A Review of Recent Developments in Nonlinear Optimization of Electric Power Systems

Daniel Molzahn
Georgia Institute of Technology
(molzahn@gatech.edu)

PSERC Webinar
March 10, 2015
Motivation

US Annual Electricity Generating Capacity Additions and Retirements

New computational tools are needed to economically and reliably operate electric grids with significant renewable generation.

http://teeic.anl.gov/er/transmission/restech/dist/index.cfm


http://teeic.anl.gov/er/transmission/restech/dist/index.cfm

A Typical Day of Solar PV Generation, [Apt and Curtright ‘08]
The Power Flow Equations

- Model the relationship between the voltages and the power injections.

| AC power flow equations | Voltages: $V_i = |V_i|\angle\theta_i$ |
|-------------------------|-----------------------------------|
| $P_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k) \right)$ |
| $Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k) \right)$ |

- Central to many power system optimization and control problems.
  - Optimal power flow, unit commitment, voltage stability, contingency analysis, transmission switching, etc.
Optimal Power Flow

\[
\begin{align*}
\min_{|V|, \theta} & \quad \sum_{i \in G} \left( c_{2i} P_{Gi}^2 + c_{1i} P_{Gi} + c_{0i} \right) \\
\text{subject to} & \quad P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \\
& \quad Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \\
& \quad V_{i}^{\min} \leq |V_i| \leq V_{i}^{\max} \\
& \quad \theta_{ik}^{\min} \leq \theta_i - \theta_k \leq \theta_{ik}^{\max} \\
& \quad |S_{flow,ik}| \leq S_{flow,ik}^{\max}
\end{align*}
\]

\[
\begin{align*}
P_i &= |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right) \\
Q_i &= |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right)
\end{align*}
\]
A Disconnected Feasible Space

- Five-bus example problem
  [Bukhsh, Grothey, McKinnon, & Trodden ‘13]
Another Disconnected Space

- Nine-bus example problem
  [Bukhsh, Grothey, McKinnon, & Trodden ‘13]

Lower limits on voltage magnitudes and reactive power generation

Local Solutions

Global Solution

Lower limits on voltage magnitudes and reactive power generation

[Molzahn ’17]
DC Power Flow Approximation

• Linearization of the power flow equations:

\[ P_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right) \]

\[ Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right) \]

• Advantages:
  – Fast and reliable solution using linear programming.

• Disadvantages:
  – No consideration of voltage magnitudes or reactive power.
  – Approximation error.
**DC Power Flow Approximation**

- **Linearization** of the power flow equations:

\[
P_i = |V_i| \sum_{k=1}^{n} |V_k| \left(G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k)\right)
\]

\[
Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left(G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k)\right)
\]

- **Advantages:**
  - Fast and reliable solution using linear programming.

- **Disadvantages:**
  - No consideration of voltage magnitudes or reactive power.
  - Approximation error.
**DC Power Flow Approximation**

- **Linearization** of the power flow equations:

\[
P_i = \left| V_i \right| \sum_{k=1}^{n} \frac{1}{V_k} \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right)
\]

\[
Q_i = \left| V_i \right| \sum_{k=1}^{n} V_k \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right)
\]

- **Advantages:**
  - Fast and reliable solution using linear programming.

- **Disadvantages:**
  - No consideration of voltage magnitudes or reactive power.
  - Approximation error.
DC Power Flow Approximation

- **Linearization of the power flow equations:**

\[
P_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right)\]

\[
Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right)\]

- **Advantages:**
  - Fast and reliable solution using linear programming.

- **Disadvantages:**
  - No consideration of voltage magnitudes or reactive power.
  - Approximation error.
DC Power Flow Approximation

- Linearization of the power flow equations:

  \[ P_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right) \]

  \[ Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right) \]

- Advantages:
  - Fast and reliable solution using linear programming.

- Disadvantages:
  - No consideration of voltage magnitudes or reactive power.
  - Approximation error.
**DC Power Flow Approximation**

- **Linearization** of the power flow equations:

\[
P_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k) \right)
\]

\[
Q_i = |V_i| \sum_{k=1}^{n} |V_k| \left( G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k) \right)
\]

- **Advantages:**
  - Fast and reliable solution using linear programming.

- **Disadvantages:**
  - No consideration of voltage magnitudes or reactive power.
  - Approximation error.
Current Practice

Analysis using the DC power flow approximation → Engineering intuition and heuristics → Feasible operating point for the AC power flow equations

"I think you should be more explicit here in step two."

Sidney Harris, Science Cartoons Plus
Methods for Handling Nonlinearities

- Local optimization
  Recent successes in Dept. of Energy ARPA-E Grid Optimization Competition

- Approximation

- Convex relaxation

- Convex restriction
Local Optimization

- Seek a “local solution” that is superior to all nearby points but possibly inferior to more distant points.

- Dependent on the initialization.

- Two main classes of tools:
  - Interior point methods (e.g., Ipopt).
  - Sequential quadratic programming methods (e.g., SNOPT).
Dept. of Energy ARPA-E Grid Optimization Competition

• Security-Constrained AC Optimal Power Flow problem
  – Jointly optimize the generators’ real power outputs and voltage magnitude setpoints.
  – N-1 preventative security requirements.
• Security-Constrained AC Optimal Power Flow problem

minimize \quad \text{Piecewise Linear Generation Cost}

subject to

- **Base Case**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Contingency 1**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Contingency 2**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Contingency k**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations
minimize \quad \text{Piecewise Linear Generation Cost}

subject to

- **Reactive Power Limits**
  - $Q_g^\text{min} \leq Q_g \leq Q_g^\text{max}$
  - $V_g^* \leq V_g 

- **Base Case**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Real Power Limits**
  - $P_g^\text{min} \leq P_g \leq P_g^\text{max}$

- **Contingency 1**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Contingency 2**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations

- **Contingency k**
  - Voltage Magnitude Limits
  - Line Flow Limits
  - Real and Reactive Generation Limits
  - AC Power Flow Equations
Dept. of Energy ARPA-E Grid Optimization Competition

- Security-Constrained AC Optimal Power Flow problem
  - Jointly optimize the generators’ real power outputs and voltage magnitude setpoints.
  - N-1 preventative security requirements.

- Key challenges:
  - **Problem size**: up to 30,000 buses and 10,800 contingencies.
  - **Nonlinearity** from the AC power flow equations.
  - **Complementarity conditions** from generator limits.
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
- Evaluate in initial ranked order
- Select contingencies with largest penalties

Master Problem Solve
- Base case + selected contingencies
- Solve using Ipopt

Contingency Selection

Contingency Evaluation

Contingency Evaluation
Our Local Optimization Approach

- Preprocessing
- Base case solution: Ipopt
- Initial Contingency Ranking
- Contingency Selection
  - Evaluate in initial ranked order
  - Select contingencies with largest penalties
- Master Problem Solve
  - Base case + selected contingencies
  - Solve using Ipopt
- Contingency Selection
  - Contingency Evaluation
  - Contingency Evaluation
  - ...
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
- Evaluate in initial ranked order
- Select contingencies with largest penalties

Master Problem Solve
- Base case + selected contingencies
- Solve using Ipopt

Contingency Selection

Contingency Evaluation
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
• Evaluate in initial ranked order
• Select contingencies with largest penalties

Master Problem Solve
• Base case + selected contingencies
• Solve using Ipopt

Contingency Selection

Contingency Evaluation

Contingency Evaluation

Contingency Evaluation
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
- Evaluate in initial ranked order
- Select contingencies with largest penalties

Master Problem Solve
- Base case + selected contingencies
- Solve using Ipopt

Contingency Selection

Contingency Evaluation
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
- Evaluate in initial ranked order
- Select contingencies with largest penalties

Master Problem Solve
- Base case + selected contingencies
- Solve using Ipopt

Contingency Selection

Contingency Evaluation

Contingency Evaluation

Contingency Evaluation
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
  • Evaluate in initial ranked order
  • Select contingencies with largest penalties

Master Problem Solve
  • Base case + selected contingencies
  • Solve using Ipopt

Contingency Selection

Contingency Evaluation
  • In parallel
Our Local Optimization Approach

Preprocessing

Base case solution: Ipopt

Initial Contingency Ranking

Contingency Selection
- Evaluate in initial ranked order
- Select contingencies with largest penalties

Master Problem Solve
- Base case + selected contingencies
- Solve using Ipopt

Contingency Selection

Local Solver
(Ipopt: Interior Point Method)

Contingency Evaluation
Final Results

- Top teams reliably output solutions for large-scale systems with low generation cost and small constraint violation penalties.
## Final Results

**LEADERBOARD - CHALLENGE 1 - FINAL EVENT**

*Updated: 2/12/2020*

<table>
<thead>
<tr>
<th>Top 10 Placement</th>
<th>Division 1</th>
<th>Division 2</th>
<th>Division 3</th>
<th>Division 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization:</td>
<td>Lawrence Livermore National Laboratory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Name:</td>
<td>gallito</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Lead:</td>
<td>Cosmin G. Petra</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Members:</td>
<td>Omar DeGuchy, Ignacio Andres Aravena Solis, Deepak Rajan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Lehigh University</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Team Name:</td>
<td>GO-SNIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Lead:</td>
<td>Frank Edward Curtis</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Members:</td>
<td>Daniel Kenneth Mabohm, Andreas Weschler, Ermin Wei, Elizabeth Wong</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Georgia Institute of Technology</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Team Name:</td>
<td>GMI-GO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Lead:</td>
<td>Xu Sun</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Members:</td>
<td>Santanu Subhas Day, Amin Gholami, Kaithao Sun, Shixuan Zhang</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Individuals</td>
<td>8</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Team Name:</td>
<td>BAT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Lead:</td>
<td>Andrew George Telyatnik</td>
<td>8</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Members:</td>
<td>Oleg Mikhailovich Strelnikov</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization:</td>
<td>Individuals</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Name:</td>
<td>gravityx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team Lead:</td>
<td>Nathan Lemons</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Members:</td>
<td>Hassan Lionel Hijazi</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Local Solver (Ipopt used by at least four of the top five teams)
Final Results

• Top teams reliably output solutions for large-scale systems with low generation cost and small constraint violation penalties.

**Takeaway**: State-of-the-art nonlinear solvers are capable of jointly optimizing real power and voltage magnitude setpoints in security-constrained AC optimal power flow problems!
Methods for Handling Nonlinearities

- Local optimization
- Approximation
- Convex relaxation
- Convex restriction
Power Flow Approximations

- Approximate nonlinearities using assumptions regarding typical system characteristics.
- DC power flow for transmission systems.

\[
P_i = |V_i| \sum_{k=1}^{n} |V_k| \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} (G_{ik} \cos (\theta_i - \theta_k) + B_{ik} \sin (\theta_i - \theta_k))
\]

\[
Q_i = |V_i| \sum_{k=1}^{n} |V_k| (G_{ik} \sin (\theta_i - \theta_k) - B_{ik} \cos (\theta_i - \theta_k))
\]
**Power Flow Approximations**

- Approximate nonlinearities using assumptions regarding typical system characteristics.
- DC power flow for transmission systems.
- Linearized DistFlow for distribution systems. [Baran & Wu ‘89]

\[ P_{ik} = R_{ik} \ell_{ik} - P_k + \sum_{m:k \rightarrow m} P_{km} \]

\[ Q_{ik} = X_{ik} \ell_{ik} - Q_k + \sum_{m:k \rightarrow m} Q_{km} \]

\[ |V_k|^2 = |V_i|^2 - 2(R_{ik} P_{ik} + X_{ik} Q_{ik}) + (R^2_{ik} + X^2_{ik}) \ell_{ik} \]

\[ \ell_{ik} |V_i|^2 = P^2_{ik} + Q^2_{ik} \]
Power Flow Approximations

- Approximate nonlinearities using assumptions regarding typical system characteristics.
- DC power flow for transmission systems.
- Linearized DistFlow for distribution systems. [Baran & Wu ‘89]
- Many recently proposed alternatives!

Foundations and Trends® in Electric Energy

A Survey of Relaxations and Approximations of the Power Flow Equations

Daniel K. Molzahn
Georgia Institute of Technology
molzahn@gatech.edu

Ian A. Hiskens
University of Michigan
hiskens@umich.edu
Optimal Adaptive Approximations

• Compute linear approximations that minimize the worst-case error for a specific system and operating range

[Misra, Molzahn, & Dvijotham PSCC‘18],
[Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech‘19]
Optimal Adaptive Approximations

- Compute linear approximations that minimize the worst-case error for a specific system and operating range

[Misra, Molzahn, & Dvijotham PSCC‘18], [Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech‘19]
Optimal Adaptive Approximations

- Compute linear approximations that minimize the worst-case error for a specific system and operating range

[Misra, Molzahn, & Dvijotham PSCC’18],
[Mülpfordt, Molzahn, Hagenmeyer, & Misra PowerTech’19]
**Optimal Adaptive Approximations**

- Compute linear approximations that minimize the worst-case error for a specific system and operating range.

---

[Misra, Molzahn, & Dvijotham PSCC‘18],
[Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech‘19]
Optimal Adaptive Approximations

- Compute linear approximations that minimize the worst-case error for a specific system and operating range

[Misra, Molzahn, & Dvijotham PSCC‘18],
[Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech’19]
Optimal Adaptive Approximations

- Compute linear approximations that minimize the worst-case error for a specific system and operating range.

[Misra, Molzahn, & Dvijotham PSCC‘18],
[Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech’19]
Optimal Adaptive Approximations

• Compute linear approximations that minimize the worst-case error for a specific system and operating range

[Misra, Molzahn, & Dvijotham PSCC‘18],
[Mühlpfordt, Molzahn, Hagenmeyer, & Misra PowerTech‘19]
Methods for Handling Nonlinearities

- Local optimization
- Approximation
- Convex relaxation
- Convex restriction
Power Flow Relaxations

- Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.
**Power Flow Relaxations**

- Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.
Power Flow Relaxations

• Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.
**Power Flow Relaxations**

- Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.

The solution to the relaxation matches the solution to the non-convex problem → *Zero Relaxation Gap*
Power Flow Relaxations

• Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.

The solution to the relaxation does not match the solution to the non-convex problem
→ Non-zero Relaxation Gap
Power Flow Relaxations

• Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.

• Three main advantages over local solution algorithms:

  1. Bounds the optimal objective value.
  2. Provides a sufficient condition for infeasibility.
  3. Solutions which satisfy an easily checkable conditions are guaranteed to be globally optimal.

Power Flow Relaxations

- Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.

- Three main advantages over local solution algorithms:
  1. Bounds the optimal objective value.
  2. Provides a sufficient condition for infeasibility.
  3. Solutions which satisfy an easily checkable conditions are guaranteed to be globally optimal.

Power Flow Relaxations

• Relax nonlinearities using less stringent conditions in order to enclose the non-convex feasible space within a larger convex space.

• Three main advantages over local solution algorithms:

1. Bounds the optimal objective value.
2. Provides a sufficient condition for infeasibility.
3. Solutions which satisfy an easily checkable conditions are guaranteed to be globally optimal.

**Power Flow Relaxations**

- **Relax nonlinearities** using less stringent conditions in order to **enclose the non-convex feasible space within a larger convex space**.

- **Three main advantages over local solution algorithms:**
  1. Bounds the optimal objective value.
  2. Provides a sufficient **condition for infeasibility**.
  3. **Solutions which satisfy an easily checkable conditions are guaranteed to be globally optimal.**

---

Example: The QC Relaxation

• Based on the polar form of the power flow equations:

\[
P_{ik} = g_{ik} |V_i|^2 - g_{ik} |V_i| |V_k| \cos(\theta_i - \theta_k) - b_{ik} |V_i| |V_k| \sin(\theta_i - \theta_k)
\]

\[
Q_{ik} = -(b_{c,ik}/2 + b_{ik}) |V_i|^2 - g_{ik} |V_i| |V_k| \sin(\theta_i - \theta_k) + b_{ik} |V_i| |V_k| \cos(\theta_i - \theta_k)
\]

Trilinear monomials in variables representing \(|V_i|, |V_k|\), and convex envelopes for \(\cos(\theta_i - \theta_k)\) or \(\sin(\theta_i - \theta_k)\):

\[
|V_i| \cdot |V_k| \cdot \cos(\theta_i - \theta_k),
\]

\[
|V_i| \cdot |V_k| \cdot \sin(\theta_i - \theta_k).
\]

[Coffrin, Hijazi & Van Hentenryck ‘15]
Example: The QC Relaxation

- Constructs convex envelopes around the sine and cosine functions in the power flow equations with polar voltages.

\[
\sin(\theta_i - \theta_k) \quad \quad \cos(\theta_i - \theta_k)
\]
Example: The QC Relaxation

- Constructs **convex envelopes** around the sine and cosine functions in the power flow equations with polar voltages.

\[
\sin (\theta_i - \theta_k)
\]

\[
\cos (\theta_i - \theta_k)
\]

[Coffrin, Hijazi & Van Hentenryck ‘15]
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms

Non-convex space
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms

Split feasible space

Non-convex space
Methods for Tightening Relaxations

- Combining non-dominated relaxations
- Valid inequalities
- Branch-and-bound algorithms

![Diagram showing a non-convex space, split feasible space, and creating two relaxations.](image)
Applications of Convex Relaxations

(Five Examples)
1. Global Solution via Spatial Branch-and-Bound

1. Segment the feasible space into adjoining subregions.

2. In each subregion, compute an upper bound using a local solver and a lower bound using a relaxation.

3. Eliminate subregions whose lower bounds are greater than an upper bound obtained in any other subregion.

4. Iterate until finding an upper bound that is sufficiently close to the least lower bound.
1. Global Solution via Spatial Branch-and-Bound

1. Segment the feasible space into adjoining subregions.

2. In each subregion, compute an upper bound using a local solver and a lower bound using a relaxation.

3. Eliminate subregions whose lower bounds are greater than an upper bound obtained in any other subregion.

4. Iterate until finding an upper bound that is sufficiently close to the least lower bound.
1. Global Solution via Spatial Branch-and-Bound

1. Segment the feasible space into adjoining subregions.

2. In each subregion, compute an upper bound using a local solver and a lower bound using a relaxation.

3. Eliminate subregions whose lower bounds are greater than an upper bound obtained in any other subregion.

4. Iterate until finding an upper bound that is sufficiently close to the least lower bound.
2. **Certify Power Flow Insolvability**

- Compute upper bounds on the **maximum achievable loading**.

Are the power flow equations feasible for a specified loading scenario?

[Molzahn, Lesieutre, & DeMarco ’13]
2. Certify Power Flow Insolvability

- Compute upper bounds on the maximum achievable loading.

![Diagram showing the relationship between power demand and voltage magnitude.]

[Molzahn, Lesieutre, & DeMarco ’13]
2. **Certify Power Flow Insolvability**

- Compute upper bounds on the maximum achievable loading.

![Graph showing voltage magnitude vs. power demand]

Upper bound from a relaxation

[Source: Molzahn, Lesieutre, & DeMarco ’13]
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a security margin, interpreted as tightened constraints.

\[ i_{\ell m}^{max} \]

Current flow magnitude

Scheduled flow from OPF solution

[Molzahn & Roald ’18]
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a **security margin**, interpreted as **tightly constraints**.

![Diagram](source)

- **Current flow magnitude**
- **Scheduled flow from OPF solution**
- **Realized flow**

[Source: Molzahn & Roald ’18]
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a security margin, interpreted as tightened constraints.

---

Current flow magnitude

\[ i_{\text{lm}} \]

\[ i_{\text{max}} \]

Scheduled flow from OPF solution

Realized flow

[1] Molzahn & Roald ’18
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a security margin, interpreted as tightened constraints.

\[ i_{\ell m}^{\text{max}} - \lambda i_{\ell m} \]

Current flow magnitude

Scheduled flow from OPF solution

Realized flow

[Molzahn & Roald ’18]
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a security margin, interpreted as tightened constraints.

[Diagram showing current flow magnitude, scheduled flow from OPF solution, and realized flow.]

Constraint Tightening, $\lambda i_{\ell m}$

Scheduled flow from OPF solution

Realized flow
3. Robust Optimal Power Flow

- Avoid constraint violations by enforcing a security margin, interpreted as tightened constraints.

Define appropriate constraint tightenings using convex relaxations.

\[ i_{lm}^{\max} - \lambda i_{lm} \]

\[ i_{lm}^{\max} \]

Current flow magnitude

Constraint Tightening, \( \lambda i_{lm} \)

Scheduled flow from OPF solution

Realized flow

[Molzahn & Roald ’18]
Robust OPF Example

- 6-bus system “case6ww”
  - ±5% uncertainty in each load demand

Feasible Space

Uncertainty Realizations

[Molzahn & Roald ’18]
Robust OPF Example

- 6-bus system “case6ww”
  - ±5% uncertainty in each load demand

Feasible Space

Uncertainty Realizations

[Molzahn & Roald ’18]
4. Certify Distribution Grid Security

- For specified ranges of variable power injections, use convex relaxations to certify that limited measurements and control are sufficient to preclude constraint violations.

- Certify security if upper bounds on worst-case violations are within desired operational limits.

Stay Tuned!  PSERC Project T-64: Who Controls the DERs?

Insufficient measurement and control to certify security

Voltage control at bus 11 is sufficient to certify security

[Molzahn & Roald ’19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.
2. Sample points inside the polytope.
3. Calculate the power flow solutions at each sampled point.
4. Classify feasible and infeasible points.

[Molzahn '17], [Venzke, Molzahn, & Chatzivasileiadis ’19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.
2. Sample points inside the polytope.
3. Calculate the power flow solutions at each sampled point.
4. Classify feasible and infeasible points.

[Molzahn ’17], [Venzke, Molzahn, & Chatzivasileiadis ’19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.

2. Sample points inside the polytope.

3. Calculate the power flow solutions at each sampled point.

4. Classify feasible and infeasible points.
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.

2. Sample points inside the polytope.

3. Calculate the power flow solutions at each sampled point.

4. Classify feasible and infeasible points.

[Molzahn '17], [Venzke, Molzahn, & Chatzivasileiadis '19]
5. **Create Machine Learning Datasets**

1. Use convex relaxations to construct an **enclosing polytope**.

2. **Sample points inside the polytope**.

3. Calculate the **power flow solutions** at each sampled point.

4. **Classify feasible and infeasible points**.

---

[Molzahn ‘17], [Venzke, Molzahn, & Chatzivasileiadis ‘19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.

2. Sample points inside the polytope.

3. Calculate the power flow solutions at each sampled point.

4. Classify feasible and infeasible points.

[Molzahn ’17], [Venzke, Molzahn, & Chatzivasileiadis ’19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.

2. Sample points inside the polytope.

3. Calculate the power flow solutions at each sampled point.

4. Classify feasible and infeasible points.

[Molzahn '17],
[Venzke, Molzahn, & Chatzivasileiadis '19]
5. Create Machine Learning Datasets

1. Use convex relaxations to construct an enclosing polytope.

2. Sample points inside the polytope.

3. Calculate the power flow solutions at each sampled point.

4. Classify feasible and infeasible points.

[Molzahn ’17], [Venzke, Molzahn, & Chatzivasileiadis ’19]
Methods for Handling Nonlinearities

- Local optimization

- Approximation

- Convex relaxation

- Convex restriction
Convex Restrictions

• Compute convex regions that are completely contained within the non-convex feasible space.

• Based on fixed-point theorems.

• Dependent on a base point.

[Cui & Sun ’19], [Lee, Nguyen, Dvijotham, Turitsyn ’19]
Convex Restrictions

• Compute convex regions that are completely contained within the non-convex feasible space.

• Based on fixed-point theorems.

• Dependent on a base point.

[Cui & Sun ’19], [Lee, Nguyen, Dvijotham, Turitsyn ’19]
**Convex Restrictions**

- Compute convex regions that are **completely contained within** the non-convex feasible space.

- Based on **fixed-point theorems**.

- Dependent on a **base point**.

[Cui & Sun ’19], [Lee, Nguyen, Dvijotham, Turitsyn ’19]
Convex Restrictions

- Compute convex regions that are completely contained within the non-convex feasible space.
- Based on fixed-point theorems.
- Dependent on a base point.

[Cui & Sun ’19], [Lee, Nguyen, Dvijotham, Turitsyn ’19]
Convex Restrictions

- Compute convex regions that are completely contained within the non-convex feasible space.
- Based on fixed-point theorems.
- Dependent on a base point.

[Cui & Sun ’19], [Lee, Nguyen, Dvijotham, Turitsyn ’19]
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

Figures courtesy of Dongchan Lee
**Application 1: Feasible Path**

- Goal: Compute a path between operating points that is guaranteed to *avoid constraint violations*.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald ’20]
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

Figures courtesy of Dongchan Lee [Lee, Turitsyn, Molzahn, & Roald ’20]
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

Figures courtesy of Dongchan Lee
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

Figures courtesy of Dongchan Lee
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.
**Application 1: Feasible Path**

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

![Diagram](image)

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald ’20]
**Application 1: Feasible Path**

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald ’20]
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.
Application 1: Feasible Path

- Goal: Compute a path between operating points that is guaranteed to avoid constraint violations.
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.

Figures courtesy of Dongchan Lee
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald, to be submitted]
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald, to be submitted]
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald, to be submitted]
Application 2: Robust Optimization

- Goal: Compute an operating point that is robust to variations in the net power injections.

Figures courtesy of Dongchan Lee

[Lee, Turitsyn, Molzahn, & Roald, to be submitted]
Conclusion

• Four approaches for handling power system nonlinearities:
  - Local Optimization
  - Approximation
  - Convex Relaxation
  - Convex Restriction

• Capabilities of each are useful for different applications.

Questions?

Local Optimization

Approximation

Convex Relaxation

Convex Restriction


Dan Molzahn
molzahn@gatech.edu
References


Bound Tightening

• Many of the relaxations require **narrow bounds** on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

[Cofrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

\[ |V_1|^2 \]

\[ |V_2|^2 \]

[Coffrin, Hijazi, Van Hentenryck ‘15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\max |V_1|^2 \text{ subject to } \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
**Bound Tightening**

- Many of the relaxations require **narrow bounds** on voltage angles, voltage magnitudes, etc.

- **Relaxations** can be used to tighten the bounds:

\[
\text{max } |V_1|^2 \text{ subject to } \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.
- Relaxations can be used to tighten the bounds:

\[
\text{max } |V_1|^2 \text{ subject to } \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.
- Relaxations can be used to tighten the bounds:

\[
\max |V_1|^2 \text{ subject to} \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
**Bound Tightening**

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

  \[
  \text{max } |V_1|^2 \text{ subject to } \\
  1) \text{ Operational constraints} \\
  2) \text{ A power flow relaxation}
  \]

  [Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\text{max } & \quad |V_2|^2 \\
\text{subject to } & \quad 1) \text{ Operational constraints} \\
& \quad 2) \text{ A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ’16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\max |V_2|^2 \text{ subject to}
\begin{align*}
1) & \text{ Operational constraints} \\
2) & \text{ A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\max & \quad |V_2|^2 \\
\text{subject to} & \quad 1) \text{ Operational constraints} \\
& \quad 2) \text{ A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\min |V_1|^2 & \text{ subject to} \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
**Bound Tightening**

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.
- Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\text{min} & \quad |V_2|^2 \\
\text{subject to} & \\
1) & \quad \text{Operational constraints} \\
2) & \quad \text{A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]
**Bound Tightening**

- Many of the relaxations require **narrow bounds** on voltage angles, voltage magnitudes, etc.
- **Relaxations** can be used to tighten the bounds:

\[
\text{max } |V_1|^2 \text{ subject to } \\
1) \text{ Operational constraints} \\
2) \text{ A power flow relaxation}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\text{max} & \quad |V_2|^2 \\
\text{subject to} & \quad 1) \text{Operational constraints} \\
& \quad 2) \text{A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

• Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

• Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\min \ |V_1|^2 \ & \text{subject to} \\
1) \ & \text{Operational constraints} \\
2) \ & \text{A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\min & \quad |V_2|^2 \\
\text{subject to} & \quad \text{Operational constraints} \\
\text{2) A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\min \ |V_2|^2 \quad &\text{subject to} \\
1) \ &\text{Operational constraints} \\
2) \ &\text{A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ‘16], [Chen, Atamturk, & Oren ‘16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

\[
\begin{align*}
\text{max} & \quad |V_1|^2 \\
\text{subject to} & \quad 1) \text{Operational constraints} \\
& \quad 2) \text{A power flow relaxation}
\end{align*}
\]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ’16]
Bound Tightening

- Many of the relaxations require narrow bounds on voltage angles, voltage magnitudes, etc.

- Relaxations can be used to tighten the bounds:

  \[ |V_1|^2 \quad |V_2|^2 \]

[Coffrin, Hijazi, Van Hentenryck ’15], [Kocuk, Dey, & Sun ’16], [Chen, Atamturk, & Oren ‘16]