Cyber Security Analysis of Power Networks by Hypergraph Cut Algorithms

Yutaro Yamaguchi\textsuperscript{1}, Anna Ogawa\textsuperscript{2},
Akiko Takeda\textsuperscript{1}, Satoru Iwata\textsuperscript{1}

1. Department of Mathematical Informatics, University of Tokyo
2. Department of Administration Engineering, Keio University

SmartGridComm 2014 @Venice November 4, 2014
Cyber Threat to Power Networks

Power network

State

Measurement

Estimate to control

System administrator
Cyber Threat to Power Networks

- Power network
- State
- False data injection
- Measurement
- Misrecognize!!
- System administrator
Outline

• Model and Problem Definitions
  — Undetectable (false data injection) attacks
  — Sparsest attack problem (Global security analysis)
  — Security index problem (Local security analysis)

• Existing Methods vs. Proposed Methods
  — Approx. by LP-relaxation
  — Approx. by min-cut in graphs
  — Exact by min-cut in auxiliary graphs
  — Exact by min-cut in hypergraphs (Proposed)

• Experimental Results
Outline

• Model and Problem Definitions
  — Undetectable (false data injection) attacks
  — Sparsest attack problem (Global security analysis)
  — Security index problem (Local security analysis)

• Existing Methods vs. Proposed Methods
  — Approx. by LP-relaxation
  — Approx. by min-cut in graphs
  — Exact by min-cut in auxiliary graphs
  — Exact by min-cut in hypergraphs (Proposed)

• Experimental Results
Linearized State Estimation Model

Power network

State

Measurement

Estimate

System administrator
Linearized State Estimation Model

$G = (V, A)$: directed graph

System administrator
Linearized State Estimation Model

Power network

\[ G = (V, A): \text{directed graph} \]

State: Voltage angle on each node
\[ \theta \in \mathbb{R}^V \]

Measurement

Estimate

System administrator
Linearized State Estimation Model

\[ G = (V, A): \text{directed graph} \]

Power network

\[ \theta \in \mathbb{R}^V \]

State: Voltage angle on each node

Measurement: Active power on arcs & nodes

\[ z \in \mathbb{R}^{A\cup V} \]

Estimate

System administrator
Linearized State Estimation Model

State: Voltage angle on each node
\[ \theta \in \mathbb{R}^V \]

Measurement: Active power on arcs & nodes
\[ z \in \mathbb{R}^{A\cup V} \]

Estimate
\[ z = H\theta \]

System administrator

\[ G = (V, A) \text{: directed graph} \quad H \in \mathbb{R}^{(A\cup V) \times V} \]
False Data Injection

State on each node: $\theta \in \mathbb{R}^V$

Measurement on arcs & nodes: $z \in \mathbb{R}^{A \cup V}$

$z = H\theta$

$H \in \mathbb{R}^{(A \cup V) \times V}$

System administrator

False data injection $\Delta z \in \mathbb{R}^{A \cup V}$

$z + \Delta z$
False Data Injection

Power network

State on each node: \( \theta \in \mathbb{R}^V \)

\[
z = H\theta
\]

\( H \in \mathbb{R}^{(A \cup V) \times V} \)

Measurement on arcs & nodes:

\( z \in \mathbb{R}^{A \cup V} \)

\[z + \Delta z\]

False data injection

\( \Delta z \in \mathbb{R}^{A \cup V} \)

Anything is OK.

No such \( \tilde{\theta} \) !!

Something wrong!!

Detectable!!

System administrator
False Data Injection

State on each node: $\theta \in \mathbb{R}^V$

Measurement on arcs & nodes:

$z = H\theta$

$H \in \mathbb{R}^{(A \cup V) \times V}$

$z \in \mathbb{R}^{A \cup V}$

$z + \Delta z$

False data injection

$\Delta z \in \mathbb{R}^{A \cup V}$

$\Delta z = H\Delta \theta$

($\Delta \theta \in \mathbb{R}^V$)

OK, $\tilde{\theta} = \theta + \Delta \theta$ !!!

Misrecognize!!

System administrator
Undetectable (False Data Injection) Attack

(Liu, Ning, Reiter 2009)

A difference $\Delta z \in \mathbb{R}^{AUV}$ of measurement values is called an undetectable attack.

$$\begin{align*}
\text{def} & \quad \Leftrightarrow \exists \Delta\theta \in \mathbb{R}^V \text{ s.t. } \Delta z = H\Delta\theta \\
\text{Actual:} & \quad z = H\theta \\
\text{Attack:} & \quad \Delta z = H\Delta\theta \\
\text{Misrecognition:} & \quad z + \Delta z = H(\theta + \Delta\theta)
\end{align*}$$
Sparsest Attack (Global Security)  
(Liu, Ning, Reiter 2009)

A nonzero undetectable attack $H\Delta \theta \in \mathbb{R}^{A \cup V} \setminus \{0\}$ with the fewest nonzero entries (attacked points)

\[
\begin{align*}
\text{minimize} & \quad ||H\Delta \theta||_0 \\
\text{subject to} & \quad H\Delta \theta \neq 0
\end{align*}
\]

Attacking many points  
→ Easy to prevent

\[
\Delta z = H\Delta \theta
\]

Attacking few points  
→ Hard to prevent

\[
\Delta z = H\Delta \theta
\]
The **minimum number of nonzero entries** of an **undetectable attack** $H\Delta \theta \in \mathbb{R}^{A \cup V}$ to attack a specified arc or node $k \in A \cup V$

minimize $\|H\Delta \theta\|_0$
subject to $H_k \Delta \theta \neq 0$

Attacking **many** points $\rightarrow$ **Easy to prevent**

Attacking **few** points $\rightarrow$ **Hard to prevent**
Sparsest Attack and Security Index

Fact

Any **sparsest attack** attains the **security indices** of the arcs and nodes to be attacked.

Δ\(z\)\(^k\) is a sparsest attack.

\((\text{security index of } k) = \|\Delta z\|_0\)
Sparsest Attack and Security Index

Fact

Any **sparsest attack** attains the **security indices** of the arcs and nodes to be attacked.

\[ \Delta z \text{ is a sparsest attack.} \]

\[ (\text{security index of } k) = \| \Delta z \|_0 \]

A **sparsest attack** can be found by computing the **security indices** of **ALL** arcs and nodes!
Sparsest Attack and Security Index

**Fact**
Any sparsest attack attains the security indices of the arcs and nodes to be attacked.

**Fact**
The security index of a node is equal to the minimum security index among its incident arcs'.

\[ (S.I. \text{ of } v) = \min_{i=1,2,3} (S.I. \text{ of } a_i) \]

A sparsest attack can be found by computing the security indices of ALL arcs!!!
Outline

• Model and Problem Definitions
  — Undetectable (false data injection) attacks
  — Sparsest attack problem (Global security analysis)
  — Security index problem (Local security analysis)

• Existing Methods vs. Proposed Methods
  — Approx. by LP-relaxation
  — Approx. by min-cut in graphs
  — Exact by min-cut in auxiliary graphs
  — Exact by min-cut in hypergraphs (Proposed)

• Experimental Results
Solution Methods for **Security Index**

**Approx. by min-cut**
(Sou, Sandberg, Johansson 2011)

**Approx. by LP-relax**
(Sou, Sandberg, Johansson 2013)

**Exact by min-cut**
*in auxiliary graph*
(Hendrickx, Johansson, Junger, Sandberg, Sou 2012)

**Exact by min-cut**
*in hypergraph*
Solution Methods for Sparsest attack

Approx. by min-cut
(Sou, Sandberg, Johansson 2011)

Approx. by LP-relax
(Sou, Sandberg, Johansson 2013)

Exact by min-cut
in auxiliary graph
(Hendrickx, Johansson, Junger, Sandberg, Sou 2012)

Security indices of ALL arcs

Exact by min-cut
in hypergraph

Single computation!!
Why min-cut?
Elementary Attack

An undetectable attack $H\Delta \theta \in \mathbb{R}^{A\cup V}$ is \textit{elementary}. \iff $\Delta \theta \in \{0, 1\}^V$

Lemma (Sou et al. 2011)

For any arc or node, there exists an \textbf{elementary attack} attaining the \textbf{security index}.

→ Consider only \textbf{elementary attacks}

→ Assign 0 or 1 to each node (\textit{Bipartition the node set }$V$)
Elementary Attack

An undetectable attack $H \Delta \theta \in \mathbb{R}^{A \cup V}$ is **elementary**. $\iff \Delta \theta \in \{0, 1\}^V$

**Fact**

An arc $uv \in A$ is **attacked** in an elementary attack. $\iff \Delta \theta(u) \neq \Delta \theta(v)$

$\iff uv$ is **cut off** by separating 0-nodes and 1-nodes.

→ # of **attacked arcs** = # of **arcs cut off** = **cut capacity**

→ Approx. by min-cut (Sou et al. 2011)
Elementary Attack

An undetectable attack \( H\Delta \theta \in \mathbb{R}^{A \cup V} \) is **elementary**. \( \iff \Delta \theta \in \{0, 1\}^V \)

**Fact**

An arc \( uv \in A \) is attacked in an **elementary attack**. \( \iff \Delta \theta(u) \neq \Delta \theta(v) \)

\( \iff uv \) is cut off by separating 0-nodes and 1-nodes.

→ # of attacked arcs = # of arcs cut off = cut capacity

→ Approx. by min-cut (Sou et al. 2011) How about attacked nodes?
Counting **Attacked Nodes**

Construct auxiliary graph
(Hendrickx, Johansson, Junger, Sandberg, Sou 2012)

**Use hypergraph**

- Large size
- A **sparsest attack** requires (# of arcs) min-cut comps.

- No additional node
- A **sparsest attack** can be found by **single min-cut computation!!**
Hypergraphs

Each edge connects \textbf{two nodes}.

Each hyperedge connects an \textbf{arbitrary number of nodes}.
Construction of Hypergraph

- Start with the input graph (ignoring the direction)
Construction of Hypergraph

• Start with the input graph (ignoring the direction)
• For each node $v \in V$, add a hyperedge consisting of the node $v$ itself and all neighbors of $v$. 

[Diagram of hypergraph with a node $v$ and its neighbors]
Construction of Hypergraph

• Start with the input graph (ignoring the direction)
• For each node \( v \in V \), add a hyperedge consisting of the node \( v \) itself and all neighbors of \( v \).
Construction of Hypergraph


Cut capacity in this hypergraph
\[ \Delta \theta = 1 \]
\[ \Delta \theta = 0 \]

# of arcs & nodes to be attacked

\[ \Delta \theta = 1 \]
\[ \Delta \theta = 0 \]
Computing **Security Index**

Computing the *security index of an arc* $a = st \in A$

→ Finding a **minimum $s$—$t$ cut** in a hypergraph

$$\Delta \theta = 1$$

$$\Delta \theta = 0$$

Fact

An arc $st \in A$ **is attacked**.

$\iff \Delta \theta(s) \neq \Delta \theta(t)$

$\iff st$ **is cut off**.
Computing Security Index

Computing the security index of an arc \( a = st \in A \)

\[ \rightarrow \text{Finding a minimum } s-t \text{ cut} \text{ in a hypergraph} \]


For any arc in any directed graph \( G = (V, A) \), one can compute the security index in \( O(|V||A|) \) time.

- By a **hypergraph min s—t cut** algorithm (Pistorius, Minoux 2003)
- The same order as the existing exact method (Hendrickx et al. 2012), but **faster in practice** because their auxiliary graph is large.
Finding Sparsest Attack

Finding a **sparsest attack** in the whole network

→ Finding a **minimum cut** in a hypergraph

*Theorem (Y.-O.-T.-I. 2014)*

For any directed graph $G = (V, A)$, one can find a **sparsest attack** in $O(|V||A| + |V|^2 \log|V|)$ time.

- By a **hypergraph min-cut** algorithm *(Klimmek, Wagner 1996)*
- **Essential speeding up!!**
  
  Applying the existing exact method *(Hendrickx et al. 2012)* to all arcs
  
  $\rightarrow O(|V||A|^2)$ time
Outline

• Model and Problem Definitions
  — Undetectable (false data injection) attacks
  — Sparsest attack problem (Global security analysis)
  — Security index problem (Local security analysis)

• Existing Methods vs. Proposed Methods
  — Approx. by LP-relaxation
  — Approx. by min-cut in graphs
  — Exact by min-cut in auxiliary graphs
  — Exact by min-cut in hypergraphs (Proposed)

• Experimental Results
Problems and Solution Methods

• Finding a **sparsest attack** in the whole network
  — **hyp. global min. cut**: exact method by hypergraph min-cut

• Computing the **security index of an arc** \( a \in A \)
  — **hyp. min. s-t cut**: exact method by hypergraph min-cut
  — **min. s-t cut exact**: exact method by min-cut in auxiliary graph (Hendrickx et al. 2012)
  — **min. s-t cut relax**: approx. method by min-cut in input graph (Sou et al. 2011)
  — **L1-relax (LP)**: approx. method by LP-relaxation (Sou et al. 2013)
  — **L0-exact (MIP)**: exact method by MIP solver (CPLEX)

\[
\begin{align*}
\text{minimize} & \quad \|H\Delta \theta\|_0 \\
\text{subject to} & \quad H_a \Delta \theta \neq 0
\end{align*}
\]
Computational Time for Security Index

Proposed method

Fails to obtain an exact solution for 10~20% arcs

About 1.8 times faster on average than the existing exact method.
Proposed methods Predominantly fastest!!

Computational Time for Sparsest Attack

![Graph showing computational time for various methods across different data sets and system types.](image)

- IEEE data sets
- Tokyo-Tohoku (East Japan)
- Polish systems
Conclusion

• A sparsest attack and the security index of each measurement point are significant security criteria for power networks.

• A sparsest attack can be found fast and exactly by finding a minimum cut in a hypergraph.

• The security index of each measurement point can be computed fast and exactly by finding a minimum $s-t$ cut in a hypergraph.