Precise Structural Vulnerability Assessment via Mathematical Programming

Thang N. Dinh and My T. Thai

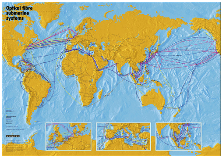
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL 32611

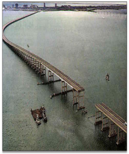
November 09, 2011

1. Introduction
   - Pairwise-connectivity and β-disrutpor
2. Sparse Metric
   - Linear Programming
   - Sparse Metric
3. Branch & Cut Method
   - Cutting Plane
   - Primal Heuristic
4. Experimental Analysis
   - Synthetic Networks
   - Case study: Western Power Grid
5. Conclusion

Vulnerability of Network Systems

- Many networks are vulnerable to both natural disaster and intentional attacks
- Failures of network infrastructure, nodes or links, lead to disruption of services.
- Example:
  - Earthquake near Taiwan disrupts Internet traffic
  - Anchor cut undersea fiber-optic
  - Terrorism activities targeting highway system





Network Vulnerability Assessment

- Assessing the fatal failure schemes before they happen is essential
  - Disaster recovery plan
  - Hardening critical infrastructure
  - Reduce the worst case impact
- Finding the most critical nodes and links that failures will severly disrupt the network
- How to measure the impact of the failures?

Network Vulnerability Assessment

- Assessing the fatal failure schemes before they happen is essential
    - Disaster recovery plan
    - Hardening critical infrastructure
    - Reduce the worst case impact
- Finding the most critical nodes and links that failures will severly disrupt the network
- How to measure the impact of the failures?
- The connectivity level in the residual network: The number of connected pairs a.k.a. pairwise connectivity.

Vulnerability Assessment as an Optimization Problem

- Given a network modeled by a graph $G = (V, E)$, a β-vertex disruptor is a set of vertices that removal lessen the pairwise connectivity to no more than $\beta\binom{|V|}{2}$.
- Smaller the size of β-vertex disruptors, the more vulnerable the network is
- Compute the minimum size of a *beta*-vertex disruptor as an index to measure the network vulnerablity
- We can define β-edge disruptor in the same manor.

β-vertex disruptor

Definition (β-vertex disruptor problem)

Given an undirected graph $G = (V, E)$, find a minimum size β-edge disruptor, a set that removal lessen the pairwise connectivity in the graph to at most $\beta\binom{|V|}{2}$.

- β-edge disruptor is NP-hard and admit an $O(\log n \log\log n)$ pseudo-approximation [Dinh et. al. INFOCOM 2010]
- Exact solution is available only for small instances, sparse networks of around 100 nodes.

Linear Integer Programming (IP)

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} s_i & & (1)\\
\text{subject to} \quad & d_{ij} \le s_i + s_j, & (i,j) \in E, \quad & (2)\\
& d_{ij} + d_{jk} \ge d_{ik}, & \forall i \ne j \ne k \quad & (3)\\
& \sum_{i<j} d_{ij} \ge (1-\beta)\binom{n}{2}, & & (4)\\
& s_i \le d_{ij}, & i \ne j \quad & (5)\\
& s_i, d_{ij} \in \{0,1\}, & i,j \in [1..n] \quad & (6)
\end{aligned}
$$

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

$$
s_i = \begin{cases} 0 & \text{if node } i \text{ is not removed} \\ 1 & \text{if } i \text{ is removed.} \end{cases}
$$

Linear Integer Programming (cont.)

- IP has $\frac{1}{3}n \times (n-1) \times (n-2) = \theta(n^3)$ constraints!!!
- The number of integral variables $n + \binom{n}{2} = \theta(n^2)$.
- Require excessive amount of memory and processing time.
- Largest solved instances have around 100 nodes.
- Denote the LP relaxation in which $0 \leq s_i, d_{ij} \leq 1$ by LP(large)

Mixed Integer Programming

- Replace $s_i, d_{ij} \in \{0,1\}$ by $s_i \in \{0,1\}, 0 \leq d_{ij} \leq 1$.
- Only n integral variables s_i.
- Equivalent to the IP.
- It still has $\theta(n^3)$ constraints.

Lemma

The MIP and IP have the same set of optimal basic solutions.

Sparse Metric

- The constraint $d_{ij} + d_{jk} \geq d_{ik}$ is equivalent to saying that d_{ij} is a (semi) metric.
- Not all constraints $d_{ij} + d_{jk} \geq d_{ik}$ are tight at the optimal solutions (for both IP and LP)
- Remove those unnessary (non-binding) constraints can reduce both time and memory requirements.
- Which constraints to keep/remove?

Sparse Metric (cont.)

A new compact Mixed Integer Programming, called MIP(Sparse):

$$d_{ij} + d_{jk} \geq d_{ik}, k \in [1..n] \rightarrow d_{ij} + d_{jk} \geq d_{ik}, k \in N_{\min}(i,j)$$

$N_{\min}(i,j) = N(i)$ if $d(i) < d(j)$, and $N(j)$ otherwise.

- There are at most $O(mn)$ constraints that is $O(n^2)$ for many networks of interests

$$\#\text{constraints} = |E| + \sum_{i<j} \min\{d(i), d(j)\} + \frac{n(n-1)}{2}$$

$$\leq n^2 + \sum_{i<j} \frac{d(i) + d(j)}{2} = n^2 + \frac{n-1}{2} \sum_{i=1}^{n} d(i) = O(mn)$$

- Denote the LP relaxation in which $d_{ij} \in [0,1]$ by LP(sparse)

Sparse Metric (cont.)

Both the integer programming and the relaxation of the sparse metric are equivalent to those of the original formula.

Sparse Metric (cont.)

Theorem

The MIP (sparse) and the IP (large) have the same set of optimal solutions.

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 (3) 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 $d_{ij} = d_{j,k} = 0$ that contradicts the constraint $d_{ij} + d_{jk} \geq d_{ik}$. It follows that rounding gives a feasible integral solution to the MIP.
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 (3) that $d_{ij} \leq d_{ik} + d_{kj} = 0$. Thus, the statement holds for all $t > 0$. □

Sparse Metric (cont.)

Theorem

The LP (sparse) and the LP (large) have the same set of optimal basic solutions.

Proof.

Let (s, d) be an optimal fraction solution of the LP relaxation of $\mathrm{MIP}_{\mathrm{vd}}$. 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 previous proof. The second statement comes from the definition of d'_{ij}.

Furthermore, we define $d^*_{ij} = \min\{d'_{ij}, 1\}$. 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 solution of IP(large).

Obviously, the minimum objective of the LP relaxation of MIP(sparse) is smaller or equal to that of IP(large).

Since, the objective values associate with (s, d^*) and (s, d) are the same, (s, d^*) must be a minimum solution of the LP relaxation of IP(large). $\square$

Branch & Cut Method

- A generalization of branch-and-bound, an efficient method to solve exactly many combinatorial optimization problems.
- After solving the LP relaxation, try to find a violated cut.
- If some violated cuts are found, we add them to the formulation and solve the LP again
- If none are found, we branch.

Vertex-Cut

- 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 the VC inequality

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

Separation Procedure

- A separation procedure finds a violated cut if any or return no such cut exists.
- Solve the maximum-flow (min-cut) problem in networks with both node and edge capacities.

Algorithm 1. Separation procedure for VC inequalities

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

Primal-heuristic

Provide upper bounds for pruning subproblems

- Let (s, d) be a fractional solution of the LP.
- Sort s_i in non-decreasing order $s_{i_1} \leq s_{i_2} \leq \ldots \leq s_{i_n}$.
- Round down all $s_{i_1}, s_{i_2}, \ldots, s_{i_k}$ to zero and round up $s_{i_{k+1}}, s_{i_{k+2}}, \ldots, s_{i_n}$ to one, where k runs from 1 to n.
- If the obtain solution is a β-vertex disruptor, a local search method is then used to refine the solution.

Primal-heuristic (cont.)

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.

Algorithm 1. Separation procedure for VC inequalities

```
for each pair (u, v) ∈ V × V do
    Construct a flow network G = (V, E) as follows
        Assign u and v as source and sink, respectively
        Each node k ∈ V has capacity s̄_k
        Every edge has capacity ∞
        if (u, v) ∈ E, then (u, v) has capacity zero.
    Find the maximum-flow (min-cut)
    if maximum-flow is less than d̄_uv, then
        Find the min vertex-cut S
        Add the VC inequality associated with S to MIP
    end if
end for
```

New Branh &Cut vs. the Original IP

MIP/IP Solver: GUROBI 4.0, Computer: Intel Xeon 2.93 Ghz, 64 GB memory.

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

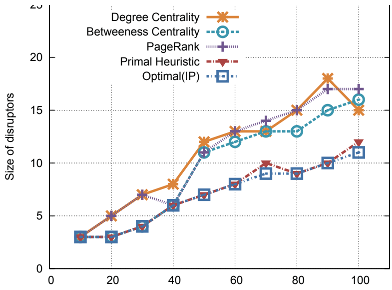
β-vertex disruptor vs. Centrality Measures

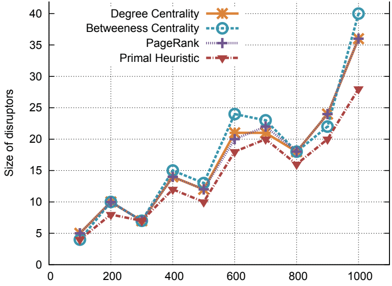
Compared methods:

1. *Degree Centrality*
2. *Betweenness Centrality*: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*: Default damping factor of 85%
4. *Primal heuristic*: Primal heuristics in the B& C algorithm

β-vertex disruptor vs. Centrality Measures

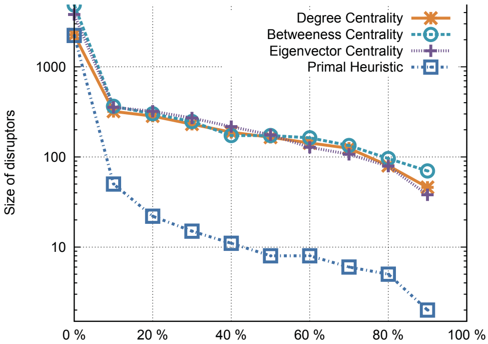
- Erdos-Rényi: edge exists with the same probability $p = 10\%$
- Power-law network: Barabasi's model, preferential attachment.





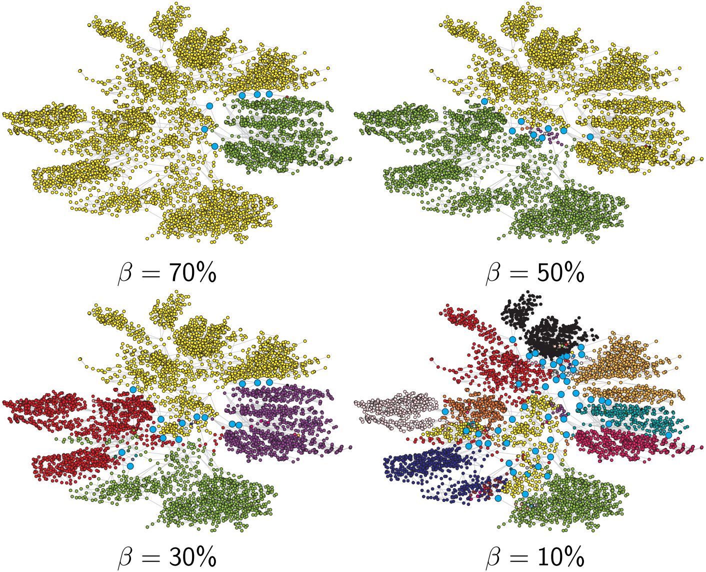
Disruptor size in Random networks (left) and Power-law networks (right) at disruptive level $\beta = 60\%$

Western Power Grid



Vulnerability assessment of Western States Power Grid

Western Power Grid (cont.)



Vertex disruptor at different disruption levels.

Summary

- Propose Sparse metric technique to substantially remove unwanted constraints
- A new Branch& Cut algorithm with a strong cutting plane
- Results can be applied for β-edge disruptor, critical nodes/edges detection and many connectivity-related problems
- Future work includes devising an efficient price-and-cut algorithm based on the sparse metric.

Thank you for listening!
Q&A