Survivable Network Design with Vertex and Edge Connectivity Constraints

Elham Sadeghi and Neng Fan

Systems and Industrial Engineering
University of Arizona

November 9, 2014
Survivable network: one of the most important issues in designing networks which have to remain functional when component failures occur.

Survivable network design: removal of specific number of vertices and edges does not disconnect the network.
- a railway network with both some tracks and depots being destroyed
- a traffic network with congestions on some roads and/or intersections
- telecommunication networks with failures of connections/stations

How to design a survivable network such that it is still connected after failures of some network components?
Edge- and vertex-connectivity

- $G = (V, E)$: $V$, set of vertices; $E$, set of edges.
- $l$-edge-connected: a graph which remains connected whenever fewer than $l$ edges are removed
  - edge-connectivity: the largest $l$ for which the graph is $l$-edge-connected
Edge- and vertex-connectivity

- $G = (V, E)$: $V$, set of vertices; $E$, set of edges.
- $l$-edge-connected: a graph which remains connected whenever fewer than $l$ edges are removed
  - edge-connectivity: the largest $l$ for which the graph is $l$-edge-connected
- $k$-vertex-connected: a graph which remains connected whenever fewer than $k$ vertices are removed
  - vertex-connectivity: the largest $k$ for which the graph is $k$-vertex-connected
Outline

- Mixed connectivity
- IP formulation
- Valid inequalities
- Cutting plane algorithm
- Numerical experiments
- Conclusion
Mixed connectivity

- **$(k, l)$-connected**: A graph $G$ is $(k, l)$-connected, if removal of $p$ vertices and $q$ edges, where

$$p \leq k - 1, q \leq l \quad \text{or} \quad p \leq k, q \leq l - 1,$$

does not disconnect the graph.
Mixed connectivity

- *(k, l)-connected*: A graph $G$ is *(k, l)-connected*, if removal of $p$ vertices and $q$ edges, where

$$p \leq k - 1, q \leq l \quad \text{or} \quad p \leq k, q \leq l - 1,$$

does not disconnect the graph.

- *(k, l)-connectivity*: A graph $G$ has *(k, l)-connectivity* if any pair of vertices are *(k, l)-connected* with at least one pair not being *(k + 1, l)*- nor *(k, l + 1)*-connected

- graph has *(k, 0)*-connectivity iff the vertex-connectivity is $k$
- graph has *(0, l)*-connectivity iff the edge-connectivity is $l$
Example

$k = 1, l = 1$

$k = 2, l = 1$
Network design problem

Problem: Given a graph $G = (V, E)$ with a nonnegative cost $c_{ij}$ associated with each edge $(i, j) \in E$, we want to find minimum-cost $(k, l)$-connected spanning subgraph.

- minimum spanning tree: $k = 0$, $l = 1$
- minimum-cost $k$-vertex connected subgraph: $k \geq 2$, $l = 0$
- minimum-cost $l$-edge connected subgraph: $k = 0$, $l \geq 2$

The minimum-cost $(k, l)$-connected spanning subgraph problem is NP-hard.
Network design problem

- **Problem**: Given a graph $G = (V, E)$ with a nonnegative cost $c_{ij}$ associated with each edge $(i, j) \in E$, we want to find minimum-cost $(k, l)$-connected spanning subgraph.
  - *minimum spanning tree*: $k = 0, l = 1$
  - *minimum-cost $k$-vertex connected subgraph*: $k \geq 2, l = 0$
  - *minimum-cost $l$-edge connected subgraph*: $k = 0, l \geq 2$

- The minimum-cost $(k, l)$-connected spanning subgraph problem is NP-hard.
**Theorem 1**

A graph is \((k, l)\)-connected if and only if the resulted graph after removal of any \(k\) vertices and \(l - 1\) edges is connected.

- **Lemma 1**: If a graph remains connected after removal of any \(k\) vertices and \(l - 1\) edges, then it also remains connected after removal of any \(k - 1\) vertices and any \(l\) edges (Erves and Zeronnik, 2010)

- **Proof of Theorem 1**:
  - \(p \leq k - 1, q \leq l\) or \(p \leq k, q \leq l - 1\)
  - \(p = k - 1, q = l\) or \(p = k, q = l - 1\)
  - according to Lemma, \(p = k, q = l - 1\)
Theorem 2
A graph $G = (V, E)$ is $(k, l)$-connected ($k, l \geq 1$) iff
(i) $G$ is $(k + 1)$-vertex-connected and
(ii) $G$ is $(k + l)$-edge-connected.

Lemma 2: (Menger’s Theorem, 1927)
- $G$ is $k$-vertex-connected iff every pair of vertices is connected by at least $k$ vertex-disjoint paths
- $G$ is $l$-edge-connected iff every two vertices are connected by at least $l$ edge-disjoint paths
Proof of Theorem 2

- **Lemma 3**: If a graph $G$ is $k$-vertex-connected, it is $k$-edge-connected.
- **Lemma 4**: If a graph $G$ is $(k, l)$-connected, it is $(0, k+l)$-connected.
  - $(k, l)$-connected $\Rightarrow$ resulted graph is still connected after removal of $k-1$ vertices and $l$ edges $\Rightarrow T^{hm} (k-1, l+1)$-connected $\Rightarrow (k-2, l+2)$-connected $\Rightarrow \cdots \Rightarrow (0, k+l)$-connected
Proof of Theorem 2

- **Lemma 3**: If a graph $G$ is $k$-vertex-connected, it is $k$-edge-connected.

- **Lemma 4**: If a graph $G$ is $(k, l)$-connected, it is $(0, k + l)$-connected.
  
  - $(k, l)$-connected $\Rightarrow$ resulted graph is still connected after removal of $k - 1$ vertices and $l$ edges $\Rightarrow$ Thm 1 $(k - 1, l + 1)$-connected $\Rightarrow$ $(k - 2, l + 2)$-connected $\Rightarrow \cdots \Rightarrow (0, k + l)$-connected

- **Proof of Theorem 2**: “$\Rightarrow$”
  
  - $G$ is $(k, l)$-connected so it is still connected after removal of any $k$ vertices. Thus it is $(k + 1)$-vertex-connected and condition (i) is proved.

  - According to Lemma 4, $G$ is $(0, k + l)$-connected. So $G$ is $(k + l)$-edge-connected and condition (ii) is proved.
Proof of Theorem 2 (cont.)

▶ “⇐”

For any pair $s, t \in V$, we want to show, they would be connected after removal of any $k$ vertices and $(l - 1)$ edges. According to condition (i), $G$ is $(k + 1)$-vertex connected:

- Case 1: By removal of any $k$ vertices, from condition (i), there exists at least one path $P$ from $s$ to $t$. If none of the $(l - 1)$ removed edges are in the path $P$, the connectivity between $s$ and $t$ is guaranteed.

- Case 2: By removal of any $k$ vertices $(v_1, \ldots, v_k)$ and $(l - 1)$ edges $(e_1, \ldots, e_{l-1})$, assume that $(k + 1)$-vertex-disjoint paths between $s$ and $t$ are destroyed.
- the degree of $s$ is at least $k + l$
- there exists a vertex $u_1$, such that $(s, u_1) \in E$
- there exists a vertex $u_2$ such that the edge $(u_1, u_2)$ exists in the resulted graph after removal of $v_1, \ldots, v_k, e_1, \ldots, e_{l-1}$
- continuing this process until we find $(u_m, t)$
- $(s, u_1), (u_1, u_2), \ldots, (u_m, t)$ forms a path between $s$ and $t$
Corollary of Theorem 2

- **Corollary**: A graph is \((k, l)\)-connected \((k \geq 0, l \geq 1)\) if and only if it has \(k + l\) edge-disjoint paths between every pair of its vertices, of which \(k + 1\) paths are vertex-disjoint.

- Beineke and Harry (1967) claimed properties for \((k, l)\)-connected subgraphs but Mader (1979) found a gap in their proof.
Stronger conditions on \((k, l)\)-connected

**Theorem 3**
Assume \(k \geq 1, l \geq 1\) and \(k \leq \frac{n}{2} - 1\), and the vertex set \(V\) of \(G\) is divided into \(V_1 = \{1, \cdots, n - k - 1\}\) and \(V_2 = \{n - k, \cdots, n\}\). A graph \(G\) is \((k, l)\)-connected if and only if it has \(k + l\) edge-disjoint paths between every vertex \(i \in V_1\) and every vertex \(j \in V_2\), of which \(k + 1\) paths are vertex-disjoint.

▶ **Proof.**  \(\Leftarrow\)
- **Case 1.** \(i \in V_1, j \in V_2\): \((k + l)\) edge-disjoint paths between \(i\) and \(j\), of which \((k + 1)\) are vertex-disjoint
- **Case 2.** \(i, j \in V_1\): There exists at least one vertex \(j'\) in \(V_2\). \(i, j\) are both connected to \(j'\).
- **Case 3.** \(i, j \in V_2\): If all removed \(k\) vertices are in \(V_2\) and removed edges are any ones in \(G\), there exists one vertex \(i' \in V\) such that both \(i, j\) are connected to it.

▶ Number of pairs considering in Theorem 2 (or its corollary) is \(\frac{n(n-1)}{2}\), while the number of pairs in Theorem 3 is \((n - k - 1)(k + 1)\).
IP formulation for \((k, l)\)-connected subgraph

\[
\begin{align*}
\text{min} & \quad \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_j y_{ij}^{(st)l'} - \sum_j y_{ji}^{(st)l'} = \begin{cases} 1, & i = s \\ -1, & i = t \\ 0, & i \in V \setminus \{s, t\} \end{cases} \\
& \quad \forall s \in V_1, t \in V_2, l' = 1, \ldots, k + l \\
\sum_{l' = 1}^{k+l} (y_{ij}^{(st)l'} + y_{ji}^{(st)l'}) & \leq x_{ij}, \forall s \in V_1, t \in V_2, (i,j) \in E \\
\sum_{l' = 1}^{k+1} \sum_j y_{ij}^{(st)l'} & \leq 1, \forall s \in V_1, t \in V_2, \forall i \in V \setminus \{s, t\} \\
y_{ij}^{(st)l'} & \in \{0, 1\}, \forall s \in V_1, t \in V_2, (i,j) \in E \cup E', l' = 1, \ldots, k + l \\
x_{ij} & \in \{0, 1\}, \forall (i,j) \in E \\
\end{align*}
\]

\( (k + l) \) paths for any pair of vertices \((s, t)\)  
\( (k + l) \) paths between \(s\) and \(t\) are edge-disjoint  
\( (k + 1) \) paths between \(s\) and \(t\) are vertex-disjoint
Valid inequalities

- Degree-cut:
  \[\sum_{j: j \in V} x_{ij} \geq (k + l), \; \forall i \in V\]

- Vertex cutset
  \[V(s, t)\] be the minimum \((s, t)\)-vertex-cutset
  \[|V(s, t)| < (k + 1)\]
  \[S \subset V \setminus V(s, t)\] and \[S' \subset V \setminus (S \cup V(s, t))\]
  \[\sum_{i \in S, j \in S'} x_{ij} \geq (k + 1) - |V(s, t)|\]
Cutting plane algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cutting Plane Algorithm for (k, l)-subgraph problem (Cutset)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
<td>A graph $G = (V, E)$ with cost matrix $(c_{ij})_{n \times n}$, and two integers $k, l$</td>
</tr>
<tr>
<td><strong>Output:</strong></td>
<td>minimum-cost $(k, l)$-subgraph of $G$</td>
</tr>
</tbody>
</table>

1: $t \leftarrow 0$

2: solve the problem consisting of objective functions and binary variables $x$ and $y$

3: obtain optimal solutions $x^t$

4: for each pair of vertices check the maximum edge-disjoint paths and vertex-disjoint paths by Ford-Folkerson algorithm

5: if maximum edge-disjoint paths $< k + l$ for a division $(S, V \setminus S)$ ($S \subset V, S \neq \emptyset$)

6: add cut $\sum_{i \in S, j \in V \setminus S} x^t_{ij} \geq k + l$ to the problem in Step 2

7: if maximum vertex-disjoint paths $< k + 1$

8: determine $V(s, t)$ and for a division $(S \setminus V(s, t), V \setminus (S \cup V(s, t)))$ ($S \subset V, S \neq \emptyset$)

9: add cut $\sum_{i \in S, j \in S'} x_{ij} \geq (k + 1) - |V(s, t)|$ to the problem in Step 2

10: goto Step 2, $t \leftarrow t + 1$

11: if no cut added in Steps 4-9, exit; $x^t$ is optimal.
Numerical experiments

- All MIP formulations are implemented in C++ using CPLEX 12.3 via IBM’s Concert Technology library, version 2.9.
- All experiments were performed on a Linux workstation with 4 Intel(R) Core(TM)2 CPU 2.40GHz processors and 8GB RAM.
### Numerical experiments

| $|V|$ | $|E|$ | $k$ | $l$ | Cutting plane algorithm formulation | $|V|$ | $|E|$ | $k$ | $l$ | Cutting plane algorithm formulation |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 45 | 1 | 3 | 0.02 | 0.38 | 40 | 780 | 1 | 3 | 1.27 |
| 5 | 0.02 | 0.46 | 3 | 3 | 9.59 |
| 7 | 0.01 | 0.83 | 5 | 0.08 | - |
| 3 | 3 | 0.02 | 0.7 | 9 | 0.03 | - |
| 5 | 0.01 | 1.32 | 5 | 3 | 7.28 |
| 5 | 0.01 | 1.5 | 5 | 1.1 | - |
| 15 | 105 | 1 | 3 | 0.02 | 0.8 | 9 | 0.09 | - |
| 5 | 1.67 | 13 | 1.43 | - |
| 11 | 0.02 | 2.38 | 9 | 3 | 1.21 | - |
| 13 | 0 | 2.63 | 5 | 0.62 | - |
| 3 | 3 | 0.05 | 2.98 | 9 | 1.55 | - |
| 9 | 0.01 | 3.65 | 13 | 0.04 | - |
| 11 | 0.01 | 4.65 | 11 | 3 | 0.78 | - |
| 7 | 5 | 0.01 | 3.01 | 9 | 0.03 | - |
| 7 | 0 | 4.17 | 13 | 0.02 | - |
| 11 | 3 | 0 | 5.25 | 15 | 3 | 0.5 | - |
| 20 | 190 | 1 | 3 | 0.15 | 18.02 | 9 | 0.03 | - |
| 3 | 3 | 0.09 | 30.18 | 13 | 0.03 | - |
| 5 | 0.04 | 118.9 | 17 | 3 | 0.03 | - |
| 5 | 3 | 0.11 | 120.57 | 9 | 0.04 | - |
| 9 | 0.02 | 97.82 | 11 | 0.02 | - |
| 9 | 0.02 | 120.99 | 21 | 3 | 0.04 | - |
| 9 | 0.01 | 410.84 | 5 | 0.04 | - |
| 11 | 3 | 0.01 | 86.21 | 7 | 0.02 | - |
| 7 | 0.02 | 408 | 23 | 3 | 0.05 | - |
| 15 | 3 | 0.01 | 410.85 | 5 | 0.08 | - |
| 30 | 435 | 1 | 3 | 0.96 | 28.6 | 50 | 1225 | 1 | 3 | 1.53 |
| 3 | 3 | 0.52 | 112.66 | 3 | 3 | 9.53 |
| 5 | 0.49 | 416.3 | 9 | 0.18 | - |
| 9 | 0.13 | 658.2 | 11 | 3 | 3.72 | - |
| 5 | 3 | 0.92 | 365.56 | 9 | 0.67 | - |
| 5 | 0.29 | 471.05 | 15 | 3 | 1.73 | - |
| 9 | 0.09 | - | 5 | 0.88 | - |
| 13 | 0.03 | - | 17 | 3 | 2.15 | - |
| 9 | 3 | 0.17 | - | 5 | OM | - |
| 5 | 0.09 | - |
| 9 | 0.02 | - |
| 13 | 0.02 | - |
| 11 | 3 | 0.07 | - |
| 9 | 0.02 | - |
| 13 | 0.02 | - |
| 15 | 3 | 0.04 | - |
| 9 | 0.02 | - |
| 13 | 0.03 | - |
| 17 | 3 | 0.04 | - |
| 9 | 0.03 | - |
| 11 | 0.03 | - |
| 21 | 3 | 0 | - |
| 5 | 0.02 | - |
| 7 | 0.02 | - |
| 23 | 3 | 0.03 | - |

Note: “OM” means out of memory.
Conclusion

- proposed the network design problem with mixed connectivity requirements, as the minimum-cost \((k, l)\)-connected subgraph problem
- proposed and proved conditions by edge/vertex-disjoint paths for this problem
- developed stronger conditions, and formulated as integer programs
- designed a cutting plane algorithm
Thank You!

Elham Sadeghi
sadeghi@email.arizona.edu
University of Arizona, Tucson, AZ