Power grid vulnerability, new models, algorithms and computing

Daniel Bienstock
Abhinav Verma

Columbia University, New York

February, 2009
Talk Outline

- Lossless power flows
- Solving the N-k problem
- A better model for the N-k problem
Power flow model - lossless model

We are given a network with nodes $\mathcal{N}$ ("buses") and arcs $\mathcal{A}$ ("lines"): 

- A set of $\mathcal{G}$ of supply nodes (the "generators"); each generator $i$ has an "operating range" $0 \leq S^L_i \leq S^U_i$.

- A set $\mathcal{D}$ of demand nodes (the "loads"); for each load $i$ a "maximum demand" $0 \leq D^\text{max}_i$.

- For each arc $(i, j)$ a parameter $x_{ij}$.

Basic problem: operate network so as to maximize amount of delivered power
Power flow model - lossless model

We are given a network with nodes $\mathcal{N}$ ("buses") and arcs $\mathcal{A}$ ("lines"):

- A set of $\mathcal{G}$ of **supply** nodes (the "generators"); each generator $i$ has an "operating range" $0 \leq S_i^L \leq S_i^U$,

- A set $\mathcal{D}$ of **demand** nodes (the "loads"); for each load $i$ a "maximum demand" $0 \leq D_i^{\text{max}}$.

- For each arc $(i, j)$ a parameter $x_{ij}$.

Basic problem: operate network so as to maximize amount of delivered power.
Power flow model - lossless model

We are given a network with nodes \( \mathcal{N} \) (“buses”) and arcs \( \mathcal{A} \) (“lines”):

- A set of \( \mathcal{G} \) of supply nodes (the “generators”); each generator \( i \) has an “operating range” \( 0 \leq S_i^l \leq S_i^u \).

- A set \( \mathcal{D} \) of demand nodes (the “loads”); for each load \( i \) a “maximum demand” \( 0 \leq D_i^{\text{max}} \).

- For each arc \( (i, j) \) a parameter \( x_{ij} \).

Basic problem: operate network so as to maximize amount of delivered power.
Power flow model - lossless model

We are given a network with nodes $\mathcal{N}$ ("buses") and arcs $\mathcal{A}$ ("lines"): 

- A set of $\mathcal{G}$ of supply nodes (the “generators”); each generator $i$ has an “operating range” $0 \leq S_i^L \leq S_i^U$,  

- A set $\mathcal{D}$ of demand nodes (the “loads”); for each load $i$ a “maximum demand” $0 \leq D_i^{\max}$.  

- For each arc $(i, j)$ a parameter $x_{ij}$.  

Basic problem: operate network so as to maximize amount of delivered power
Feasible power flows

Suppose we **choose** an output level \( b_i > 0 \) for each generator \( i \), and a demand level \(-b_i > 0\) for each load \( i \).

Write \( b_i = 0 \) for each other node \( i \).

A **power flow** is a solution \( f, \theta \) to:

- \( \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i \), for all \( i \) (so must assume \( \sum_i b_i = 0 \))
- \( x_{ij} f_{ij} - \sin(\theta_i - \theta_j) = 0 \), for all \((i, j)\), (physics)
- \( |\theta_i - \theta_j| \leq \frac{\pi}{2} \) (“stability”)

A difficult system?
Feasible power flows

Suppose we choose an output level \( b_i > 0 \) for each generator \( i \), and a demand level \(-b_i > 0\) for each load \( i \).

Write \( b_i = 0 \) for each other node \( i \).

A power flow is a solution \( f, \theta \) to:

- \( \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i \), for all \( i \) (so must assume \( \sum_i b_i = 0 \))
- \( x_{ij} f_{ij} - \sin(\theta_i - \theta_j) = 0 \), for all \( (i, j) \), (physics)
- \( |\theta_i - \theta_j| \leq \frac{\pi}{2} \) ("stability")

A difficult system?
Feasible power flows

Suppose we choose an output level $b_i > 0$ for each generator $i$, and a demand level $-b_i > 0$ for each load $i$.

Write $b_i = 0$ for each other node $i$.

A power flow is a solution $f, \theta$ to:

- $\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i$, for all $i$ (so must assume $\sum_i b_i = 0$)
- $x_{ij}f_{ij} - \sin(\theta_i - \theta_j) = 0$, for all $(i, j)$, (physics)
- $|\theta_i - \theta_j| \leq \frac{\pi}{2}$ ("stability")

A difficult system?
Feasible power flows

Suppose we choose an output level $b_i > 0$ for each generator $i$, and a demand level $-b_i > 0$ for each load $i$.

Write $b_i = 0$ for each other node $i$.

A power flow is a solution $f, \theta$ to:

- $\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i$, for all $i$ (so must assume $\sum_i b_i = 0$)
- $x_{ij} f_{ij} - \sin(\theta_i - \theta_j) = 0$, for all $(i, j)$, (physics)
- $|\theta_i - \theta_j| \leq \frac{\pi}{2}$ (“stability”)

A difficult system?
Some facts

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i \]
\[ x_{ij} f_{ij} - \sin(\theta_i - \theta_j) = 0, \text{ for all } (i, j), \]
\[ |\theta_i - \theta_j| \leq \frac{\pi}{2} \]

**Lemma:** If the above system is feasible (for a given choice of \( b \)) then the solution is unique.

**Theorem:** The set of vectors \( b \), for which the above system is feasible, is convex.
Some facts

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i \]
\[ x_{ij}f_{ij} - \sin(\theta_i - \theta_j) = 0, \text{ for all } (i, j), \]
\[ |\theta_i - \theta_j| \leq \frac{\pi}{2} \]

**Lemma:** If the above system is feasible (for a given choice of \( b \)) then the solution is unique.

**Theorem:** The set of vectors \( b \), for which the above system is feasible, is *convex*. 
Some facts

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i \]

\[ x_{ij} f_{ij} - \sin(\theta_i - \theta_j) = 0, \text{ for all } (i, j), \]

\[ |\theta_i - \theta_j| \leq \frac{\pi}{2} \]

**Lemma:** If the above system is feasible (for a given choice of \( b \)) then the solution is unique.

**Theorem:** The set of vectors \( b \), for which the above system is feasible, is **convex**.
A basic consequence

Throughput maximization:

\[ \sum_{i \in G} b_i \]

s.t. \( b \) feasible,

is a \textit{convex optimization} problem!

... but we don’t know the feasible set.
A basic consequence

Throughput maximization:

\[ \sum_{i \in G} b_i \]

s.t. \( b \) feasible,

is a **convex optimization** problem!

... but we don’t know the feasible set.
A basic consequence

Throughput maximization:

$$\sum_{i \in G} b_i$$

s.t. \( b \) feasible,

is a \textit{convex optimization} problem!

... but we don’t know the feasible set.
First idea (bad?) (trivial?) (obvious?)

A **cutting-plane** algorithm which uses an approximation to the feasible set.

Repeat:

- Compute a vector $\hat{b}$ which maximizes throughput over the feasible set
- If $\hat{b}$ is actually feasible we are done, else *compute* a separating hyperplane $c^T b \leq \beta$
- Add $c^T b \leq \beta$ to the approximation to the feasible set
First idea (bad?) (trivial?) (obvious?)

A **cutting-plane** algorithm which uses an approximation to the feasible set.

**Repeat:**

- Compute a vector $\hat{b}$ which maximizes throughput over the feasible set
- If $\hat{b}$ is actually feasible we are done, else **compute** a separating hyperplane $c^T b \leq \beta$
- Add $c^T b \leq \beta$ to the approximation to the feasible set

... in the works
First idea (bad?) (trivial?) (obvious?)

A cutting-plane algorithm which uses an approximation to the feasible set.

Repeat:

- Compute a vector $\hat{b}$ which maximizes throughput over the feasible set
  - If $\hat{b}$ is actually feasible we are done, else compute a separating hyperplane $c^T b \leq \beta$
  - Add $c^T b \leq \beta$ to the approximation to the feasible set

... in the works
First idea (bad?) (trivial?) (obvious?)

A cutting-plane algorithm which uses an approximation to the feasible set.

Repeat:

- Compute a vector $\hat{b}$ which maximizes throughput over the feasible set
- If $\hat{b}$ is actually feasible we are done, else compute a separating hyperplane $c^T b \leq \beta$
- Add $c^T b \leq \beta$ to the approximation to the feasible set
First idea (bad?) (trivial?) (obvious?)

A cutting-plane algorithm which uses an approximation to the feasible set.

Repeat:

- Compute a vector $\hat{b}$ which maximizes throughput over the feasible set
- If $\hat{b}$ is actually feasible we are done, else compute a separating hyperplane $c^Tb \leq \beta$
- Add $c^Tb \leq \beta$ to the approximation to the feasible set

... in the works
Convex duality

Throughput maximization problem:

\[ t^* = \max \sum_{i \in G} b_i \]

subject to

\[ (\alpha_i) \quad \sum_{(i,j)} f_{ij} - \sum_{(j,i)} f_{ji} - b_i = 0 \quad \forall i \in \mathcal{N} \]

\[ (\beta_{ij}) \quad \theta_i - \theta_j - \sin^{-1}(x_{ij}f_{ij}) = 0 \quad \forall (i, j) \in \mathcal{A} \]

\[ (p_i) \quad 0 \leq b_i \leq S_i^{\text{max}} \quad \forall i \in \mathcal{G} \]

\[ (q_j) \quad -D_j^{\text{max}} \leq b_j \leq 0 \quad \forall j \in \mathcal{D} \]
The dual: always a convex program

\[ d^* = \min \sum_{i \in G} p_i S_i^{max} - \sum_{j \in D} q_j D_j^{max} + \sum_{(i,j) \in A} g(\nu_{ij}, \beta_{ij}) \]

subject to

\[ \sum_{(i,j)} \beta_{ij} - \sum_{(j,i)} \beta_{ji} = 0 \quad \forall i \in \mathcal{N} \]

\[ \alpha_i + p_i \geq 1 \quad \forall i \in \mathcal{G} \]

\[ -\alpha_j + q_j \geq 0 \quad \forall j \in \mathcal{D} \]

\[ \alpha_i - \alpha_j - x_{ij}\nu_{ij} = 0 \quad \forall (i,j) \in \mathcal{A} \]

\[ p \geq 0, q \geq 0 \]
A horrible slide

\[
g(\nu, \beta) = \begin{cases} 
\max \left\{ \nu \sqrt{1 - \frac{\beta^2}{\nu^2}} - \beta \cos^{-1}\left(\frac{\beta}{\nu}\right), -\nu + \frac{\pi}{2} \beta \right\} & 0 \leq \beta \leq \nu \\
-\nu + \frac{\pi}{2} \beta & 0 \leq \nu < \beta \\
-\nu + \frac{\pi}{2} \beta & \beta \leq 0, \nu \geq 0 \\
\max \left\{ -\nu \sqrt{1 - \frac{\beta^2}{\nu^2}} + \beta \cos^{-1}\left(\frac{\beta}{\nu}\right), \nu - \frac{\pi}{2} \beta \right\} & \nu \leq \beta \leq 0 \\
\nu - \frac{\pi}{2} \beta & \beta < \nu \leq 0 \\
\nu - \frac{\pi}{2} \beta & \beta \geq 0, \nu \leq 0 
\end{cases}
\]
On-going work!

- Currently solving dual using SNOPT
- Results: method scales well to problems with thousands of nodes
- Duality gap: small (not always zero)
- Also: sequential linearization scheme
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput.

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{\text{min}}$ of total demand
- “Small” set of arcs = very small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput.

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{\text{min}}$ of total demand
- “Small” set of arcs = very small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
The N-k problem in power grids

Given a power grid modeled by a network, delete a **small** set of arcs, such that in the resulting network all feasible flows have **small** throughput

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{min}$ of total demand
- “Small” set of arcs = **very** small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{min}$ of total demand
- “Small” set of arcs = very small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput.

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{min}$ of total demand
- “Small” set of arcs = very small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput.

- Used to model “natural” blackouts.
- “Small” throughput: we satisfy less than some amount $D_{\min}$ of total demand.
- “Small” set of arcs = very small.
- Delete 1 arc = the “N-1” problem.
- Of interest: delete $k = 2, 3, 4, \ldots$ edges.

Naive enumeration blows up.
The N-k problem in power grids

Given a power grid modeled by a network, delete a small set of arcs, such that in the resulting network all feasible flows have small throughput.

- Used to model “natural” blackouts
- “Small” throughput: we satisfy less than some amount $D_{\text{min}}$ of total demand
- “Small” set of arcs = very small
- Delete 1 arc = the “N-1” problem
- Of interest: delete $k = 2, 3, 4, \ldots$ edges
- Naive enumeration blows up
Linear power flow model

We are given a network $G$ with:

- A set of $S$ of supply nodes (the “generators”); for each generator $i$ an “operating range” $0 \leq S^L_i \leq S^U_i$.

- A set $D$ of demand nodes (the “loads”); for each load $i$ a “maximum demand” $0 \leq D^\text{max}_i$.

- For each arc $(i, j)$ values $x_{ij}$ and $u_{ij}$.
Linear power flow model

We are given a network $G$ with:

- A set of $S$ of supply nodes (the “generators”); for each generator $i$ an “operating range” $0 \leq S^L_i \leq S^U_i$.

- A set $D$ of demand nodes (the “loads”); for each load $i$ a “maximum demand” $0 \leq D^\text{max}_i$.

- For each arc $(i, j)$ values $x_{ij}$ and $u_{ij}$. 
Linear power flow model

We are given a network $G$ with:

- A set of $S$ of supply nodes (the “generators”); for each generator $i$ an “operating range” $0 \leq S^L_i \leq S^U_i$.

- A set $D$ of demand nodes (the “loads”); for each load $i$ a “maximum demand” $0 \leq D^\text{max}_i$.

- For each arc $(i, j)$ values $x_{ij}$ and $u_{ij}$. 
A **power flow** is a solution \( f, \theta \) to:

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i, \quad \text{where}
\]

\[
S_i^L \leq b_i \leq S_i^U \quad \text{OR} \quad b_i = 0, \quad \text{for each } i \in S,
\]

\[
0 \leq -b_i \leq D_i^{\max} \quad \text{for } i \in D,
\]

and \( b_i = 0 \), otherwise.

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j). \quad \text{(Ohm's equation)}
\]

**Lemma** Given a choice for \( b \) with \( \sum_i b_i = 0 \), the system has a unique solution.

The solution is **feasible** if \( |f_{ij}| \leq u_{ij} \) for every \( (i, j) \).

Its throughput is \( \sum_{i \in D} -b_i \).
Feasible power flows

A **power flow** is a solution $f$, $\theta$ to:

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i, \text{ where }
\]
\[
S_i^L \leq b_i \leq S_i^U \quad \text{OR} \quad b_i = 0, \text{ for each } i \in S,
\]
\[
0 \leq -b_i \leq D_i^{max} \text{ for } i \in D,
\]
and $b_i = 0$, otherwise.

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j). \text{ (Ohm’s equation)}
\]

**Lemma** Given a choice for $b$ with $\sum_i b_i = 0$, the system has a **unique** solution.

The solution is **feasible** if $|f_{ij}| \leq u_{ij}$ for every $(i, j)$.

Its **throughput** is $\sum_{i \in D} -b_i$.
Feasible power flows

A **power flow** is a solution $f$, $\theta$ to:

1. $\sum ij f_{ij} - \sum ij f_{ji} = b_i$, for all $i$, where
   
   $S_i^L \leq b_i \leq S_i^U$ OR $b_i = 0$, for each $i \in S$,

   $0 \leq -b_i \leq D_i^{\text{max}}$ for $i \in D$,

   and $b_i = 0$, otherwise.

2. $x_{ij}f_{ij} - \theta_i + \theta_j = 0$ for all $(i, j)$. (Ohm’s equation)

**Lemma** Given a choice for $b$ with $\sum_i b_i = 0$, the system has a **unique** solution.

The solution is **feasible** if $|f_{ij}| \leq u_{ij}$ for every $(i, j)$.

Its **throughput** is $\sum_{i \in D} -b_i$. 
A **power flow** is a solution \( f, \theta \) to:

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i,
\]

where

\[
S^L_i \leq b_i \leq S^U_i \quad \text{OR} \quad b_i = 0, \text{ for each } i \in S,
\]

\[
0 \leq -b_i \leq D^{\text{max}}_i \quad \text{for } i \in D,
\]

and \( b_i = 0 \), otherwise.

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \quad \text{for all } (i, j). \quad \text{(Ohm’s equation)}
\]

**Lemma** Given a choice for \( b \) with \( \sum_i b_i = 0 \), the system has a **unique** solution.

The solution is **feasible** if \(|f_{ij}| \leq u_{ij}\) for every \((i, j)\).

Its throughput is \( \sum_{i \in D} -b_i \).
Feasible power flows

A **power flow** is a solution $f$, $\theta$ to:

- $\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i$, for all $i$, where $S_i^L \leq b_i \leq S_i^U$ OR $b_i = 0$, for each $i \in S$,
- $0 \leq -b_i \leq D_i^{max}$ for $i \in D$,
- and $b_i = 0$, otherwise.

- $x_{ij}f_{ij} - \theta_i + \theta_j = 0$ for all $(i, j)$. (Ohm’s equation)

**Lemma** Given a choice for $b$ with $\sum_i b_i = 0$, the system has a **unique** solution.

The solution is **feasible** if $|f_{ij}| \leq u_{ij}$ for every $(i, j)$.

Its **throughput** is $\sum_{i \in D} -b_i$. 
Three types of successful attacks

**Type 1:** Network becomes disconnected with a mismatch of supply and demand.

\[ \text{D = 8} \quad \text{S = 1} \]

\[ \text{D = 2} \quad \text{S = 9} \]

Satisfied demand = 3
Three types of successful attacks

**Type 2:** Lower bounds on generator outputs cause line overload

![Diagram showing power grid vulnerability](image)
Three types of successful attacks

**Type 3:** Uniqueness of power flows means exceeded capacities or insufficient supply.
A game:

The controller’s problem: Given a set $A$ of arcs that has been deleted by the attacker, choose a set $G$ of generators to operate, so as to feasibly meet demand (at least) $D_{\text{min}}$.

The attacker’s problem: Choose a set $A$ of arcs to delete, so as to defeat the controller, no matter how the controller chooses $G$. 
A game:

**The controller’s problem:** Given a set $A$ of arcs that has been deleted by the attacker, choose a set $G$ of generators to operate, so as to feasibly meet demand (at least) $D^{\min}$.

**The attacker’s problem:** Choose a set $A$ of arcs to delete, so as to defeat the controller, no matter how the controller chooses $G$. 
A game:

**The controller’s problem:** Given a set $\mathcal{A}$ of arcs that has been deleted by the attacker, choose a set $\mathcal{G}$ of generators to operate, so as to feasibly meet demand (at least) $D_{min}$.

**The attacker’s problem:** Choose a set $\mathcal{A}$ of arcs to delete, so as to defeat the controller, no matter how the controller chooses $\mathcal{G}$. 
The controller’s problem for a given choice of generators

Given a set $A$ of arcs that has been deleted by the attacker, AND a choice $G$ of which generators to operate, set demands and supplies so as to feasibly meet total demand (at least) $D_{\text{min}}$.

This a linear program:
The controller’s problem for a given choice of generators

Given a set $\mathcal{A}$ of arcs that has been deleted by the attacker, AND a choice $G$ of which generators to operate, set demands and supplies so as to feasibly meet total demand (at least) $D_{\text{min}}$.

This a linear program:
\( t_A(\mathcal{G}) = \min t \)

Subject to:

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i,
\]

\[
S^\text{min}_i \leq b_i \leq S^\text{max}_i \text{ for } i \in \mathcal{G}, \quad 0 \leq -b_i \leq D^\text{max}_i \text{ for } i \in D
\]

\( b_i = 0 \) otherwise.

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A
\]

\[-\sum_{i \in D} b_i + D^\text{min} t \geq 2D^\text{min} \]

\[
u_{ij} t \geq |f_{ij}| \text{ for all } (i, j) \notin A
\]

\[
f_{ij} = 0 \text{ for all } (i, j) \in A
\]

**Lemma:** \( t_A(\mathcal{G}) > 1 \) iff the attack is successful against the choice \( \mathcal{G} \).
\[ t_A(G) \triangleq \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_i^{\min} \leq b_i \leq S_i^{\max} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_i^{\max} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^{\min} t \geq 2D^{\min} \]

\[ u_{ij} t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

\[ f_{ij} = 0 \text{ for all } (i, j) \in A \]

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\[ t_A(G) \geq \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_i^{\text{min}} \leq b_i \leq S_i^{\text{max}} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_i^{\text{max}} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^{\text{min}} t \geq 2D^{\text{min}} \]

\[ u_{ij} t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

\[ f_{ij} = 0 \text{ for all } (i, j) \in A \]

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\( t_A(G) \doteq \min t \)

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_{i}^{\min} \leq b_i \leq S_{i}^{\max} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_{i}^{\max} \text{ for } i \in D \]

\( b_i = 0 \) otherwise.

\[ x_{ij}f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D_{i}^{\min} t \geq 2D_{i}^{\min} \]

\( u_{ij}t \geq |f_{ij}| \text{ for all } (i, j) \notin A \)

\( f_{ij} = 0 \text{ for all } (i, j) \in A \)

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\[ t_A(G) \overset{!}{=} \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_i^{\text{min}} \leq b_i \leq S_i^{\text{max}} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_i^{\text{max}} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij}f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^{\text{min}} t \geq 2D^{\text{min}} \]

\[ u_{ij}t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

\[ f_{ij} = 0 \text{ for all } (i, j) \in A \]

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\[ t_A(G) \triangleq \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_i^{\text{min}} \leq b_i \leq S_i^{\text{max}} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_i^{\text{max}} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^{\text{min}} t \geq 2D^{\text{min}} \]

\[ u_{ij} t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

\[ f_{ij} = 0 \text{ for all } (i, j) \in A \]

Lemma: \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\[ t_A(G) \doteq \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S_i^{\min} \leq b_i \leq S_i^{\max} \text{ for } i \in G, \quad 0 \leq -b_i \leq D_i^{\max} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^{\min} t \geq 2D^{\min} \]

\[ u_{ij} t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

\[ f_{ij} = 0 \text{ for all } (i, j) \in A \]

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
\( t_A(G) \equiv \min t \)

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \]

\[ S^\text{min}_i \leq b_i \leq S^\text{max}_i \text{ for } i \in G, \quad 0 \leq -b_i \leq D^\text{max}_i \text{ for } i \in D \]

\( b_i = 0 \) otherwise.

\[ x_{ij}f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \]

\[ -\sum_{i \in D} b_i + D^\text{min} t \geq 2D^\text{min} \]

\[ u_{ij}t \geq |f_{ij}| \text{ for all } (i, j) \notin A \]

for all \( (i, j) \in A, \quad t \geq 1 + |f_{ij}|/u_{ij} \)

**Lemma:** \( t_A(G) > 1 \) iff the attack is successful against the choice \( G \).
Attack problem

\[
\min \sum_{ij} z_{ij}
\]

Subject to:

\[z_{ij} = 0 \text{ or } 1, \text{ for all arcs } (i, j), \quad \text{(choose which arcs to delete)}\]

\[t_{\text{suppt}(z)}(\mathcal{G}) > 1, \quad \text{for every subset } \mathcal{G} \text{ of generators.}\]

[ \text{suppt}(v) = \text{support of } v ]

→ Use dual to represent \( t_{\text{suppt}(z)}(\mathcal{G}) \)
Attack problem

\[
\min \sum_{ij} z_{ij}
\]

Subject to:

\[
z_{ij} = 0 \text{ or } 1, \text{ for all arcs } (i, j), \quad \text{(choose which arcs to delete)}
\]

\[
t_{\text{suppt}(z)}(\mathcal{G}) > 1, \quad \text{for every subset } \mathcal{G} \text{ of generators.}
\]

[ \text{suppt}(v) = \text{support of } v ]

\[\rightarrow \text{ Use dual to represent } t_{\text{suppt}(z)}(\mathcal{G})\]
Building the dual

\[ t_{A}(G) = \min t \]

Subject to:

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \quad (\alpha_i) \]

\[ S_{i}^{\text{min}} \leq b_i \leq S_{i}^{\text{max}} \text{ for } i \in G, \]

\[ 0 \leq -b_i \leq D_{i}^{\text{max}} \text{ for } i \in D \]

\[ b_i = 0 \text{ otherwise.} \]

\[ x_{ij}f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j) \notin A \quad (\beta_{ij}) \]

\[ - (\sum_{i \in D} b_i) / D^{\text{min}} + t \geq 2 \]

\[ u_{ij}t \geq |f_{ij}| \text{ for all } (i, j) \notin A \quad (p_{ij}, q_{ij}) \]

\[ u_{ij}t \geq u_{ij} + |f_{ij}| \text{ for all } (i, j) \in A \quad (r_{ij}^{+}, r_{ij}^{-}) \]
Building the dual

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \quad (\alpha_i)
\]

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \quad \text{for all } (i, j) \notin A \quad (\beta_{ij})
\]

\[
u_{ij} t \geq |f_{ij}| \quad \text{for all } (i, j) \notin A \quad (p_{ij}, q_{ij})
\]

\[
u_{ij} t \geq u_{ij} + |f_{ij}| \quad \text{for all } (i, j) \in A \quad (r_{ij}^+, r_{ij}^-)
\]

\[
\sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i
\]

\[
\alpha_i - \alpha_j + x_{ij} \beta_{ij} = p_{ij} - q_{ij} + r_{ij}^+ - r_{ij}^- \quad \forall (i, j)
\]
Building the dual

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \quad \text{for all nodes } i, \quad (\alpha_i) \]
\[ x_{ij} f_{ij} - \theta_i + \theta_j = 0 \quad \text{for all } (i, j) \notin \mathcal{A} \quad (\beta_{ij}) \]
\[ u_{ij} t \geq |f_{ij}| \quad \text{for all } (i, j) \notin \mathcal{A} \quad (p_{ij}, q_{ij}) \]
\[ u_{ij} t \geq u_{ij} + |f_{ij}| \quad \text{for all } (i, j) \in \mathcal{A} \quad (r_{ij}^+, r_{ij}^-) \]
\[ \sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i \]
\[ \alpha_i - \alpha_j + x_{ij}\beta_{ij} = p_{ij} - q_{ij} + r_{ij}^+ - r_{ij}^- \quad \forall (i, j) \]
Building the dual

\[ \sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0, \text{ for all nodes } i, \quad (\alpha_i) \]
\[ x_{ij}f_{ij} - \theta_i + \theta_j = 0 \quad \text{for all } (i, j) \notin \mathcal{A} \quad (\beta_{ij}) \]
\[ u_{ij}t \geq |f_{ij}| \quad \text{for all } (i, j) \notin \mathcal{A} \quad (p_{ij}, q_{ij}) \]
\[ u_{ij}t \geq u_{ij} + |f_{ij}| \quad \text{for all } (i, j) \in \mathcal{A} \quad (r_{ij}^+, r_{ij}^-) \]
\[ \sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i \]
\[ \alpha_i - \alpha_j + x_{ij}\beta_{ij} = p_{ij} - q_{ij} + r_{ij}^+ - r_{ij}^- \quad \forall (i, j) \]
Again:

\[ \sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i \]

\[ \alpha_i - \alpha_j + x_{ij} \beta_{ij} = p_{ij} - q_{ij} + r_{ij}^+ - r_{ij}^- \quad \forall (i, j) \]

0-1 -ify: form mip-dual

\[ p_{ij} + q_{ij} \leq M_{ij}(1 - z_{ij}) \]

\[ r_{ij}^+ + r_{ij}^- \leq M'_{ij} z_{ij} \]

→ “big M” formulation: what’s the problem
Again:

\[ \sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i \]

\[ \alpha_i - \alpha_j + x_{ij} \beta_{ij} = p_{ij} - q_{ij} + r^+_{ij} - r^-_{ij} \quad \forall (i, j) \]

0-1 -ify: form mip-dual

\[ p_{ij} + q_{ij} \leq M_{ij}(1 - z_{ij}) \]

\[ r^+_{ij} + r^-_{ij} \leq M'_{ij} z_{ij} \]

→ “big M” formulation: what’s the problem
Again:

\[ \sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \quad \forall i \]

\[ \alpha_i - \alpha_j + x_{ij} \beta_{ij} = p_{ij} - q_{ij} + r_{ij}^+ - r_{ij}^- \quad \forall (i, j) \]

0-1 -ify: form mip-dual

\[ p_{ij} + q_{ij} \leq M_{ij} (1 - z_{ij}) \]

\[ r_{ij}^+ + r_{ij}^- \leq M'_{ij} z_{ij} \]

→ “big M” formulation: what’s the problem
I hate math

\[ M_{ij} = \sqrt{x_{ij}} \max_{(k,l)} \left( \sqrt{x_{kl}} u_{kl} \right)^{-1} \]
A formulation for the attack problem

\[
\min \sum_{ij} z_{ij}
\]

Subject to:

\[z_{ij} = 0 \text{ or } 1, \text{ for all arcs } (i, j),\] (choose which arcs to delete)

\[t_{\text{supp}(z)}(G) > 1, \text{ for every subset } G \text{ of generators.}\]
A formulation for the attack problem

\[ \min \sum_{ij} z_{ij} \]

Subject to:
\[ z_{ij} = 0 \text{ or } 1, \text{ for all arcs } (i, j), \quad (\text{choose which arcs to delete}) \]

value of dual mip \((G) > 1\), for every subset \(G\) of generators.

\[ \rightarrow \text{ very large} \]
A formulation for the attack problem

$$\min \sum_{ij} z_{ij}$$

Subject to:

$$z_{ij} = 0 \text{ or } 1, \text{ for all arcs } (i, j), \quad (\text{choose which arcs to delete})$$

**value of dual mip \((G)\) > 1, \text{ for every subset } G \text{ of generators.}$$

→ very large
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  - Made up of valid inequalities (for the attacker)
  - Initially empty

Iterate:

1. Solve master MIP, obtain \(0 - 1\) vector \(z^*\).
2. Solve controller problem to test whether \(\text{supp}(z^*)\) is a successful attack:
   - If successful, then \(z^*\) is an optimal solution
   - If not, then for some set of generators \(G\), \(t_{\text{supp}(z^*)}(G) \leq 1\).
3. Add to master MIP a system that cuts off \(z^*\) and go to 1.
**Algorithm outline**

→ Maintain a “master (attacker) MIP”:
  - Made up of valid inequalities (for the attacker)
  - Initially empty

**Iterate:**

1. Solve master MIP, obtain \(0 - 1\) vector \(z^*\).

2. Solve controller problem to test whether \(\text{supp}(z^*)\) is a successful attack:
   - If successful, then \(z^*\) is an optimal solution
   - If not, then for some set of generators \(G\), \(\tau_{\text{supp}(z^*)}(G) \leq 1\).

3. Add to master MIP a **system** that cuts off \(z^*\) and go to 1.
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  • Made up of valid inequalities (for the attacker)
  • Initially empty

Iterate:

1. Solve master MIP, obtain $0 - 1$ vector $z^*$. 

2. Solve controller problem to test whether $\text{supp}(z^*)$ is a successful attack:
   • If successful, then $z^*$ is an optimal solution
   • If not, then for some set of generators $G$, $t_{\text{supp}(z^*)}(G) \leq 1$.

3. Add to master MIP a system that cuts off $z^*$ and go to 1.
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  • Made up of valid inequalities (for the attacker)
  • Initially empty

Iterate:

1. Solve master MIP, obtain $0 - 1$ vector $z^*$. 
2. Solve controller problem to test whether $\text{supp}(z^*)$ is a successful attack:
   • If successful, then $z^*$ is an optimal solution
   • If not, then for some set of generators $G$, $t_{\text{supp}(z^*)}(G) \leq 1$.
3. Add to master MIP a system that cuts off $z^*$ and go to 1.
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  - Made up of valid inequalities (for the attacker)
  - Initially empty

Iterate:

1. Solve master MIP, obtain $0 - 1$ vector $z^*$.

2. Solve controller problem to test whether $\text{supp}(z^*)$ is a successful attack:
   - If successful, then $z^*$ is an optimal solution
   - If not, then for some set of generators $G$, $t_{\text{supp}(z^*)}(G) \leq 1$.

3. Add to master MIP a system that cuts off $z^*$ and go to 1.
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  • Made up of valid inequalities (for the attacker)
  • Initially empty

Iterate:

1. Solve master MIP, obtain $0 - 1$ vector $z^*$.

2. Solve controller problem to test whether $\text{supp}(z^*)$ is a successful attack:
   • If successful, then $z^*$ is an optimal solution
   • If not, then for some set of generators $G$, $t_{\text{supp}(z^*)}(G) \leq 1$.

3. Add to master MIP a system that cuts off $z^*$ and go to 1.
Algorithm outline

→ Maintain a “master (attacker) MIP”:
  - Made up of valid inequalities (for the attacker)
  - Initially empty

Iterate:

1. Solve master MIP, obtain $0 - 1$ vector $z^*$.  
2. Solve controller problem to test whether $\text{supp}(z^*)$ is a successful attack:
   - If successful, then $z^*$ is an optimal solution
   - If not, then for some set of generators $\mathcal{G}$, $t_{\text{supp}(z^*)}(\mathcal{G}) \leq 1$.
3. Add to master MIP a system that cuts off $z^*$ and go to 1.
Cutting planes = Benders’ cuts

For a given $0 - 1$ vector $\hat{z}$, and a set of generators $G$,

$$t_{\text{supp}}(\hat{z})(G) = \max \mu^T y$$

s.t.

$$Ay \leq b\hat{z}$$

$$y \in P$$

for some vectors $\mu$, $b$, matrix $A$ and polyhedron $P$, (all dependent on $G$, but not $\hat{z}$).

→ If $t_{\text{supp}}(\hat{z})(G) \leq 1$, use LP duality to separate $\hat{z}$, getting a cut $\alpha^T z \geq \beta$ violated by $\hat{z}$. 
Cutting planes = Benders’ cuts

For a given $0 - 1$ vector $\hat{z}$, and a set of generators $G$,

$$t_{supp}(\hat{z})(G) = \max \mu^T y$$

s.t.

$$Ay \leq b\hat{z}$$

$$y \in P$$

for some vectors $\mu$, $b$, matrix $A$ and polyhedron $P$, (all dependent on $G$, but not $\hat{z}$).

→ If $t_{supp}(\hat{z})(G) \leq 1$, use LP duality to separate $\hat{z}$,

getting a cut $\alpha^t z \geq \beta$ violated by $\hat{z}$. 
Given an unsuccessful attack $z^*$, “Pad” it: choose arcs $a_1, a_2, \ldots, a_k$ such that\

$\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}, a_k\}$ is successful, but $\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$ is not

Then separate $\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$

→ other definitions of “padding”
Given an unsuccessful attack $z^*$,

“Pad” it: choose arcs $a_1, a_2, \ldots, a_k$ such that

$$\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}, a_k\}$$
is successful, but

$$\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$$
is not

Then separate $\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$

→ other definitions of “padding”
Plus:

Given an unsuccessful attack $z^*$,

“Pad” it: choose arcs $a_1, a_2, \ldots, a_k$ such that

$$\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}, a_k\}$$

is successful, but

$$\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$$

is not

Then separate $\text{supp}(z^*) \cup \{a_1, a_2, \ldots, a_{k-1}\}$

$\rightarrow$ other definitions of “padding”
Plus, combinatorial relaxations

**Strengthen controller or weaken attacker** $\rightarrow$ obtain valid attacks (e.g. upper bounds)

Example: fractional controller

**Strengthen attacker or weaken controller** $\rightarrow$ obtain valid lower bounds.

Example: when an arc is attacked, flow goes to zero, but Ohm’s law still applies.
Plus, combinatorial relaxations

Strengthen controller or weaken attacker → obtain valid attacks (e.g. upper bounds)

Example: fractional controller

Strengthen attacker or weaken controller → obtain valid lower bounds.

Example: when an arc is attacked, flow goes to zero, but Ohm’s law still applies
98 nodes, 204 arcs
Entries show: (iteration count), time,
Attack status (\( F \) = cardinality too small, \( S \) = attack success)

12 generators

<table>
<thead>
<tr>
<th>Min. throughput</th>
<th>Attack cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>0.92</td>
<td>(2), 318, ( F )</td>
</tr>
<tr>
<td>0.90</td>
<td>(2), 161, ( F )</td>
</tr>
<tr>
<td>0.88</td>
<td>(2), 165, ( F )</td>
</tr>
<tr>
<td>0.84</td>
<td>(2), 150, ( F )</td>
</tr>
<tr>
<td>0.75</td>
<td>(2), 130, ( F )</td>
</tr>
</tbody>
</table>
98 nodes, 204 arcs
Entries show: (iteration count), time,
Attack status (F = cardinality too small, S = attack success)

<table>
<thead>
<tr>
<th>Min. throughput</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.94</td>
<td>(2), 223, F</td>
<td>(11), 654, S</td>
<td></td>
</tr>
<tr>
<td>0.92</td>
<td>(2), 201, F</td>
<td>(11), 10895, F</td>
<td>(18), 11223, S</td>
</tr>
<tr>
<td>0.90</td>
<td>(2), 193, F</td>
<td>(11), 6598, F</td>
<td>(16), 206350, S</td>
</tr>
<tr>
<td>0.88</td>
<td>(2), 256, F</td>
<td>(9), 15445, F</td>
<td>(18), 984743, F</td>
</tr>
<tr>
<td>0.84</td>
<td>(2), 133, F</td>
<td>(9), 5565, F</td>
<td>(15), 232525, F</td>
</tr>
<tr>
<td>0.75</td>
<td>(2), 213, F</td>
<td>(9), 7550, F</td>
<td>(11), 100583, F</td>
</tr>
</tbody>
</table>
A different model


→ The expectation is that such weaknesses exist, and we need a method to reveal them

→ Allow the adversary to selectively place stress on the grid in order to cause failure

→ Allow the adversary the ability to exceed the laws of physics, in a limited way, so as to cause failure
A different model


→ The expectation is that such weaknesses exist, and we need a method to reveal them

→ Allow the adversary to selectively place stress on the grid in order to cause failure

→ Allow the adversary the ability to exceed the laws of physics, in a limited way, so as to cause failure
A different model


→ The expectation is that such weaknesses exist, and we need a method to reveal them

→ Allow the adversary to selectively place stress on the grid in order to cause failure

→ Allow the adversary the ability to exceed the laws of physics, in a limited way, so as to cause failure
A different model


→ The expectation is that such weaknesses exist, and we need a method to reveal them

→ Allow the adversary to selectively place stress on the grid in order to cause failure

→ Allow the adversary the ability to exceed the laws of physics, in a limited way, so as to cause failure
A **power flow** is a solution $f, \theta$ to:

\[
\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i, \text{ for all } i, \text{ where }
\]

- $b_i > 0$ when $i$ is a generator,
- $b_i < 0$ when $i$ is a demand,

and $b_i = 0$, otherwise.

\[
x_{ij} f_{ij} - \theta_i + \theta_j = 0 \text{ for all } (i, j).
\]

**Lemma** Given a choice for $b$ with $\sum_i b_i = 0$, the system has a **unique** solution.

→ For fixed $b$, $f = f(x)$
A **power flow** is a solution \( f, \theta \) to:

1. \( \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i \), for all \( i \), where
   - \( b_i > 0 \) when \( i \) is a generator,
   - \( b_i < 0 \) when \( i \) is a demand,
   - and \( b_i = 0 \), otherwise.

2. \( x_{ij} f_{ij} - \theta_i + \theta_j = 0 \) for all \((i, j)\).

**Lemma** Given a choice for \( b \) with \( \sum_i b_i = 0 \), the system has a **unique** solution.

→ For fixed \( b, f = f(x) \)
Power flows (again)

A **power flow** is a solution \( f, \theta \) to:

- \( \sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i \), for all \( i \), where
  - \( b_i > 0 \) when \( i \) is a generator,
  - \( b_i < 0 \) when \( i \) is a demand,
  - and \( b_i = 0 \), otherwise.

- \( x_{ij} f_{ij} - \theta_i + \theta_j = 0 \) for all \( (i, j) \).

**Lemma** Given a choice for \( b \) with \( \sum_i b_i = 0 \), the system has a **unique** solution.

\[ \rightarrow \text{For fixed } b, \ f = f(x) \]
Power flows (again)

A power flow is a solution $f, \theta$ to:

1. $\sum_{ij} f_{ij} - \sum_{ij} f_{ji} = b_i$, for all $i$, where
   - $b_i > 0$ when $i$ is a generator,
   - $b_i < 0$ when $i$ is a demand,
   - and $b_i = 0$, otherwise.

2. $x_{ij} f_{ij} - \theta_i + \theta_j = 0$ for all $(i, j)$.

Lemma Given a choice for $b$ with $\sum_i b_i = 0$, the system has a unique solution.

$\rightarrow$ For fixed $b$, $f = f(x)$
Model

(I) The attacker sets the resistance $x_{ij}$ of any arc $(i, j)$.

(II) The attacker is constrained: we must have $x \in F$ for a certain known set $F$.

(III) The output of each generator $i$ is fixed at a given value $P_i$, and similarly each demand value $D_i$ is also fixed at a given value.

(IV) The objective of the attacker is to maximize the overload of any arc, that is to say, the attacker wants to solve

$$\max_{x \in F} \max_{ij} \left\{ \frac{|f_{ij}(x)|}{U_{ij}} \right\},$$

Example for $F$:

$$\sum_{ij} x_{ij} \leq B, \quad x_{ij}^L \leq x_{ij} \leq x_{ij}^U \quad \forall (i, j),$$
Model

(I) The attacker sets the resistance \( x_{ij} \) of any arc \((i, j)\).

(II) The attacker is constrained: we must have \( x \in F \) for a certain known set \( F \).

(III) The output of each generator \( i \) is fixed at a given value \( P_i \), and similarly each demand value \( D_i \) is also fixed at a given value.

(IV) The objective of the attacker is to maximize the overload of any arc, that is to say, the attacker wants to solve

\[
\max_{x \in F} \max_{ij} \left\{ \frac{|f_{ij}(x)|}{u_{ij}} \right\},
\]

Example for \( F \):

\[
\sum_{ij} x_{ij} \leq B, \quad x_{ij}^L \leq x_{ij} \leq x_{ij}^U \quad \forall (i, j),
\]
Lemma (excerpt)

Let $S$ be a set of arcs whose removal does not disconnect $G$.

Suppose we set $x_{st} = L$ for each arc $(s, t) \in S$.

Let $f(x)$ denote the resulting power flow, and let $\bar{f}$ the solution to the power flow problem on $G - S$.

Then

(a) $\lim_{L \to +\infty} f_{st}(x) = 0$, for all $(s, t) \in S$, 

(b) For any $(u, v) \not\in S$, $\lim_{L \to +\infty} f_{uv}(x) = \bar{f}_{uv}$. 
How to solve the problem

$$\max_{x \in F} \max_{ij} \left\{ \frac{|f_{ij}(x)|}{u_{ij}} \right\}$$

Smooth version:

$$\max_{x,p} \sum_{ij} \frac{f_{ij}(x)}{u_{ij}}(p_{ij} - q_{ij})$$

s.t. $$\sum_{ij} (p_{ij} + q_{ij}) = 1,$$

$$x \in F, \quad p, q \geq 0.$$
How to solve the problem

\[
\max_{x \in F} \max_{ij} \left\{ \frac{|f_{ij}(x)|}{u_{ij}} \right\}
\]

Smooth version:

\[
\max_{x, p} \sum_{ij} \frac{f_{ij}(x)}{u_{ij}} (p_{ij} - q_{ij})
\]

s.t. \[
\sum_{ij} (p_{ij} + q_{ij}) = 1, \quad x \in F, \quad p, q \geq 0.
\]

(but not concave)
How to solve the problem

\[
\max_{x \in F} \max_{ij} \left\{ \frac{|f_{ij}(x)|}{u_{ij}} \right\}
\]

Smooth version:

\[
\max_{x, p} \sum_{ij} \frac{f_{ij}(x)}{u_{ij}} (p_{ij} - q_{ij})
\]

s.t.

\[
\sum_{ij} (p_{ij} + q_{ij}) = 1,
\]

\[
x \in F, \quad p, q \geq 0.
\]

(but not concave)
Methodology

→ A recent research trend: adapt methodologies from smooth, convex optimization to smooth, non-convex optimization.

→ Several industrial-strength codes.

Our objective:

\[ F(x, p) = \sum_{ij} \frac{f_{ij}(x)}{u_{ij}} (p_{ij} - q_{ij}) \]

Lemma: There exist efficient, sparse linear algebra algorithms for computing the gradient \( \nabla_{x,p} F(x, p) \) and Hessian \( \frac{\partial^2 F(x, p)}{\partial^2 x,p} \)
A recent research trend: adapt methodologies from smooth, convex optimization to smooth, non-convex optimization.

Several industrial-strength codes.

Our objective:

\[
F(x, p) = \sum_{ij} \frac{f_{ij}(x)}{u_{ij}}(p_{ij} - q_{ij})
\]

Lemma: There exist efficient, sparse linear algebra algorithms for computing the gradient \( \nabla_{x,p} F(x, p) \) and Hessian \( \frac{\partial^2 F(x,p)}{\partial^2 x,p} \).
Methodology

→ A recent research trend: adapt methodologies from smooth, convex optimization to smooth, non-convex optimization.

→ Several industrial-strength codes.

Our objective:

\[ F(x, p) = \sum_{ij} \frac{f_{ij}(x)}{u_{ij}} (p_{ij} - q_{ij}) \]

Lemma: There exist efficient, sparse linear algebra algorithms for computing the gradient \( \nabla_{x,p} F(x, p) \) and Hessian \( \frac{\partial^2 F(x,p)}{\partial^2 x,p} \)
Some details

Implementation using LOQO (currently testing SNOPT)

Adversarial model:

$$\sum_{ij} x_{ij} \leq B, \quad x_{ij}^L \leq x_{ij} \leq x_{ij}^U \quad \forall (i, j),$$

where (this talk):

$$x_{ij}^L = 1, \quad x_{ij}^U = 10, \quad \forall (i, j),$$

and

$$\sum_{(i,j)} x_{ij} = \sum_{(i,j)} x_{ij}^L + \Delta B,$$

where

$$\Delta B \leq 40$$
Sample computational experience

**Table: 600 nodes, 990 arcs**

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>27</th>
<th>36</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>obj</strong></td>
<td>0.571562</td>
<td>1.076251</td>
<td>1.156187</td>
<td>1.088491</td>
<td>1.161887</td>
</tr>
<tr>
<td><strong>sec</strong></td>
<td>11848</td>
<td>7500</td>
<td>4502</td>
<td>11251</td>
<td>7800</td>
</tr>
<tr>
<td><strong>Its</strong></td>
<td>Limit</td>
<td>210</td>
<td>114</td>
<td>Limit</td>
<td>208</td>
</tr>
<tr>
<td><strong>stat</strong></td>
<td>PDfeas Iter: 300</td>
<td>$\epsilon$-L-opt.</td>
<td>$\epsilon$-L-opt.</td>
<td>PDfeas Iter: 300</td>
<td>$\epsilon$-L-opt.</td>
</tr>
</tbody>
</table>
### Table: Attack pattern

<table>
<thead>
<tr>
<th>Range</th>
<th>Count</th>
<th>Range</th>
<th>Count</th>
<th>Range</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 1]</td>
<td>8</td>
<td>[1, 1]</td>
<td>1</td>
<td>[1, 1]</td>
<td>14</td>
</tr>
<tr>
<td>(1, 2]</td>
<td>72</td>
<td>(1, 2]</td>
<td>405</td>
<td>(1, 2]</td>
<td>970</td>
</tr>
<tr>
<td>(2, 3]</td>
<td>4</td>
<td>(2, 9]</td>
<td>0</td>
<td>(2, 5]</td>
<td>3</td>
</tr>
<tr>
<td>(5, 6]</td>
<td>1</td>
<td>(9, 10]</td>
<td>3</td>
<td>(5, 6]</td>
<td>0</td>
</tr>
<tr>
<td>(6, 7]</td>
<td>1</td>
<td>(6, 7]</td>
<td>1</td>
<td>(6, 7]</td>
<td>1</td>
</tr>
<tr>
<td>(7, 8]</td>
<td>4</td>
<td>(7, 9]</td>
<td>0</td>
<td>(7, 9]</td>
<td>0</td>
</tr>
<tr>
<td>(8, 20]</td>
<td>0</td>
<td>(9, 10]</td>
<td>2</td>
<td>(9, 10]</td>
<td>2</td>
</tr>
</tbody>
</table>

$x^u = 20 \quad \Delta B = 57$

$x^u = 10 \quad \Delta B = 27$

$x^u = 10 \quad \Delta B = 36$
### Impact

<table>
<thead>
<tr>
<th>Ovl</th>
<th>Top 6 Arcs</th>
<th>R-3</th>
<th>R-3- 10%</th>
<th>C-all- 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.15</td>
<td>29(7.79), 27(7.20), 41(7.03), 67(7.02), 54(6.72), 79(5.71)</td>
<td>1.718</td>
<td>1.335</td>
<td>1.671</td>
</tr>
<tr>
<td>1.79</td>
<td>29(8.28), 27(7.72), 41(7.32), 67(7.19), 54(6.92), 79(5.78)</td>
<td>1.431</td>
<td>1.112</td>
<td>1.386</td>
</tr>
<tr>
<td>1.56</td>
<td>29(8.31), 27(7.74), 41(7.53), 67(7.48), 54(7.18), 79(6.15)</td>
<td>1.227</td>
<td>0.953</td>
<td>1.213</td>
</tr>
<tr>
<td>1.36</td>
<td>29(8.18), 27(7.58), 41(7.53), 67(7.58), 54(7.22), 79(6.25)</td>
<td>1.073</td>
<td>0.834</td>
<td>1.055</td>
</tr>
<tr>
<td>1.20</td>
<td>29(8.43), 27(7.90), 41(7.53), 67(7.48), 54(7.18), 79(6.12)</td>
<td>0.954</td>
<td>0.741</td>
<td>0.936</td>
</tr>
<tr>
<td>1.08</td>
<td>29(7.87), 27(7.29), 41(7.04), 67(7.01), 54(6.70), 79(5.63)</td>
<td>0.859</td>
<td>0.668</td>
<td>0.839</td>
</tr>
</tbody>
</table>