the constraints $d_{ij} \in \{0, 1\}$ with $d_{ij} \in [0, 1]$. However, the integrality of s_1, s_2, \dots, s_n remains.

Note that there are exactly n integer variables s_1, s_2, \dots, s_n . In addition, the number of constraints is upper bounded by

$$\begin{aligned} |E| + \sum_{i < j} \min\{d(i), d(j)\} + \frac{n(n-1)}{2} \\ \leq n^2 + \sum_{i < j} \frac{d(i) + d(j)}{2} = n^2 + \frac{n-1}{2} \sum_{i=1}^n d(i) = O(mn) \end{aligned}$$

Hence, the number constraints is substantially less than $O(n^3)$ for complex networks that are often sparse.

We proceed to prove the equivalence of the compact formulation MIP_{vd} to IP_{vd} by showing the following

- The integrality constraints on $d_{ij} \forall i, j$ are in fact redundant (Proposition 1).
- The optimal solutions of MIP_{vd} also induce optimal β -vertex disruptor in G (Theorem 1).
- The optimal fractional solution of LP relaxation of IP_{vd} can be found by solving the (smaller) LP relaxation of MIP_{vd} , following by an $O(mn + n^2 \log n)$ tuning procedure (Theorem 2).

Proposition 1: For every optimal solution of MIP_{vd} , there is a feasible solution of the MIP with the same objective value in which all variables are integral.

Proof: Round all $d_{ij} > 0$ to 1. This will not violate constraints (5) and (4). For constraints (2), if d_{ij} is rounded up to 1 then the integrality of s_i, s_j implies $s_i + s_j \geq 1$, or else if $d_{ij} = 0$ then the constraints are still satisfied. Assume the rounding violates constraints (7) for some triple (i, j, k) . This happens if and only if $d_{ik} = 1$ and $d_{ij} = d_{jk} = 0$. Hence, before rounding, $d_{ik} > 0$ and $d_{ij} = d_{jk} = 0$ that contradicts the constraint $d_{ij} + d_{jk} \geq d_{ik}$. It follows that rounding gives a feasible integral solution to the MIP. ■

Let $\mathcal{D}_{\text{MIP}} = \{i \mid s_i = 1\}$ be the disruptor induced by the optimal solution of MIP_{vd} and $\text{OPT}_{\text{vd}}^\beta$ be an optimal β -vertex disruptor.

By setting $s_i = 0 \forall i \in \text{OPT}_{\text{vd}}^\beta$ and $d_{ij} = 0$ for all i, j in a same connected component of $G_{[V \setminus \text{OPT}_{\text{vd}}^\beta]}$ and $d_{ij} = 1$ if not, we yield a feasible solution for MIP_{vd} . Therefore,

$$|\mathcal{D}_{\text{MIP}}| \leq |\text{OPT}_{\text{vd}}^\beta|$$

Theorem 1: The optimal solution $\mathcal{D}_{\text{MIP}} = \{i \mid s_i = 1\}$ obtained by solving MIP_{vd} is a minimum β -vertex disruptor of G .

Proof: Since $|\mathcal{D}_{\text{MIP}}| \leq |\text{OPT}_{\text{vd}}^\beta|$, we only need to show that \mathcal{D}_{MIP} is a β -vertex disruptor.

Assume that we can prove that $d_{ij} = 0$ for every connected pairs (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$. Then, only disconnected pairs $d_{i'j'}$ will contribute to the sum in constraint (4). Since $d_{i'j'} \leq 1 \forall i', j' \in [1..n]$, the number of disconnected pairs must be at least $(1 - \beta) \binom{n}{2}$. It will follow that \mathcal{D}_{MIP} is a β -vertex disruptor.

Hence, the rest of the proof is to show that $d_{ij} = 0$ for every connected pairs (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$.

Note that d is a pseudo-metric, i.e., the function $d(i, j) = d_{ij}$ satisfy:

- 1) $d(i, j) \geq 0$ (non-negativity)
- 2) $d(i, i) = 0$
- 3) $d(i, j) = d(j, i)$ (symmetry)
- 4) $d(i, j) \leq d(i, k) + d(k, j)$ (transitivity).

For each connected pair (i, j) in $G_{[V \setminus \mathcal{D}_{\text{MIP}}]}$, we prove that $d_{ij} = 0$ by induction on the length t of the shortest path (in number of hops) between nodes i and j .

The basis. The statement holds for $t = 1$. By constraint (2), if $(i, j) \in E$ and i, j are connected in G i.e. $s_i = s_j = 0$, then $d_{ij} \leq s_i + s_j = 0$. Since $d_{ij} \geq 0$, it follows that $d_{ij} = 0$.

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 removing all nodes in $N_{\min}(i, j)$ disconnects i from j , the path between i and j must pass through some node $k \in N_{\min}(i, j)$. In addition, the shortest paths from i to k and from k to j have lengths at most t' . Thus, by the induction hypothesis we have $d_{ik} = d_{kj} = 0$. It follows from the constraint in (7) that $d_{ij} \leq d_{ik} + d_{kj} = 0$. Thus, the statement holds for all $t > 0$. ■

Finally, we show the relationship between the LP relaxation of IP_{vd} and that of MIP_{vd} .

Theorem 2: The optimal solution of the LP relaxation IP_{vd} can be found by solving the LP relaxation of MIP_{vd} , following by an $O(mn + n^2 \log n)$ tuning procedure.

Proof: Let (s, d) be an optimal fraction solution of the LP relaxation of MIP_{vd} . Associate a weight d_{ij} for each edge $(i, j) \in E$. Let d'_{ij} be the shortest distance between two nodes (i, j) with the new edge weights. We have

- 1) $d'_{ij} \geq d_{ij}$ for all i, j and $d'_{ij} = d_{ij} \forall (i, j) \in E$.
- 2) $d'_{ij} = \min_{k=1}^n \{d'_{ik} + d'_{kj}\}$. Hence, d'_{ij} is a pseudo-metric.

The first statement can be shown by the same induction in the proof of Theorem 1. The second statement comes from the definition of d'_{ij} .

Furthermore, we define $d_{ij}^* = \min\{d'_{ij}, 1\}$. If we use the Johnson's algorithm [?] to compute all pairs shortest paths d'_{ij} , the time complexity to construct d_{ij}^* from d_{ij} is $O(mn + n^2 \log n)$. We shall prove that (s, d^*) is a feasible solution of IP_{vd} by showing that (s, d^*) satisfies all constraints in IP_{vd} .

By definition, we have $d_{ij}^* = \min\{d'_{ij}, 1\} \geq \min\{d_{ij}, 1\} = d_{ij} \forall i, j$ and $d_{ij}^* = d_{ij} \forall (i, j) \in E$. Thus, for all $(i, j) \in E$, $d_{ij}^* = d_{ij} \leq s_i + s_j$.

In addition, d^* is also a pseudo-metric as $d_{ik}^* + d_{kj}^* \geq \min\{d'_{ik} + d'_{kj}, 1\} \geq \min\{d'_{ij}, 1\} = d_{ij}^*$.

From $d_{ij}^* \geq d_{ij}$, we have $\sum_{i,j} d_{i,j}^* \geq \sum_{i,j} d_{i,j} \geq \beta \binom{n}{2}$ and $s_i \leq d_{ij} \leq d_{ij}^*$. Thus, (s, d^*) is a feasible solution of IP_{vd} .

Obviously, the minimum objective of the LP relaxation of MIP_{vd} is smaller or equal to that of IP_{vd} . Since, the objective values associate with (s, d^*) and (s, d) , a minimum solution of the LP relaxation of MIP_{vd} , are the same, (s, d^*) must be a minimum solution of the LP relaxation of IP_{vd} . ■

B. Cutting Planes

We present a class of strong cutting planes together with the separation procedure to identify those cutting planes. These can be used in conjunction with cutting planes generated automatically by optimization packages to improve the convergence of the branch-and-cut algorithm.

1) *Vertex-Connectivity and Invalid Inequalities*: One often overlooked characteristic of solutions for clustering and partitioning problems on graph is that clusters must induce connected subgraph. This characteristic is not reflected in either IP_{vd} or MIP_{vd} formulations.

A subset $S \subset V$ is a *vertex-cut* for a pair (u, v) , if removing S from graph G , disconnect s and t . For all vertex-cut S of (u, v) , if $\sum_{i \in S} s_i = |S|$, then d_{uv} must be one. Thus, we have VC inequality

$$\sum_{i \in S} s_i - d_{uv} \leq |S| - 1$$

This inequality is valid for all feasible points inside the polyhedra of MIP_{vd} .

Algorithm 1. Separation procedure for VC inequalities

- 1: **for each** pair $(u, v) \in V \times V$ **do**
 - 2: Construct a flow network $G = (V, E)$ as follows
 - 3: Assign u and v as source and sink, respectively
 - 4: Each node $k \in V$ has capacity \bar{s}_k
 - 5: Every edge has capacity ∞
 - 6: **if** $(u, v) \in E$, **then** (u, v) has capacity zero.
 - 7: Find the maximum-flow (min-cut)
 - 8: **if** maximum-flow is less than \bar{d}_{uv} , **then**
 - 9: Find the min vertex-cut S
 - 10: Add the VC inequality associated with S to MIP
 - 8: **end if**
 - 11: **end for**
-

2) *Separation Procedure for VC Inequalities*: Given a point (fractional solution) $(s, d) \in \mathbb{R}^{\binom{n+1}{2}}$, an exact separation algorithm for some class of inequalities either finds a member of the class violated by (s, d) , or proves that no such member exists. In many cases, finding such algorithm is intractable (NP-hard problem) and one has to settle for heuristic procedures. Fortunately, there is an exact algorithm for our separation procedure based on finding the max-flow on the network with node capacities.

The VC inequality can be rewritten as

$$\sum_{i \in S} \bar{s}_i - \bar{d}_{uv} \geq 0, \quad S \text{ is any vertex-cut of } (u, v)$$

where $\bar{s}_i = 1 - s_i$ and $\bar{d}_{uv} = 1 - d_{uv}$.

Therefore, the point (s, d) violates this inequality if and only if $\sum_{i \in S} \bar{s}_i < \bar{d}_{uv}$. The most violated inequality is the

Algorithm 2. Sharpest Decreasing Vertices (SDV)

- 1: Start with some β -vertex disruptor $\mathcal{D} \subseteq V$.
 - 2: **Repeat**
 - 3: **while** (true) **do**
 - 4: $u = \arg \min_{v \in \mathcal{D}} \{\mathcal{P}(G_{[V \setminus (\mathcal{D} \cup \{v\})]})\}$
 - 5: **if** $(\mathcal{D} \setminus \{u\})$ is a β -disruptor **then** $\mathcal{D} = \mathcal{D} \setminus \{u\}$
 - 6: **for each** $(w \in V \setminus \mathcal{D})$ **do**
 - 7: $D_w = \mathcal{D} \cup \{w\}$
 - 8: $u = \arg \min_{v \in \mathcal{D}} \{\mathcal{P}(G_{[V \setminus (D_w \cup \{v\})]})\}$
 - 9: **if** $(u \neq w)$ **then** $\mathcal{D} = D_w \cup \{u\}$
 - 10: **Until** $(\mathcal{D}$ not changing)
 - 11: Output \mathcal{D} .
-

one that minimize the sum $\sum_{i \in S} \bar{s}_i$, given S is a vertex-cut of (u, v) . Thus, the subset S corresponding to the most violated inequalities can be found using minimum capacitated vertex-cut of (u, v) . The separation procedure is described in Algorithm 1. Here, we need to solve the maximum-flow (min-cut) problem in networks with both node and edge capacities. If we apply Push-relabel algorithm with dynamic trees [?], the time complexity to find cutting planes for one node pair is $O(mn \log \frac{n^2}{m})$. The total time complexity for the separation procedure will be $O(n^3 m \log \frac{n^2}{m})$. In our implementation, this procedure is called sparingly in order to avoid excessive running time.

C. Primal Heuristic

The search for an optimal solution in a branch and cut algorithm can be accelerated by obtaining a high quality feasible solution to provide upper bounds for pruning other subproblems. We present a heuristic that rounds the fractional solution of MIP relaxations to get integral solutions.

Let (s, d) be a fractional solution of an LP relaxation. We first sort s_i in non-decreasing order $s_{i_1} \leq s_{i_2} \leq \dots \leq s_{i_n}$. Then we round down all $s_{i_1}, s_{i_2}, \dots, s_{i_k}$ to zero and round up $s_{i_{k+1}}, s_{i_{k+2}}, \dots, s_{i_n}$ to one, where k runs from 1 to n . If the obtain solution is a β -vertex disruptor, a local search method described in Algorithm 2 is then used to refine the solution. The local search method refines the solution by repeatedly:

- Removing node(s) from the disruptor if possible
- Swapping a node w outside the disruptor with a node u in the disruptor that gives the sharpest decrease in connectivity.

The local search terminates when no improvement exists.

IV. COMPUTATIONAL EXPERIMENTS

We implement our branch and cut algorithm using GUROBI 4.0 on a computer with Intel Xeon 2.93 Ghz processor and 12 GB memory. Table I shows results for IP_{vd} and our new branch and cut algorithm (MIP_{vd}) on power-law networks [?] of various sizes. We report 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.

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

TABLE I: Comparisons of IP(large) and MIP(sparse) on power-law networks

As shown in Table I, our branch-and-cut algorithm utilizing sparse metric technique and strong cutting planes is substantially faster and more memory-efficient than the original branch-and-cut equipped in GUROBI MIP solver. The speed up factor is from 8 times for 50 nodes to several thousand times for larger instances. For the network of 150 nodes, MIP_{vd} often takes less than 30 minutes, while IP_{vd} runs out of memory or does not terminate after 100,000 seconds (noted with n/a).

In the second part, we compare methods to identify critical nodes in literature in term of pairwise connectivity with the optimal solution of β -vertex disruptor. For most methods, nodes are selected iteratively based on their “criticality” until obtaining a β -vertex disruptor. Then the sizes of found disruptors are compared. The methods that return smaller disruptors are considered better than the others in assessing network vulnerability under the pairwise connectivity framework. Following methods are compared:

- 1) *Degree Centrality*: The algorithm sequentially remove node with the maximum degree.
- 2) *Betweenness Centrality*: We repeatedly remove the node with maximum betweenness centrality. Recall that the betweenness $Bt(v)$ for vertex 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 vertex v .
- 3) *PageRank*: Nodes are removed in descending order of their PageRank values with the default damping factor of 85% as in [?].
- 4) *Primal heuristic*: For large network (of more than 4,000 nodes) solving MIP is almost intractable. Hence, we resort to solving the LP relaxation of IP_{vd} using the tuning method in Theorem 1 and then perform rounding using the primal heuristics in subsection III-C.

The results for small test cases are shown in Figure 1. We generate random networks following Erdos-Rényi model in which every edge exists with the same probability $p = 5\%$.

The optimal disruptor is significantly smaller than the sizes of the solutions returned by other centrality methods with the quickly widening gap. In the largest instance, *degree centrality*, *betweenness* and *Pagerank* require around 16% of nodes to be

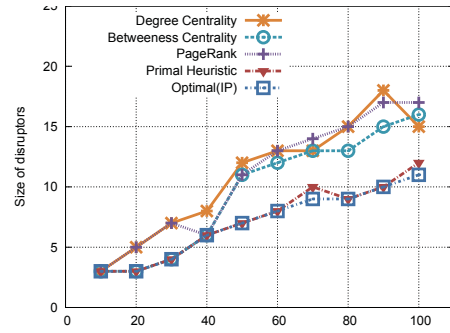


Fig. 1: Random networks at disruptive level $\beta = 60\%$

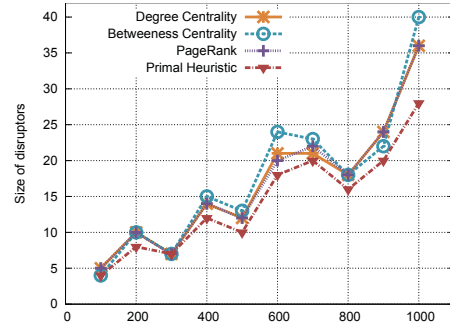


Fig. 2: Power-law networks at disruptive level $\beta = 60\%$

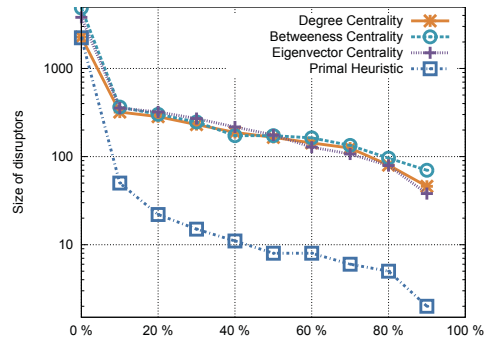


Fig. 3: Vulnerability assessment of Western States Power Grid

removed to reduce the pairwise connectivity by 40% while the size of an optimal disruptor is 50% less than those of centrality methods. It is noted that *Primal heuristic* successfully identify the optimal disruptor in the networks in all cases.

We carry out further testing on power-law networks of larger sizes. The results are shown in Figure 2. Since the running time for the MIP increases sharply, we replace the optimal disruptor solved by the branch-and-cut with the Primal heuristic which often produces optimal or near-optimal solutions.

In power-law networks, all centrality assessments behave roughly the same. Nevertheless, the centrality assessments produce solutions that are 20% larger than those of *Primal heuristic*, the close approximation of the optimal disruptor.

Clearly, common assessment methods are often too optimistic about vulnerability of networks. Only with our framework, the vulnerability in term of network connectivity will be gauged properly.

A. 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 [?]; hence the network is rather vulnerable to targeted attacks.

We show in Figure 3 vulnerability reported by different methods at various levels of disruption. The solutions returned by Primal heuristic is believed to be very close to that of the optimal with the optimality gaps less than 5%.

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. The failure of only 1% of stations can dramatically disrupts 90% connectivity in the network.

None of other methods can reveal correctly the vulnerability of the power grid. They all display significant gaps to Primal heuristic with returned disruptors 6 to 20 times larger than those of Primal heuristic. 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 the worst excluding the random method.

V. CONCLUSION

We have proposed an efficient branch-and-cut algorithm to find exact solution for β -vertex disruptor problem. The techniques used in the paper are general and can be easily

adapted to solve many graph clustering and partitioning problem. Our future work will be combining the column generation technique (branch and price) to find exact solutions for even larger instances of networks.

VI. ACKNOWLEDGEMENT

This work is partially supported by NSF Career Award 0953284, DTRA, Young Investigator Award, Basic Research Program HDTRA1-09-1-0061 and DTRA HDTRA1-08-10.

REFERENCES

- [1] J. B., "Analyzing the Internet Collapse," 2008, MIT Technology Review Online; February 2008.
- [2] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of the fiber infrastructure to disasters," in *Proc. of IEEE INFOCOM*, 2009.
- [3] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, 2004.
- [4] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," ser. SIGCOMM '06. New York, NY, USA: ACM, 2006.
- [5] G. Yan, S. Eidenbenz, S. Thulasidasan, P. Datta, and V. Ramaswamy, "Criticality analysis of internet infrastructure," *Computer Networks*, vol. 54, no. 7, 2010.
- [6] J. Salmeron, K. Wood, and R. Baldick, "Analysis of electric grid security under terrorist threat," *IEEE Transactions on Power Systems*, vol. 19, no. 2, 2004.
- [7] M. Leung, J. H. Lambert, and A. Mosenthal, "A risk-based approach to setting priorities in protecting bridges against terrorist attacks," *Risk Analysis*, vol. 24, no. 4, 2004.
- [8] A. Pinar, J. Meza, V. Donde, and B. Lesieutre, "Optimization strategies for the vulnerability analysis of the electric power grid," *SIAM J. on Optimization*, vol. 20, 2010.
- [9] T. H. Grubestic, T. C. Matisziw, A. T. Murray, and D. Snediker, "Comparative approaches for assessing network vulnerability," *Inter. Regional Sci. Review*, 2008.
- [10] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, 2006.
- [11] T. N. Dinh, Y. X., M. T. Thai, E. Park, and T. Znati, "On approximation of new optimization methods for assessing network vulnerability," in *Proc. of IEEE INFOCOM*, 2010.
- [12] A. Arulselvan, C. W. Commander, L. Eleftheriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Computers & Operations Research*, vol. 36, no. 7, 2009.
- [13] T. Matisziw, A. Murray, and T. Grubestic, "Exploring the vulnerability of network infrastructure to disruption," *The Annals of Regional Science*, vol. 43, 2009.
- [14] M. Grotschel and Y. Wakabayashi, "A cutting plane algorithm for a clustering problem," *Mathematical Programming*, vol. 45, 1989.
- [15] J. M. Mulvey and M. P. Beck, "Solving capacitated clustering problems," *European Journal of Operational Research*, vol. 18, no. 3, pp. 339–348, December 1984.
- [16] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, 2004.
- [17] G. Agarwal and D. Kempe, "Modularity-maximizing graph communities via mathematical programming," *The Eur. Phys. J. B - Condensed Matter and Complex Systems*, vol. 66, 2008.
- [18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.
- [19] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," ser. STOC '86. New York, NY, USA: ACM, 1986.
- [20] A. Barabasi, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A*, 2000.
- [21] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep., 1999.
- [22] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, 1998.