Power grid vulnerability, new models, algorithms and computing

Daniel Bienstock
Abhinav Verma

Columbia University, New York

July, 2009

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^{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

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" $0 \leq S_i^L \leq S_i^U$,
- A set D of **demand** nodes (the "loads"); for each load i a "maximum demand" $0 \leq D_i^{max}$.
- 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" $0 \leq S_i^L \leq S_i^U$,
- A set D of **demand** nodes (the "loads"); for each load i a "maximum demand" $0 \leq D_i^{max}$.
- 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" $0 \leq S_i^L \leq S_i^U$,
- A set D of **demand** nodes (the "loads"); for each load i a "maximum demand" $0 \leq D_i^{max}$.
- For each arc (i, j) values x_{ij} and u_{ij}.

Feasible power flows

A **power flow** is a solution f, θ 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 $\sum_{i \in D} -b_i$.

Feasible power flows

A **power flow** is a solution f, θ 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 $\sum_{i \in D} -b_i$.

Feasible power flows

A **power flow** is a solution f, θ 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 $\sum_{i \in D} -b_i$.

Feasible power flows

A **power flow** is a solution f, θ 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 $\sum_{i \in D} -b_i$.

Feasible power flows

A **power flow** is a solution f, θ 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)

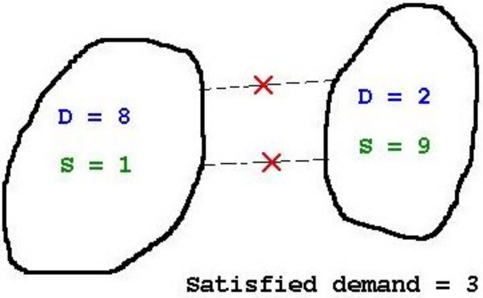
Lemma Given a choice for b with $\sum_i b_i = 0$, the system has a **unique** solution.

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

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

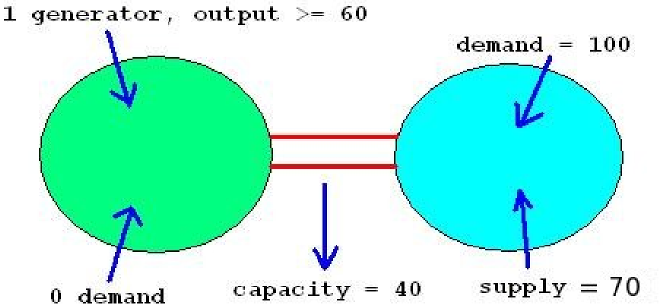
Three types of successful attacks

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



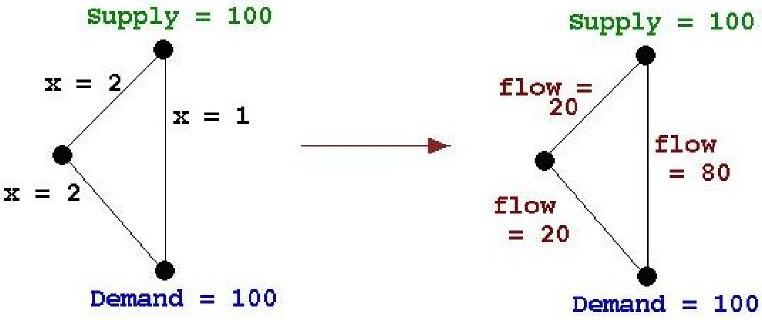
Three types of successful attacks

Type 2: Lower bounds on generator ouptuts cause line overload



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 $\mathcal{A}$ 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 $\mathcal{G}$.

A game:

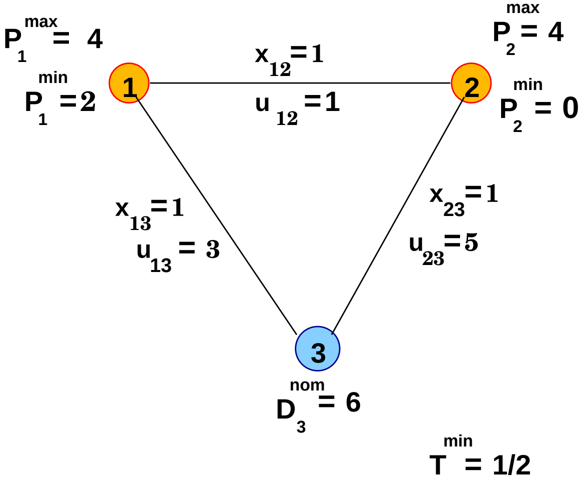
The *controller's problem:* Given a set $\mathcal{A}$ of arcs that has been deleted by the attacker, choose a set $\mathcal{G}$ of generators to operate, so as to feasibly meet demand (at least) D^{min}.

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 $\mathcal{G}$.

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 $\mathcal{G}$.


$P_1^{max} = 4$
$P_1^{min} = 2$
$x_{12} = 1$
$u_{12} = 1$
$P_2^{max} = 4$
$P_2^{min} = 0$
1
2
3
$x_{13} = 1$
$u_{13} = 3$
$x_{23} = 1$
$u_{23} = 5$
$D_3^{nom} = 6$
$T^{min} = 1/2$

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 $\mathcal{G}$ of which generators to operate, set demands and supplies so as to feasibly meet total demand (at least) D^{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 $\mathcal{G}$ of which generators to operate, set demands and supplies so as to feasibly meet total demand (at least) D^{min}.

This a linear program:

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

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

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \leq -b_i \leq D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

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

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

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

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

$\sum_{ij} f_{ij} \; - \; \sum_{ij} f_{ji} \; - \; b_i \; = \; 0$, for all nodes i,

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \leq -b_i \leq D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

$x_{ij} f_{ij} \; - \; \theta_i \; + \; \theta_j \; = \; 0$ for all $(i,j) \notin \mathcal{A}$

$-\sum_{i \in D} b_i \; + \; D^{\min} \; t \geq 2D^{\min}$

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

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

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

$S_i^{min} \le b_i \le S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \le -b_i \le D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

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

$-\sum_{i \in D} b_i + D^{\min} t \ge 2D^{\min}$

$u_{ij} t \ge |f_{ij}|$ for all $(i, j) \notin \mathcal{A}$

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

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

$S_i^{min} \le b_i \le S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \le -b_i \le D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

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

$-\sum_{i \in D} b_i + D^{\min} t \ge 2D^{\min}$

$u_{ij} t \ge |f_{ij}|$ for all $(i,j) \notin \mathcal{A}$

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

$\sum_{ij} f_{ij} \; - \; \sum_{ij} f_{ji} \; - \; b_i \; = \; 0$, for all nodes i,

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \leq -b_i \leq D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

$x_{ij} f_{ij} \; - \; \theta_i \; + \; \theta_j \; = \; 0$ for all $(i, j) \notin \mathcal{A}$

$-\sum_{i \in D} b_i \; + \; D^{\min} \; t \geq 2D^{\min}$

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

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

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

$S_i^{min} \le b_i \le S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \le -b_i \le D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

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

$-\sum_{i \in D} b_i + D^{\min} t \ge 2D^{\min}$

$u_{ij} t \ge |f_{ij}|$ for all $(i, j) \notin \mathcal{A}$

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

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

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \leq -b_i \leq D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

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

$-\sum_{i \in D} b_i + D^{\mathrm{min}} t \geq 2D^{\mathrm{min}}$

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

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

$\sum_{ij} f_{ij} \; - \; \sum_{ij} f_{ji} \; - \; b_i \; = \; 0$, for all nodes i,

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$, $\quad 0 \leq -b_i \leq D_i^{max}$ for $i \in D$

$b_i = 0$ otherwise.

$x_{ij} f_{ij} \; - \; \theta_i \; + \; \theta_j \; = \; 0$ for all $(i,j) \notin \mathcal{A}$

$-\sum_{i \in D} b_i \; + \; D^{\mathrm{min}} \; t \geq 2D^{\mathrm{min}}$

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

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

Lemma: $t_{\mathcal{A}}(\mathcal{G}) > 1$ iff the attack is successful against the choice $\mathcal{G}$.

Attack problem

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

Subject to:

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

$t_{suppt(z)}(\mathcal{G}) > 1$, for every subset $\mathcal{G}$ of generators.

[$suppt(v) =$ support of v]

$\longrightarrow$ Use LP dual to represent $t_{suppt(z)}(\mathcal{G})$

Attack problem

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

Subject to:

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

$t_{suppt(z)}(\mathcal{G}) > 1$, for every subset $\mathcal{G}$ of generators.

[$suppt(v)$ = support of v]

$\longrightarrow$ Use LP dual to represent $t_{suppt(z)}(\mathcal{G})$

Building the dual

$t_{\mathcal{A}}(\mathcal{G}) \doteq \min t$

Subject to:

$\sum_{ij} f_{ij} - \sum_{ij} f_{ji} - b_i = 0$, for all nodes i, (α_i)

$S_i^{min} \leq b_i \leq S_i^{max}$ for $i \in \mathcal{G}$,

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

$b_i = 0$ otherwise.

$x_{ij} f_{ij} - \theta_i + \theta_j = 0$ for all $(i, j) \notin \mathcal{A}$ (β_{ij})

$-\left(\sum_{i \in D} b_i\right) / D^{min} + t \geq 2$

$u_{ij} t \geq |f_{ij}|$ for all $(i, j) \notin \mathcal{A}$ (p_{ij}, q_{ij})

$u_{ij} t \geq u_{ij} + |f_{ij}|$ for all $(i, j) \in \mathcal{A}$ (r_{ij}^{+}, r_{ij}^{-})

Building the dual

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

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

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

$u_{ij} t \geq u_{ij} + |f_{ij}|$ 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

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

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

$$u_{ij} t \geq |f_{ij}| \text{ for all } (i,j) \notin \mathcal{A} \qquad (p_{ij}, q_{ij})$$

$$u_{ij} t \geq u_{ij} + |f_{ij}| \text{ for all } (i,j) \in \mathcal{A} \qquad (r^+_{ij}, r^-_{ij})$$

$$\sum_{ij} \beta_{ij} - \sum_{ji} \beta_{ji} = 0 \qquad \forall i$$

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

Building the dual

$$\sum_{ij} f_{ij} \; - \; \sum_{ij} f_{ji} \; - \; b_i \; = \; 0, \text{ for all nodes } \; i, \qquad (\alpha_i)$$

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

$$u_{ij} t \geq |f_{ij}| \text{ for all } \; (i,j) \notin \mathcal{A} \qquad (p_{ij}, q_{ij})$$

$$u_{ij} t \geq u_{ij} + |f_{ij}| \text{ for all } \; (i,j) \in \mathcal{A} \qquad (r^+_{ij}, r^-_{ij})$$

$$\sum_{ij} \beta_{ij} \; - \; \sum_{ji} \beta_{ji} \; = \; 0 \qquad \forall i$$

$$\alpha_i \; - \; \alpha_j \; + \; x_{ij}\beta_{ij} = \; p_{ij} - q_{ij} + r^+_{ij} - r^-_{ij} \qquad \forall (i,j)$$

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

$$p_{ij} + q_{ij} \le M_{ij}(1 - z_{ij})$$

$$r^+_{ij} + r^-_{ij} \le M'_{ij} z_{ij}$$

$\rightarrow$ "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

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

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

$\rightarrow$ "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

$$p_{ij} + q_{ij} \le M_{ij}(1 - z_{ij})$$

$$r^+_{ij} + r^-_{ij} \le M'_{ij} z_{ij}$$

$\longrightarrow$ "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:

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

$t_{suppt(z)}(\mathcal{G}) > 1$, for every subset $\mathcal{G}$ of generators.

A formulation for the attack problem

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

Subject to:

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

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

$\longrightarrow$ very large

A formulation for the attack problem

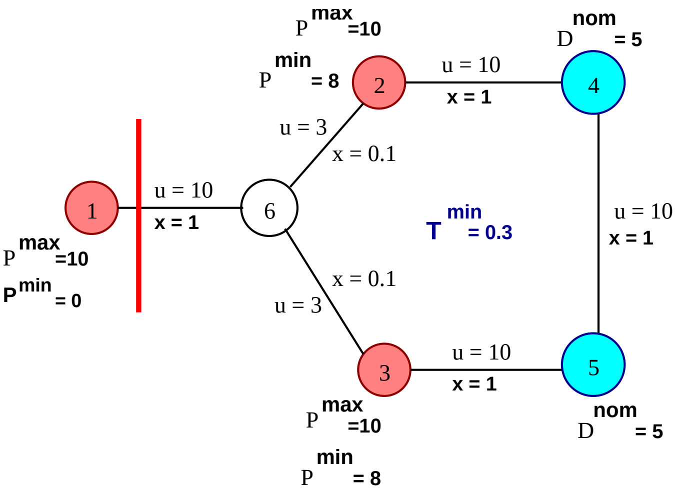
$\min \sum_{ij} z_{ij}$

Subject to:

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

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

$\longrightarrow$ very large


P max =10
P min = 8
u = 10
x = 1
nom
D = 5
1
2
4
6
u = 3
x = 0.1
u = 10
x = 1
T min = 0.3
u = 10
x = 1
P max =10
P min = 0
x = 0.1
u = 3
3
5
u = 10
x = 1
P max =10
P min = 8
nom
D = 5

Algorithm outline

$\longrightarrow$ 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 $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \leq 1$.

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

Algorithm outline

$\longrightarrow$ 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 $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \le 1$.

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

Algorithm outline

$\longrightarrow$ Maintain a “master (attacker) MIP”:

- Made up of valid inequalities (for the attacker)
- Initially empty

Iterate:

1. Solve master MIP, obtain $\mathbf{0-1}$ vector z^*.

2. Solve controller problem to test whether $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \leq 1$.

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

Algorithm outline

$\longrightarrow$ Maintain a “master (attacker) MIP”:

- Made up of valid inequalities (for the attacker)
- Initially empty

Iterate:

1. Solve master MIP, obtain $\mathbf{0-1}$ vector z^*.

2. Solve controller problem to test whether $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \le 1$.

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 $\mathbf{0-1}$ vector z^*.

2. Solve controller problem to test whether $\boldsymbol{supp(z^*)}$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \leq 1$.

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

Algorithm outline

$\rightarrow$ Maintain a "master (attacker) MIP":

- Made up of valid inequalities (for the attacker)
- Initially empty

Iterate:

1. Solve master MIP, obtain $\mathbf{0-1}$ vector z^*.

2. Solve controller problem to test whether $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \leq 1$.

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

Algorithm outline

$\longrightarrow$ Maintain a “master (attacker) MIP”:

- Made up of valid inequalities (for the attacker)
- Initially empty

Iterate:

1. Solve master MIP, obtain $\mathbf{0-1}$ vector z^*.

2. Solve controller problem to test whether $supp(z^*)$ is a successful attack:

- If successful, then z^* is an optimal solution
- If not, then for some set of generators $\mathcal{G}$, $t_{supp(z^*)}(\mathcal{G}) \leq 1$.

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

Cutting planes = Benders' cuts

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

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

s.t.

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

$$y \;\in\; P$$

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

$\rightarrow$ If $t_{suppt(\hat{z})}(\mathcal{G}) \leq 1$, 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 $\mathbf{0}-\mathbf{1}$ vector $\hat{\boldsymbol{z}}$, and a set of generators $\mathcal{G}$,

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

s.t.

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

$$y \;\in\; P$$

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

$\rightarrow$ If $t_{suppt(\hat{z})}(\mathcal{G}) \leq \mathbf{1}$, use LP duality to separate $\hat{\boldsymbol{z}}$,

getting a cut $\alpha^t z \geq \beta$ violated by $\hat{\boldsymbol{z}}$.

Plus:

Given an **unsuccessful** attack z^*,

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

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

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

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

$\longrightarrow$ other definitions of "padding"

Plus:

Given an **unsuccessful** attack z^*,

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

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

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

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

$\longrightarrow$ other definitions of "padding"

Plus:

Given an **unsuccessful** attack z^*,

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

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

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

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

$\longrightarrow$ other definitions of "padding"

Plus, combinatorial relaxations

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

Example: fractional controller

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

Example: fractional controller

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

Example: when an arc is attacked, flow goes to zero, but Ohm's law still applies

IEEE 57 nodes, 78 arcs, 4 generators Entries show: (iteration count), CPU seconds, Attack status (**F** = cardinality too small, **S** = attack success)					
	Attack cardinality				
Min. thrpt	**2**	**3**	**4**	**5**	**6**
0.75	(1), 2, **F**	(2), 3, **S**			
0.70	(1), 1, **F**	(3), 7, **F**	(48), 246, **F**	(51), 251, **S**	
0.60	(2), 2, **F**	(3), 6, **F**	(6), 21, **F**	(6), 21, **S**	
0.50	(2), 2, **F**	(3), 7, **F**	(6), 13, **F**	(6), 13, **F**	(6), 13, **S**
0.30	(1), 1, **F**	(2), 3, **F**	(2), 3, **F**	(2), 3, **F**	(2), 3, **F**

Table: ***IEEE 57-bus test case***

118 nodes, 186 arcs, 17 generators Entries show: (iteration count), CPU seconds, Attack status (**F =** cardinality too small, **S** = attack success)			
	Attack cardinality		
Min.	**2**	**3**	**4**
thrpt			
0.92	(4), 18, **S**		
0.90	(5), 180, **F**	(6), 193, **S**	
0.88	(4), 318, **F**	(6), 595, **S**	
0.84	(2), 23, **F**	(6), 528, **F**	(148), 6562, **S**
0.80	(2), 18, **F**	(5), 394, **F**	(7), 7755, **F**
0.75	(2), 14, **F**	(4), 267, **F**	(7), 6516, **F**

Table: ***IEEE 118-bus test case***

98 nodes, 204 arcs Entries show: (iteration count), time, Attack status (**F =** cardinality too small, **S** = attack success)			
12 generators			
	Attack cardinality		
Min. throughput	**2**	**3**	**4**
0.92	(2), 318, **F**	(11), 7470, **F**	(14), 11819, **S**
0.90	(2), 161, **F**	(11), 14220, **F**	(18), 16926, **S**
0.88	(2), 165, **F**	(10), 11178, **F**	(15), 284318, **S**
0.84	(2), 150, **F**	(9), 4564, **F**	(16), 162645, **F**
0.75	(2), 130, **F**	(9), 7095, **F**	(15), 93049, **F**

98 nodes, 204 arcs Entries show: (iteration count), time, Attack status (**F** = cardinality too small, **S** = attack success)			
15 generators			
	Attack cardinality		
Min. throughput	**2**	**3**	**4**
0.94	(2), 223, **F**	(11), 654, **S**	
0.92	(2), 201, **F**	(11), 10895, **F**	(18), 11223, **S**
0.90	(2), 193, **F**	(11), 6598, **F**	(16), 206350, **S**
0.88	(2), 256, **F**	(9), 15445, **F**	(18), 984743, **F**
0.84	(2), 133, **F**	(9), 5565, **F**	(15), 232525, **F**
0.75	(2), 213, **F**	(9), 7550, **F**	(11), 100583, **F**

Min. Throughput	Min. Attack Size	Time (sec.)
0.95	2	2
0.90	3	20
0.85	4	246
0.80	5	463
0.75	6	2158
0.70	6	1757
0.65	7	3736
0.60	7	1345
0.55	8	2343
0.50	8	1328

Table: 49 nodes, 84 arcs, one configuration

A different model

What are we looking for? **“Hidden”**, **“small”**, **“counterintuitive”** weaknesses of a grid.

→ 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

What are we looking for? **"Hidden"**, **"small"**, **"counterintuitive"** weaknesses of a grid.

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

$\longrightarrow$ Allow the adversary to selectively place stress on the grid in order to cause failure

$\longrightarrow$ Allow the adversary the ability to **exceed** the laws of physics, in a limited way, so as to cause failure

A different model

What are we looking for? **“Hidden”**, **“small”**, **“counterintuitive”** weaknesses of a grid.

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

$\longrightarrow$ Allow the adversary to selectively place stress on the grid in order to cause failure

$\longrightarrow$ Allow the adversary the ability to **exceed** the laws of physics, in a limited way, so as to cause failure

A different model

What are we looking for? **“Hidden”**, **“small”**, **“counterintuitive”** weaknesses of a grid.

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

$\longrightarrow$ Allow the adversary to selectively place stress on the grid in order to cause failure

$\longrightarrow$ Allow the adversary the ability to **exceed** the laws of physics, in a limited way, so as to cause failure

Power flows (again)

A **power flow** is a solution f, θ 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.

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

Power flows (again)

A **power flow** is a solution f, θ 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.

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

Power flows (again)

A **power flow** is a solution f, θ 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.

$\longrightarrow$ For **fixed** b, $f = f(x)$

Power flows (again)

A **power flow** is a solution f, θ 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.

$\rightarrow$ 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 :

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

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 :

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

Lemma (excerpt)

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

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

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

Then

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

(b) For any $(u, v) \notin S$, $\lim_{L \to +\infty} f_{uv}(x) \; = \; \bar{f}_{uv}$.

How to solve the problem

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

Smooth version:

$$\begin{aligned} \max_{x,p} \quad & \sum_{ij} \frac{f_{ij}(x)}{u_{ij}}(p_{ij} - q_{ij}) \\ \text{s.t.} \quad & \sum_{ij}(p_{ij} + q_{ij}) = 1, \\ & x \in F, \quad p, q \geq 0. \end{aligned}$$

(but not concave)

How to solve the problem

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

Smooth version:

$$\begin{aligned} \max_{x,p} \quad & \sum_{ij} \frac{f_{ij}(x)}{u_{ij}} (p_{ij} - q_{ij}) \\ \text{s.t.} \quad & \sum_{ij} (p_{ij} + q_{ij}) \; = \; 1, \\ & x \in F, \quad p, q \geq 0. \end{aligned}$$

(but not concave)

How to solve the problem

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

Smooth version:

$$\begin{aligned}
\max_{x,p} \quad & \sum_{ij} \frac{f_{ij}(x)}{u_{ij}}(p_{ij} - q_{ij}) \\
\text{s.t.} \quad & \sum_{ij}(p_{ij} + q_{ij}) = 1, \\
& x \in F, \quad p, q \geq 0.
\end{aligned}$$

(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 $\frac{\partial^2 F(x,p)}{\partial^2 x,p}$

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 $\frac{\partial^2 F(x,p)}{\partial^2 x,p}$

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 $\boldsymbol{\nabla_{x,p} F(x,p)}$ and Hessian $\boldsymbol{\frac{\partial^2 F(x,p)}{\partial^2 x,p}}$

Some details

Implementation using LOQO (currently testing SNOPT)

Adversarial model:

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

where (this talk):

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

and

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

where

$$\Delta\mathrm{B} \le 40$$

Sample computational experience

Table: ***57 nodes, 78 arcs***

Iteration Limit: 700, $\epsilon = 0.01$

	ΔB 9	18	27	36
Max Cong	1.070	1.190	1.220	1.209
Time (sec)	8	19	19	19
Iterations	339	Limit	Limit	Limit
Exit Status	ϵ-L-opt.	PDfeas. Iter: 700	PDfeas. Iter: 700	PDfeas. Iter: 700

Sample computational experience

Table: *118 nodes, 186 arcs*

Iteration Limit: 700, $\epsilon = 0.01$

	ΔB			
	9	**18**	**27**	**36**
Max Cong	1.807	2.129	2.274	2.494
Time (sec)	88	200	195	207
Iterations	Limit	578	Limit	Limit
Exit Status	PDfeas. Iter: 302	ϵ-L-opt.	PDfeas. Iter: 700	PDfeas. Iter: 700

Sample computational experience

Table: *600 nodes, 990 arcs*

Iteration Limit: **300**, $\epsilon = 0.01$

	ΔB				
	10	**20**	**27**	**36**	**40**
obj	0.571562	1.076251	1.156187	1.088491	1.161887
sec	11848	7500	4502	11251	7800
Its	Limit	210	114	Limit	208
stat	PDfeas Iter: 300	ϵ-L-opt.	ϵ-L-opt.	PDfeas Iter: 300	ϵ-L-opt.

Sample computational experience

Table: *649 nodes, 1368 arcs,* $\Gamma(2)$

Iteration Limit: 500, $\epsilon = 0.01$

	ΔB		
	20	**30**	**40**
Max Cong	(0.06732) 1.294629	1.942652	(0.049348) 1.395284
Time (sec)	66420	36274	54070
Iterations	Limit	374	Limit
Exit Status	DF	ϵ-L-opt.	DF

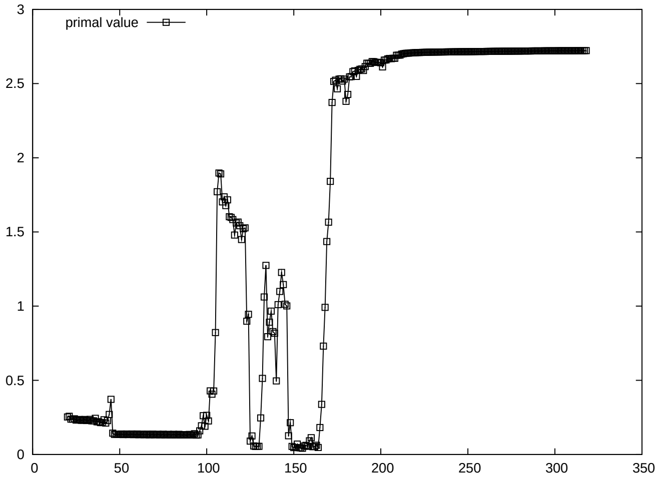

primal value
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
300
350

Table: Attack pattern

$x^u = 20$	$\Delta B = 57$	$x^u = 10$	$\Delta B = 27$	$x^u = 10$	$\Delta B = 36$
Range	**Count**	**Range**	**Count**	**Range**	**Count**
$[1, 1]$	8	$[1, 1]$	1	$[1, 1]$	14
$(1, 2]$	72	$(1, 2]$	405	$(1, 2]$	970
$(2, 3]$	4	$(2, 9]$	0	$(2, 5]$	3
$(5, 6]$	1	$(9, 10]$	3	$(5, 6]$	0
$(6, 7]$	1			$(6, 7]$	1
$(7, 8]$	4			$(7, 9]$	0
$(8, 20]$	0			$(9, 10]$	2

Impact

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