Security constrained optimal power flow problems

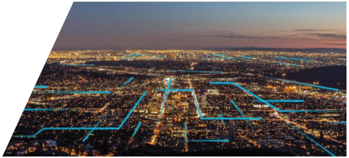
A study of different optimization techniques

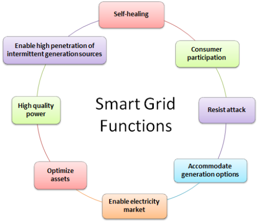
Samuel A. Cruz Alegría
Prof. Dr. Olaf Schenk, Dr. Drosos Kourounis, Juraj Kardoš

(University of Lugano)

January 23, 2018

Motivation




Smart Grid Functions
Self-healing
Consumer participation
Resist attack
Accommodate generation options
Enable electricity market
Optimize assets
High quality power
Enable high penetration of intermittent generation sources

What is an OPF?

What is an OPF?

In general, **OPF** includes any **optimization problem** which seeks to optimize the operation of an **electric power system** (specifically, the generation and transmission of electricity) subject to the physical constraints imposed by electrical laws and engineering limits on the decision variables.

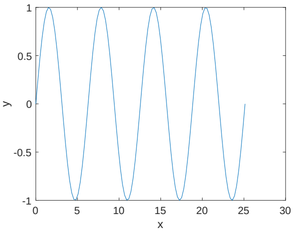


Figure: Nonlinear.

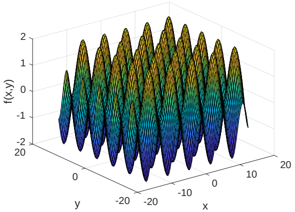
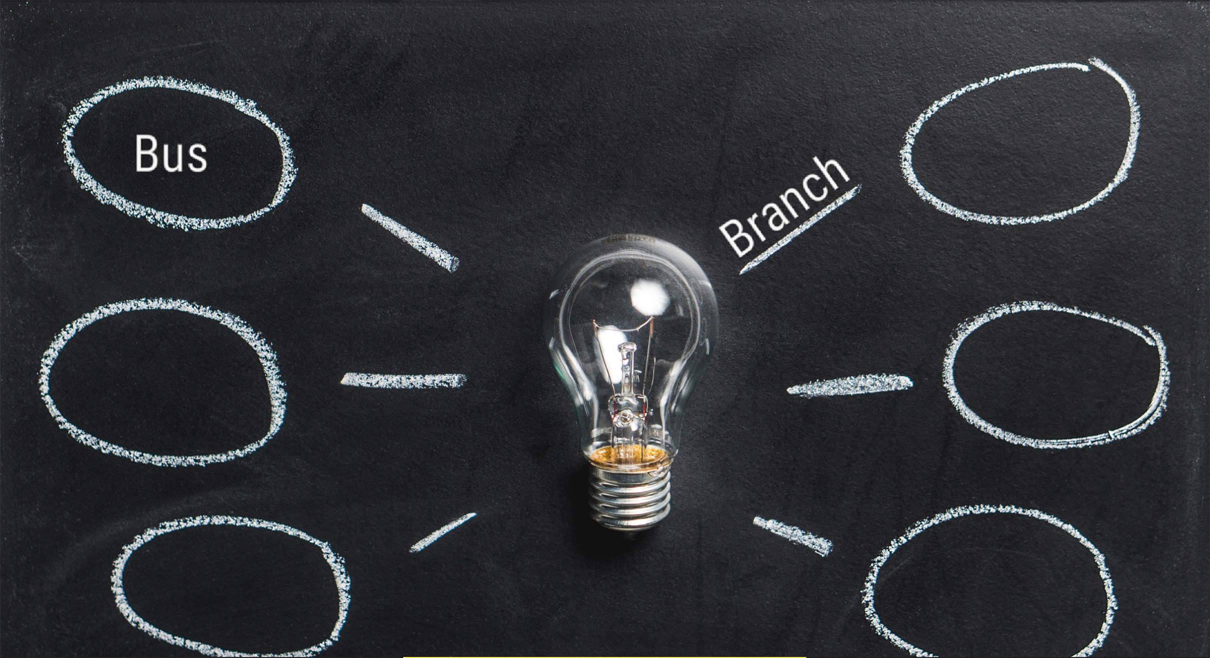


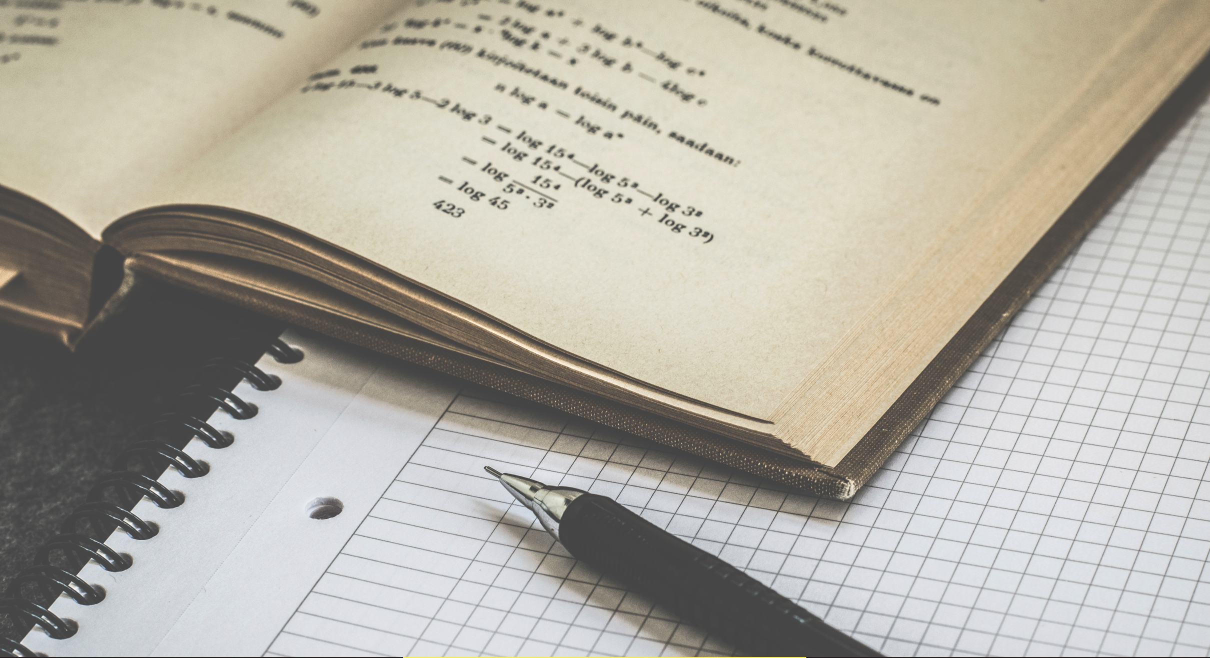
Figure: Nonconvex.


Bus
Branch


Security

What is a SCOPF?

- Also known as security-constrained economic dispatch (SCED).
- It is an OPF formulation which includes power system contingency constraints.
- A **contingency** is defined as an event which removes one or more generators or transmission lines from the power system, increasing the stress on the remaining network.
- SCOPF seeks an optimal solution that remains *feasible* under any of a pre-specified set of likely contingency events.
- The justification for the restriction is that SCOPF mitigates the risk of a system failure (**blackout**) should one of the contingencies occur.


n log a = log aⁿ
= log 15⁴—log 5²—log 3²
= log 15⁴—(log 5² + log 3²)
= log 15⁴/5²·3²
= log 45
423

What is a SCOPF?

The SCOPF problem is formulated as follows:

$$\underset{\boldsymbol{\theta},\mathbf{v},\mathbf{p},\mathbf{q}}{\text{minimize}} \sum_{n=1}^{N_c} \sum_{l=1}^{N_g} a_l(\mathbf{p}_n^l)^2 + b_l \mathbf{p}_n^l + c_l \tag{1a}$$

$$\text{subject to } \forall n = 0, 1, 2, \ldots, N_c,$$

$$\operatorname{diag}(\underline{\mathbf{v}}_{\mathbf{n}}) \left(\underline{\mathbf{Y}}_{\mathbf{n}}^{\mathbf{B}} \underline{\mathbf{v}}_{\mathbf{n}} \right)^* = \mathbf{C}^{\mathbf{B}}(\mathbf{p}_n + j\mathbf{q}_n) - \underline{\mathbf{s}}_{\mathbf{n}}^{\mathbf{D}}, \tag{1b}$$

$$\underline{\mathbf{v}}_{\mathbf{n}} = \operatorname{diag}(\underline{\mathbf{v}}_{\mathbf{n}}) \exp(j\boldsymbol{\theta}_n), \tag{1c}$$

$$\left| \operatorname{diag}(C^L \underline{\mathbf{v}}_{\mathbf{n}}) \left(\underline{Y}_n^L \underline{\mathbf{v}}_{\mathbf{n}} \right)^* \right| \leq s^{L,\max}, \tag{1d}$$

$$\boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta}_n \leq \boldsymbol{\theta}^{\max}, \quad \mathbf{v}^{\min} \leq \underline{\mathbf{v}}_{\mathbf{n}} \leq \mathbf{v}^{\max}, \tag{1e}$$

$$\mathbf{p}^{\min} \leq \mathbf{p}_n \leq \mathbf{p}^{\max}, \quad \mathbf{q}^{\min} \leq \mathbf{q}_n \leq \mathbf{q}^{\max}, \tag{1f}$$

$$\forall b \in \mathcal{B}_{PV} : \mathbf{v}_{\mathbf{n}} = \mathbf{v}_{n0}, \quad \forall g \in \mathcal{B}_{PV} : \mathbf{p}_{\mathbf{n}} = \mathbf{p}_{n0}. \tag{1g}$$

What is a SCOPF?

For each contingency $c \in C$, the post-contingency power flow must remain feasible for the original decision variables:

- The power flow equations must have a solution.
- The contingency state variables must remain within limits.
- Any inequality constraints, such as branch flow limits, must be satisfied.

Project's Purpose

Main Contribution

- The contribution of this work is the implementation of the **interface** of **SCOPF** problem to **Optizelle** solver and the *analysis of its computational complexity*.
- Implementation of the SCOPF problem and its interface to the **IPOPT** solver were created **before** starting this project.
- Important to analyze solvers given that they are a **bottleneck** for energy system modelers.


OPTIZELLE
VS
IPOPT

Frameworks

Frameworks

- MATPOWER
- IPOPT
- Optizelle

MATPOWER

- MATPOWER is an *open-source* software package for MATLAB including functions for both conventional PF and OPF.
- Its modular structure facilicates the extension of the standard OPF model by additional security constraints, yielding the SCOPF problem.
- The MATPOWER **case format** is a set of standard matrix structures used to store power systems case data.

IPOPT

- **I**nterior **P**oint **OPT**imizer (pronounced eye-pea-opt).
- An *open-source* software package for *large-scale* nonlinear optimization.
- It can be used to solve unconstrained, equality constrained, inequality constrained, and constrained problems.

Optizelle

Optizelle (pronounced op-tuh-zel) is an *open-source* software library designed to solve *general purpose* nonlinear optimization problems of the following form(s): unconstrained, equality constrained, inequality constrained, and constrained.

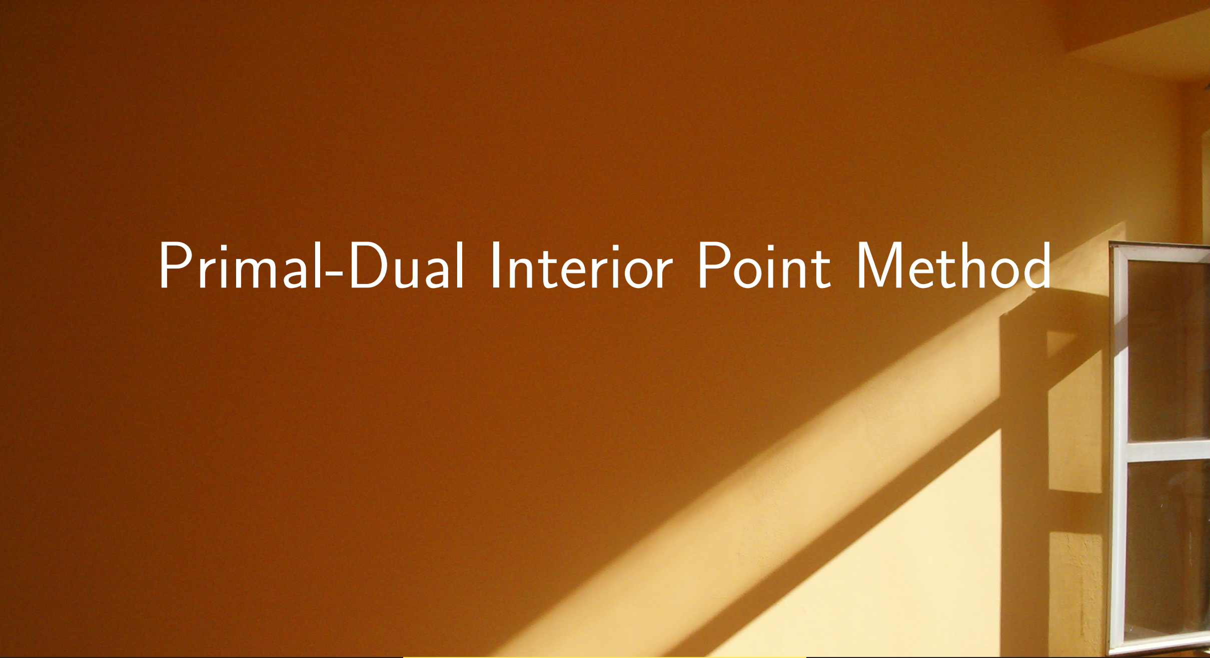
Problem Types

Unconstrained	**Equality Constrained**
$\underset{\mathbf{x}\in X}{\text{minimize}} \quad f(\mathbf{x})$	$\underset{\mathbf{x}\in X}{\text{minimize}} \quad f(\mathbf{x})$ $\text{subject to} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0}.$
Inequality Constrained	**Constrained**
$\underset{\mathbf{x}\in X}{\text{minimize}} \quad f(\mathbf{x})$ $\text{subject to} \quad \mathbf{h}(\mathbf{x}) \geq \mathbf{0}.$	$\underset{\mathbf{x}\in X}{\text{minimize}} \quad f(\mathbf{x})$ $\text{subject to} \quad \mathbf{g}(\mathbf{x}) = \mathbf{0},$ $\mathbf{h}(\mathbf{x}) \geq \mathbf{0}.$

Pros and Cons

Solver	*Main Advantage*	*Main Disadvantage*
IPOPT	Designed for large-scale problems.	Hessian of the Lagrangian required.
Optizelle	Objective function, equality constraints and inequality constraints can be specified separately.	Inequality constraints need to be *affine*.

Primal-Dual Interior Point Method



Primal-Dual Interior Point Method

- The primal-dual interior point method is used by IPOPT and Optizelle to solve contrained problems.
- We used **slack variables** to change our inequality constraints to equality constraints.

Primal-Dual Interior Point Method

$$\underset{\mathbf{x} \in X}{\text{minimize}} \; f(\mathbf{x}) \tag{2a}$$

$$\text{subject to } \mathbf{g}(\mathbf{x}) = \mathbf{0}, \tag{2b}$$

$$\mathbf{h}(\mathbf{x}) - \mathbf{s} = \mathbf{0}, \tag{2c}$$

$$\mathbf{s} \geq \mathbf{0}, \tag{2d}$$

$$\mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}. \tag{2e}$$

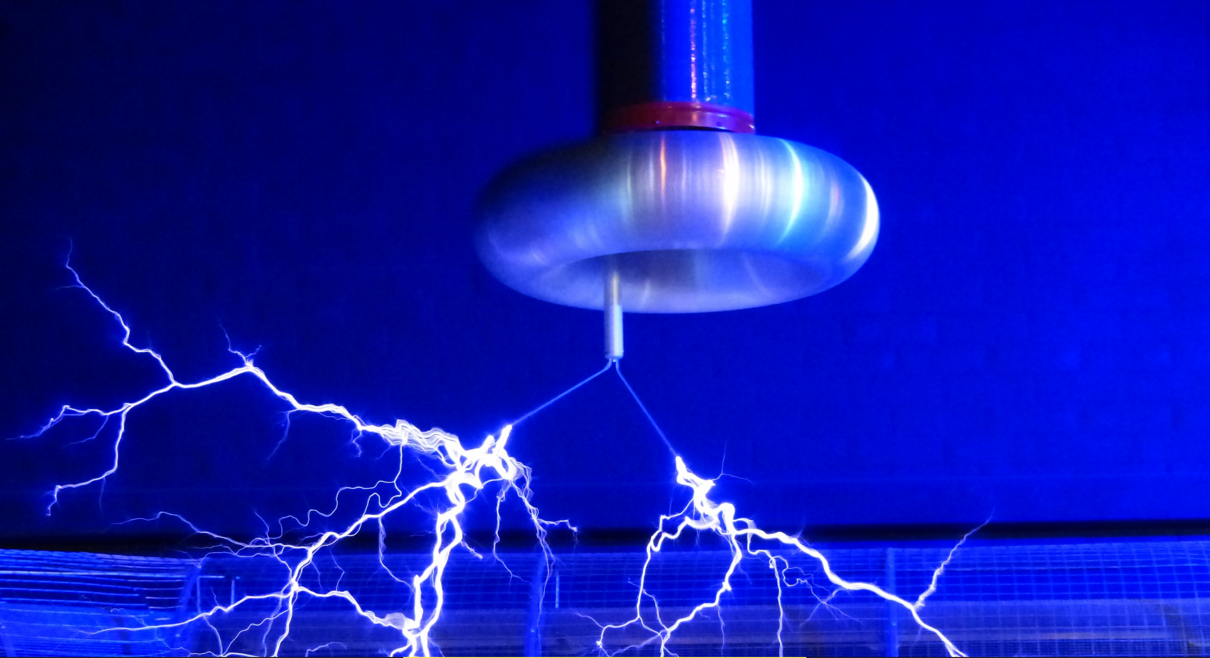
Primal-Dual Interior Point Method

Adding a logarithmic barrier term with weight μ, we solve a sequence of barrier problems with a decreasing value of μ. For $\mu \to 0$, the **solution of the barrier problem converges to the solution of the original problem** with inequality constraints $\mathbf{s} \geq \mathbf{0}$, with $\mu > 0$ being the required optimality conditions.

$$\underset{\mathbf{x} \in X}{\text{minimize}}\ \phi_\mu(\mathbf{x}, \mathbf{s}) := f(\mathbf{x}) - \mu \sum_{i \in I} \ln\left(s^{(i)}\right), \tag{3a}$$

$$\text{subject to } \mathbf{c}_E(\mathbf{x}, \mathbf{s}) = \mathbf{0}, \tag{3b}$$

with $\mathbf{c}_E(\mathbf{x}, \mathbf{s}) := [\mathbf{g}(\mathbf{x}), \mathbf{h}(\mathbf{x}) - \mathbf{s}]^T$.



The Experiment

- Used power grid **case9** provided by MATPOWER.
- Began with an **empty contingency set**, sequentially incrementing the number of contingencies, reaching a **maxmimum of six contingencies**.
- Measured **convergence behaviour** by examining the value of the objective function $f(\mathbf{x})$, the norm of its gradient, $\|\nabla f(\mathbf{x})\|$, and the value of the power flow equations, $\mathbf{g}(\mathbf{x})$.
- Measured the **average time per iteration**.
- Tolerance value of 10^{-6} for all relevant parameters with the exception of the power flow equations, with a tolerance of 10^{-2}.

The Experiment

- IPOPT managed to converge to a feasible solution in **less than fifteen iterations**.
- Optizelle **never** managed to converge to a feasible solution.

Average Time per Iteration

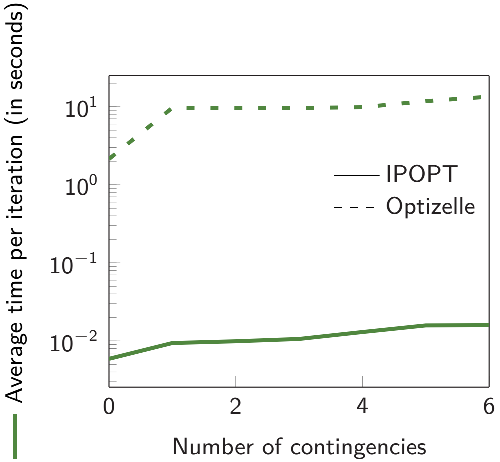
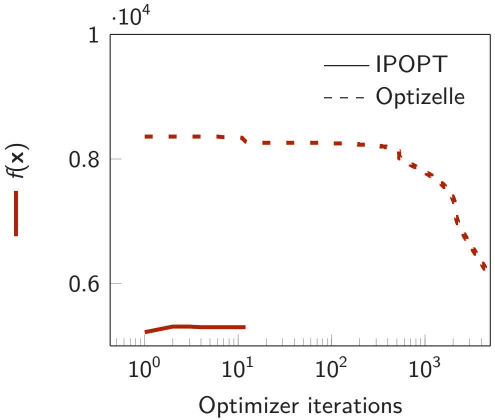


Figure: Average time per iteration.

No Contingencies



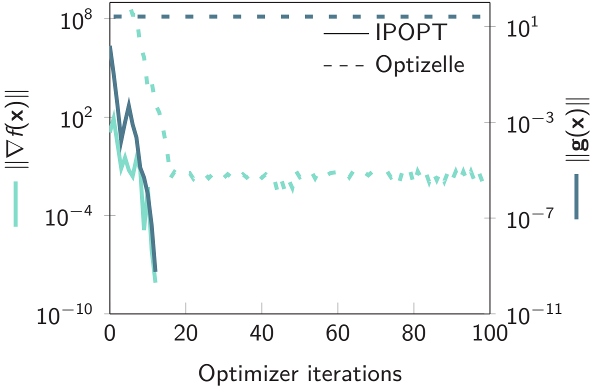


Figure: $|\mathrm{cont}| = 0$, $f(\mathbf{x})$, $\|\nabla f(\mathbf{x})\|$, $\|\mathbf{g}(\mathbf{x})\|$.

Six Contingencies

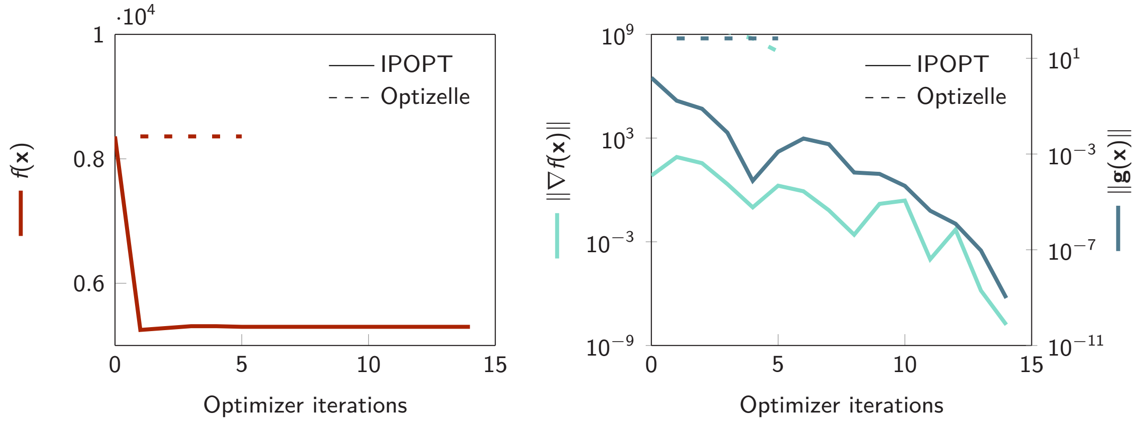


Figure: $|\mathrm{cont}| = 6$, $f(\mathbf{x})$, $\|\nabla f(\mathbf{x})\|$, $\|\mathbf{g}(\mathbf{x})\|$.

Maximum PF Error

$\lvert G \rvert$	*Maximum PF error Optizelle*	*Maximum PF error IPOPT*
0	4.62×10^{-1}	2.94×10^{-13}
1	4.01	5.68×10^{-10}
2	1.55	5.75×10^{-10}
3	1.55	2.72×10^{-10}
4	1.55	5.61×10^{-10}
5	1.55	2.47×10^{-7}
6	1.55	9.64×10^{-10}

Table: Maximum Power Flow (PF) Evaluation Error.

Analysis

- IPOPT was at least **100 times** faster than Optizelle.
- Optizelle's average time per iteration increased **six-fold** going from no contingencies to six contingencies, while IPOPT's average time per iteration increased **three-fold**.

Conclusion

- IPOPT converged to an admissible solution in all cases. Optizelle did not converge to an acceptable solution in any of the cases.
- IPOPT's average time per iteration with respect to a growing contingency set appears to be relatively constant, while Optizelle's does not appear to be so.
- Considering case9 is in fact a small power grid, given the time per iteration of both solvers, it appears IPOPT is scalable to larger problems while Optizelle does not appear to be so.
- IPOPT appears to be more performant than Optizelle.


Thank
You!