Los Alamos
NATIONAL LABORATORY
EST. 1943

advanced network science initiative (ansi)

PowerModels.jl a Brief Introduction

Carleton Coffrin, et. al.

A Bit About Me

- Trained as **Computer Scientist**
  - BS - University of Connecticut
  - PhD - Brown University

Pascal
Van Hentenryck

Laurent
Michel

A Bit About Me

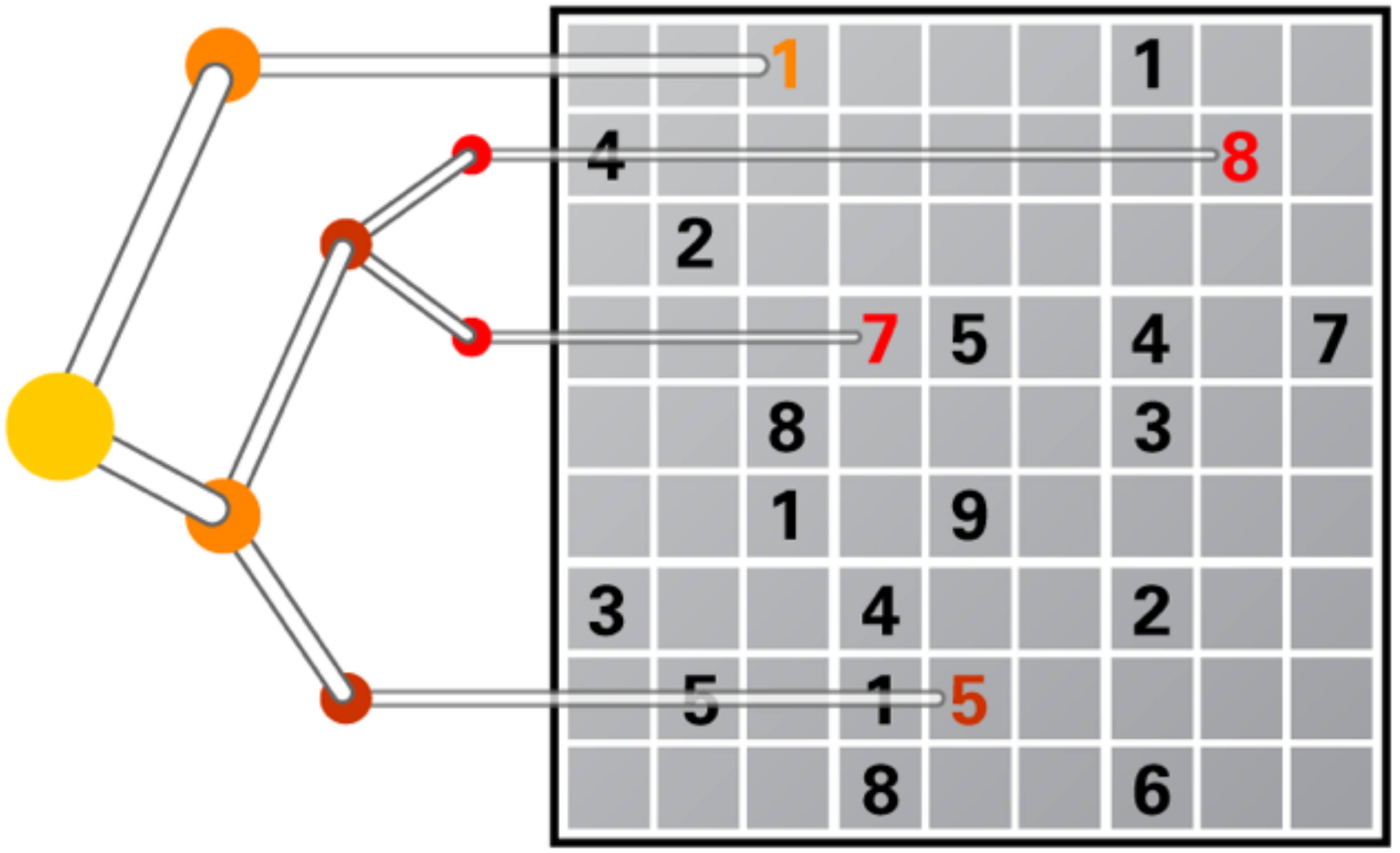
- Trained as **Computer Scientist**
  - BS - University of Connecticut
  - PhD - Brown University
- Know about **CS Stuff**
  - Software Engineering
  - Programming Language Design
  - Computational Research Focus

Pascal
Van Hentenryck

Laurent
Michel

A Bit About Me

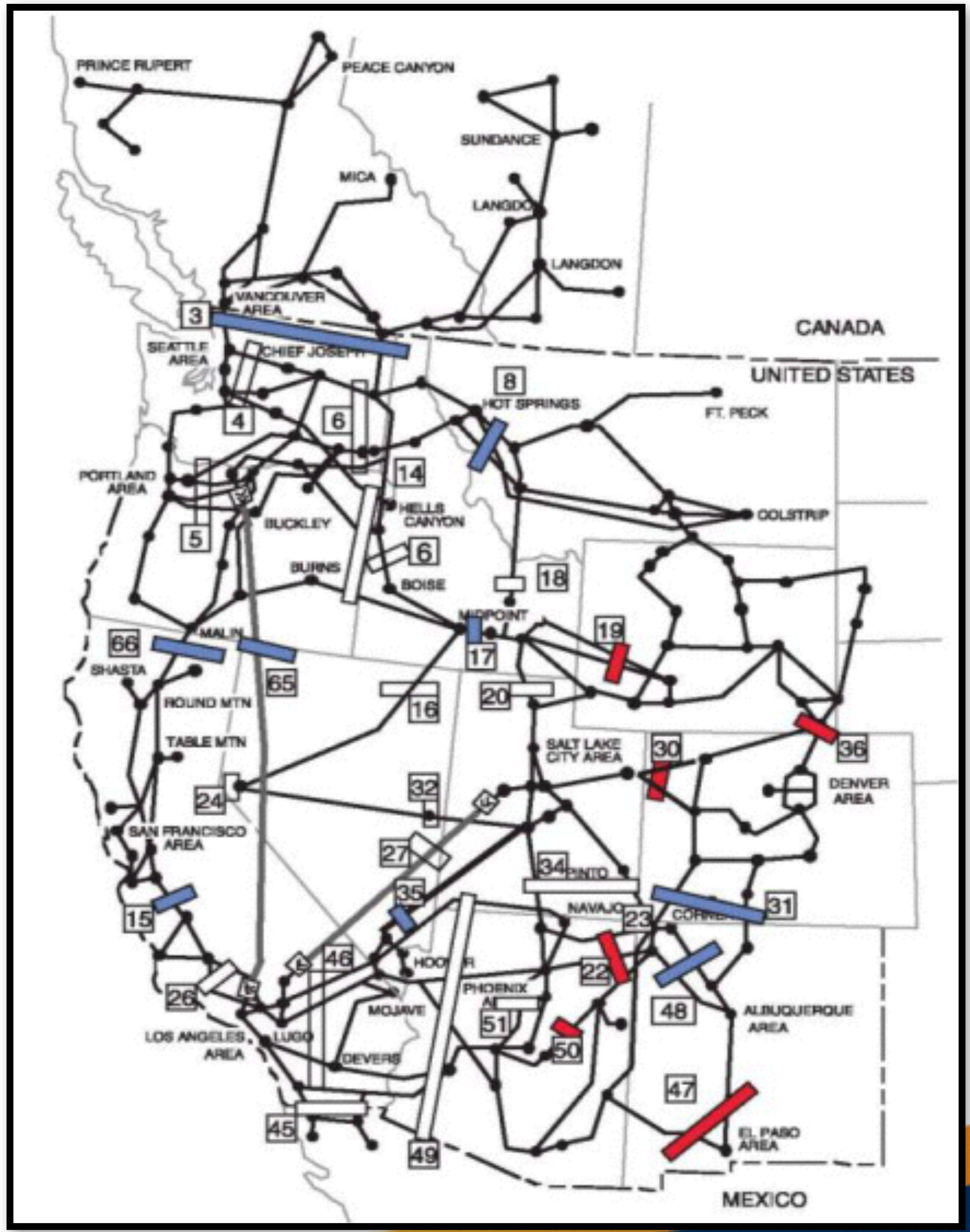
- Discrete Optimization Research
- Generalist
  - Local Search / Heuristics
  - Constraint Programming
  - MIP
  - NLP & MINLP (more recently)



Discrete Optimization

A Bit About LANL

- Advanced Network Science Initiative (**ANSI**)
- 10+ Diverse Staff
    - Optimization, ML, Applied Math, Statistical Physics
- Applications in **complex networks**
    - e.g. Electric Power, Natural Gas, Water
- Developing novel **algorithmic methods**



A Bit About LANL

ANSI
LOVES
JuliaOpt

Outline

- **Motivation**
  - Challenges of R&D in Power Network Optimization
- A **Brief Introduction** to PowerModels.jl
- **Plans** for the Near-Future

Motivation

Power Network Optimization is Complicated

AC Power Flow

$$\theta_{\boldsymbol{r}} = 0$$

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \;\; \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E\cup E^R} q_{ij} \;\; \forall i \in N$$

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \;\; (i,j) \in E \cup E^R$$

$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \;\; (i,j) \in E \cup E^R$$

$$\theta_{ij}^{\Delta} = \theta_i - \theta_j \;\; \forall (i,j) \in E$$

$$p_{ij}^2 + q_{ij}^2 \leq (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \;\; \forall (i,j) \in E \cup E^R$$

Power Network Optimization is Complicated

AC Power Flow

$$\theta_{\boldsymbol{r}} = 0$$

$$p^g_i - \boldsymbol{p}^d_i = \sum_{(i,j)\in E\cup E^R} p_{ij} \quad \forall i \in N$$

$$q^g_i - \boldsymbol{q}^d_i = \sum_{(i,j)\in E\cup E^R} q_{ij} \quad \forall i \in N$$

Flow Conservation (i.e. KCL)

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta^\Delta_{ij}) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta^\Delta_{ij}) \quad (i,j) \in E \cup E^R$$

$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta^\Delta_{ij}) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta^\Delta_{ij}) \quad (i,j) \in E \cup E^R$$

$$\theta^\Delta_{ij} = \theta_i - \theta_j \quad \forall (i,j) \in E$$

$$p_{ij}^2 + q_{ij}^2 \leq (\boldsymbol{s}^{\boldsymbol{u}}_{ij})^2 \quad \forall (i,j) \in E \cup E^R$$

Power Network Optimization is Complicated

AC Power Flow

$$\theta_{\boldsymbol{r}} = 0$$

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \quad \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E\cup E^R} q_{ij} \quad \forall i \in N$$

Flow Conservation (i.e. KCL)

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \quad (i,j) \in E \cup E^R$$

$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \quad (i,j) \in E \cup E^R$$

Line Power Flow (i.e. Ohm's Law)

$$\theta_{ij}^{\Delta} = \theta_i - \theta_j \quad \forall (i,j) \in E$$

$$p_{ij}^2 + q_{ij}^2 \le (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \quad \forall (i,j) \in E \cup E^R$$

Power Network Optimization is Complicated

AC Power Flow

$$\theta_{\boldsymbol{r}} = 0$$

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \quad \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E\cup E^R} q_{ij} \quad \forall i \in N$$

Flow Conservation (i.e. KCL)

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta_{ij}^\Delta) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta_{ij}^\Delta) \quad (i,j) \in E \cup E^R$$

$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta_{ij}^\Delta) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta_{ij}^\Delta) \quad (i,j) \in E \cup E^R$$

Line Power Flow (i.e. Ohm's Law)

$$\theta_{ij}^\Delta = \theta_i - \theta_j \quad \forall (i,j) \in E$$

$$p_{ij}^2 + q_{ij}^2 \le (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \quad \forall (i,j) \in E \cup E^R$$

Line Flow Limits

Power Network Optimization is Complicated

AC Power Flow

$$\theta_{\boldsymbol{r}} = 0$$

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \quad \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E\cup E^R} q_{ij} \quad \forall i \in N$$

Flow Conservation (i.e. KCL)

$$p_{ij} = \boldsymbol{g}_{ij} v_i^2 - \boldsymbol{g}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{b}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \quad (i,j) \in E \cup E^R$$

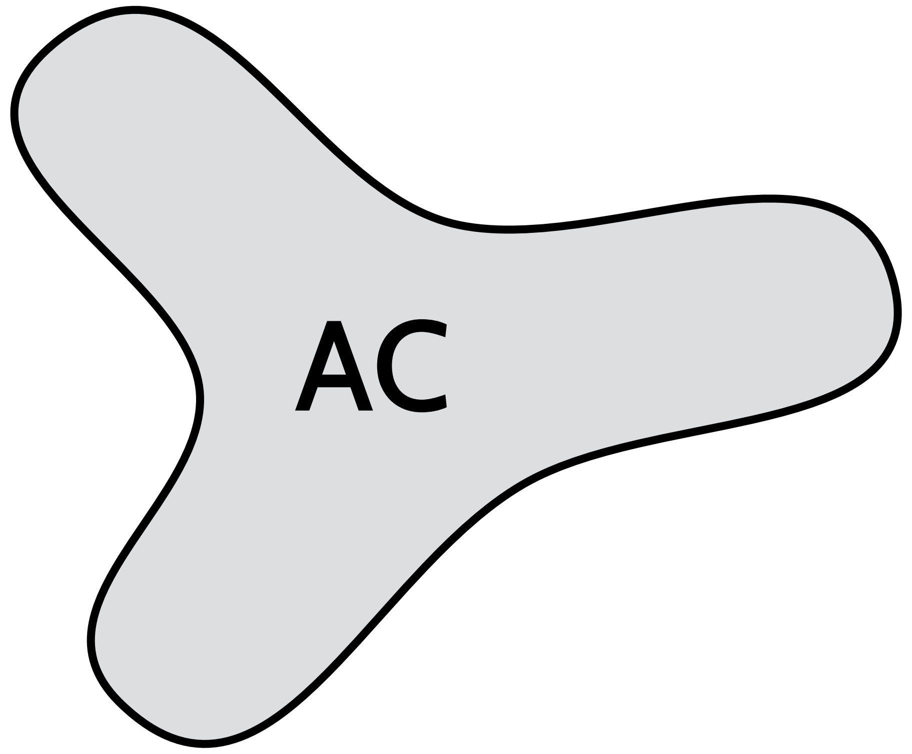
$$q_{ij} = -\boldsymbol{b}_{ij} v_i^2 + \boldsymbol{b}_{ij} v_i v_j \cos(\theta_{ij}^{\Delta}) - \boldsymbol{g}_{ij} v_i v_j \sin(\theta_{ij}^{\Delta}) \quad (i,j) \in E \cup E^R$$

Line Power Flow (i.e. Ohm's Law)

$$\theta_{ij}^{\Delta} = \theta_i - \theta_j \quad \forall (i,j) \in E$$

$$p_{ij}^2 + q_{ij}^2 \le (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \quad \forall (i,j) \in E \cup E^R$$

Line Flow Limits



non-convex

Power Network Optimization is Complicated

IEEE TRANSACTIONS ON POWER SYSTEMS 1

AC-Feasibility on Tree Networks is NP-Hard

Karsten Lehmann, Alban Grastien, and Pascal Van Hentenryck

***Abstract*—Recent years have witnessed significant interest in convex relaxations of the power flows, with several papers showing that the second-order cone relaxation is tight for tree networks under various conditions on loads or voltages. This paper shows that ac-feasibility, i.e., to find whether some generator dispatch can satisfy a given demand, is NP-hard for tree networks.**

***Index Terms*—Computational complexity, optimal power flow (OPF).**

NOMENCLATURE

$\mathcal{N}$	AC-network.
N	Set of buses.
N_G	Set of generators.
N_L	Set of loads.
i	Bus of a network.
j	Bus of a network.

I. INTRODUCTION

MANY interesting applications in power systems, including optimal power flows, optimize an objective function over the steady-state power flow equations, which are nonlinear and nonconvex. These applications typically include an *ac-feasibility* (AC-FEAS) subproblem: find whether some generator dispatch can satisfy a given demand.

Although the set of ac-feasible solutions is in general a nonconvex set, this does not imply that the ac-feasibility problem is NP-hard,[1] as nonconvexity does not imply NP-hardness. For example, the family of optimization problems $\min y$ such that $0 \leq y \leq \prod_{i=1}^{n} x_i$ where $n \in \mathbb{N}$ has a nonconvex constraint and a nonconvex solution set but the optimal solution is always $y = 0$ and can be trivially computed.

The first NP-hardness proof for ac-feasibility was given for a cyclic network structure in [1]. It relies on a variant of the dc model [2] but uses a sine function around the phase angle dif-

Power Network Optimization is Complicated

DC Power Flow Approximation

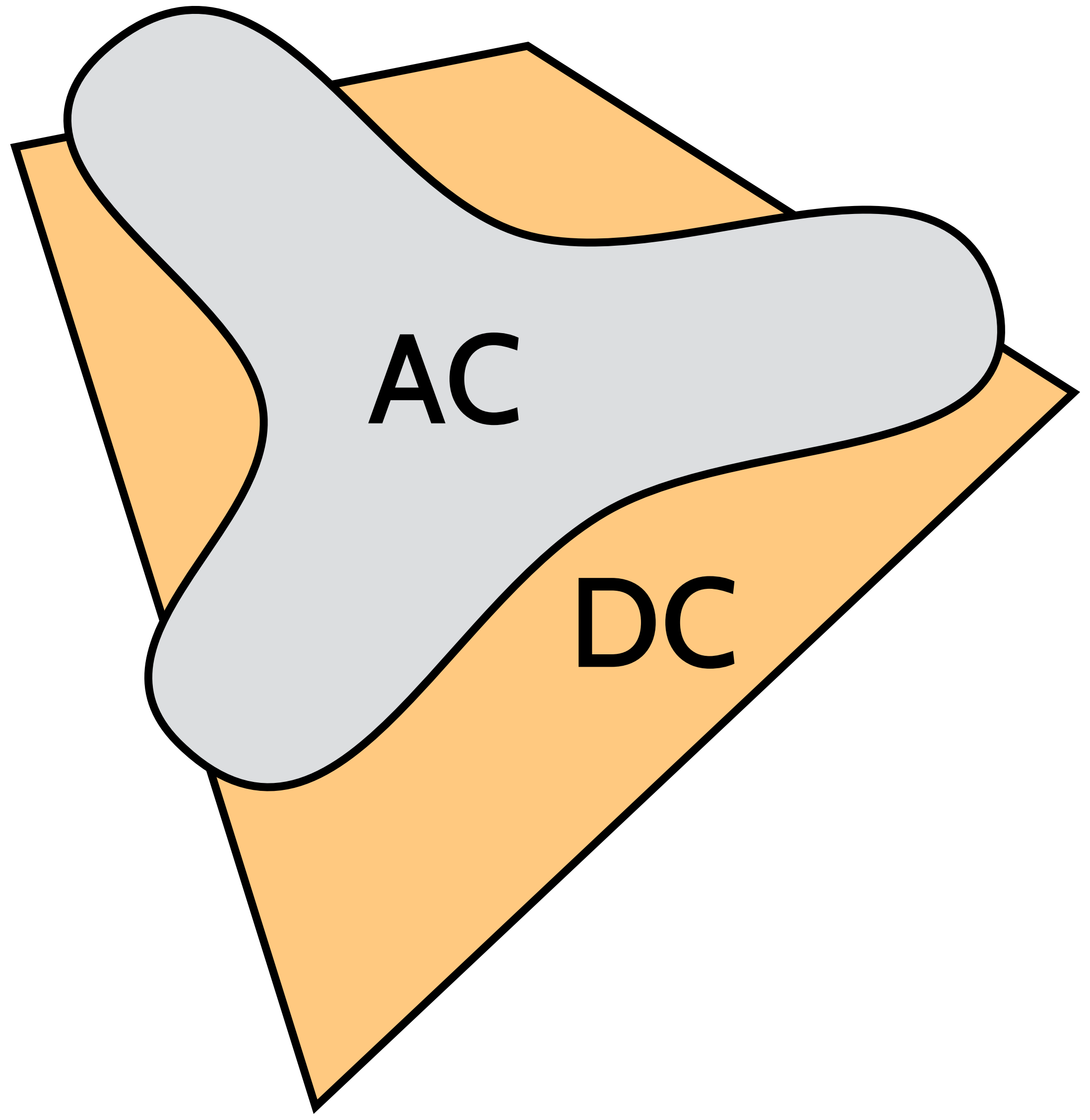
$$\theta_{\boldsymbol{r}} = 0$$

$$p^g_i - \boldsymbol{p}^d_i = \sum_{(i,j)\in E\cup E^R} p_{ij} \;\; \forall i \in N$$

$$p_{ij} = -\boldsymbol{b}_{ij}(\theta^{\Delta}_{ij}) \;\; (i,j) \in E \cup E^R$$

$$\theta^{\Delta}_{ij} = \theta_i - \theta_j \;\; \forall (i,j) \in E$$

$$|p_{ij}| \leq \boldsymbol{s}^{\boldsymbol{u}}_{ij} \;\; \forall (i,j) \in E \cup E^R$$



linear

Power Network Optimization is Complicated

DC Power Flow Approximation

$$\theta_{\boldsymbol{r}} = 0$$

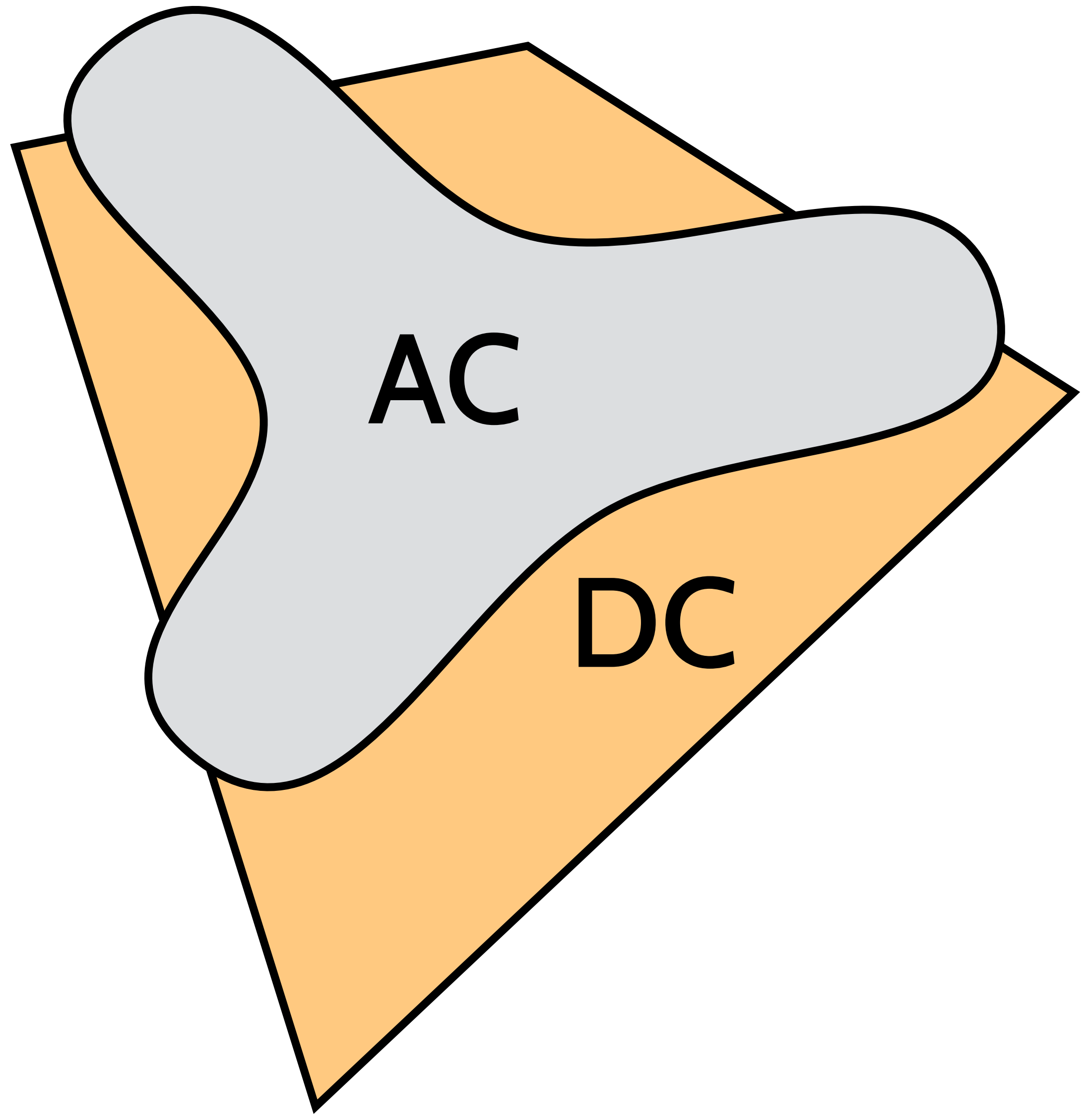
$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \;\; \forall i \in N$$

$$p_{ij} = -\boldsymbol{b}_{ij}(\theta_{ij}^{\Delta}) \;\; (i,j) \in E \cup E^R$$

← Linear Model

$$\theta_{ij}^{\Delta} = \theta_i - \theta_j \;\; \forall (i,j) \in E$$

$$|p_{ij}| \leq \boldsymbol{s}_{ij}^{\boldsymbol{u}} \;\; \forall (i,j) \in E \cup E^R$$



linear

Power Network Optimization is Complicated

SOC Power Flow Relaxation

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E \cup E^R} p_{ij} \quad \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E \cup E^R} q_{ij} \quad \forall i \in N$$

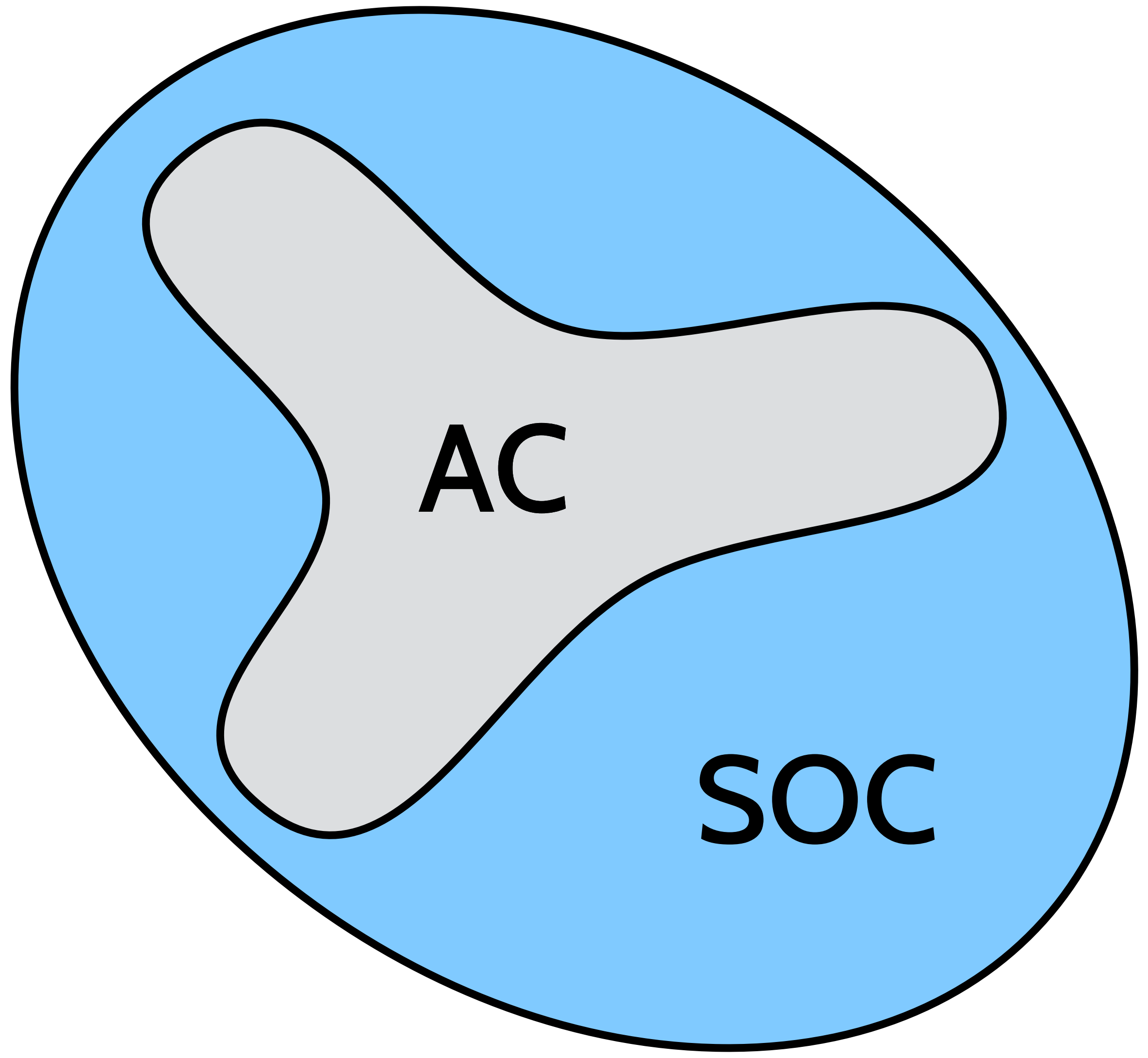
$$p_{ij} = \boldsymbol{g}_{ij} w_i - \boldsymbol{g}_{ij} w_{ij}^R - \boldsymbol{b}_{ij} w_{ij}^I \quad (i,j) \in E \cup E^R$$

$$q_{ij} = -\boldsymbol{b}_{ij} w_i + \boldsymbol{b}_{ij} w_{ij}^R - \boldsymbol{g}_{ij} w_{ij}^I \quad (i,j) \in E \cup E^R$$

$$p_{ij}^2 + q_{ij}^2 \le (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \quad \forall (i,j) \in E \cup E^R$$

$$(w_{ij}^R)^2 + (w_{ij}^I)^2 \le w_i w_j \quad (i,j) \in E$$

$$\boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta l}} w_{ij}^R \le w_{ij}^I \le w_{ij}^R \boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta u}} \quad (i,j) \in E$$



convex

UNCLASSIFIED

Power Network Optimization is Complicated

SOC Power Flow Relaxation

$$p_i^g - \boldsymbol{p}_i^d = \sum_{(i,j)\in E\cup E^R} p_{ij} \quad \forall i \in N$$

$$q_i^g - \boldsymbol{q}_i^d = \sum_{(i,j)\in E\cup E^R} q_{ij} \quad \forall i \in N$$

$$p_{ij} = \boldsymbol{g}_{ij} w_i - \boldsymbol{g}_{ij} w_{ij}^R - \boldsymbol{b}_{ij} w_{ij}^I \quad (i,j) \in E \cup E^R$$

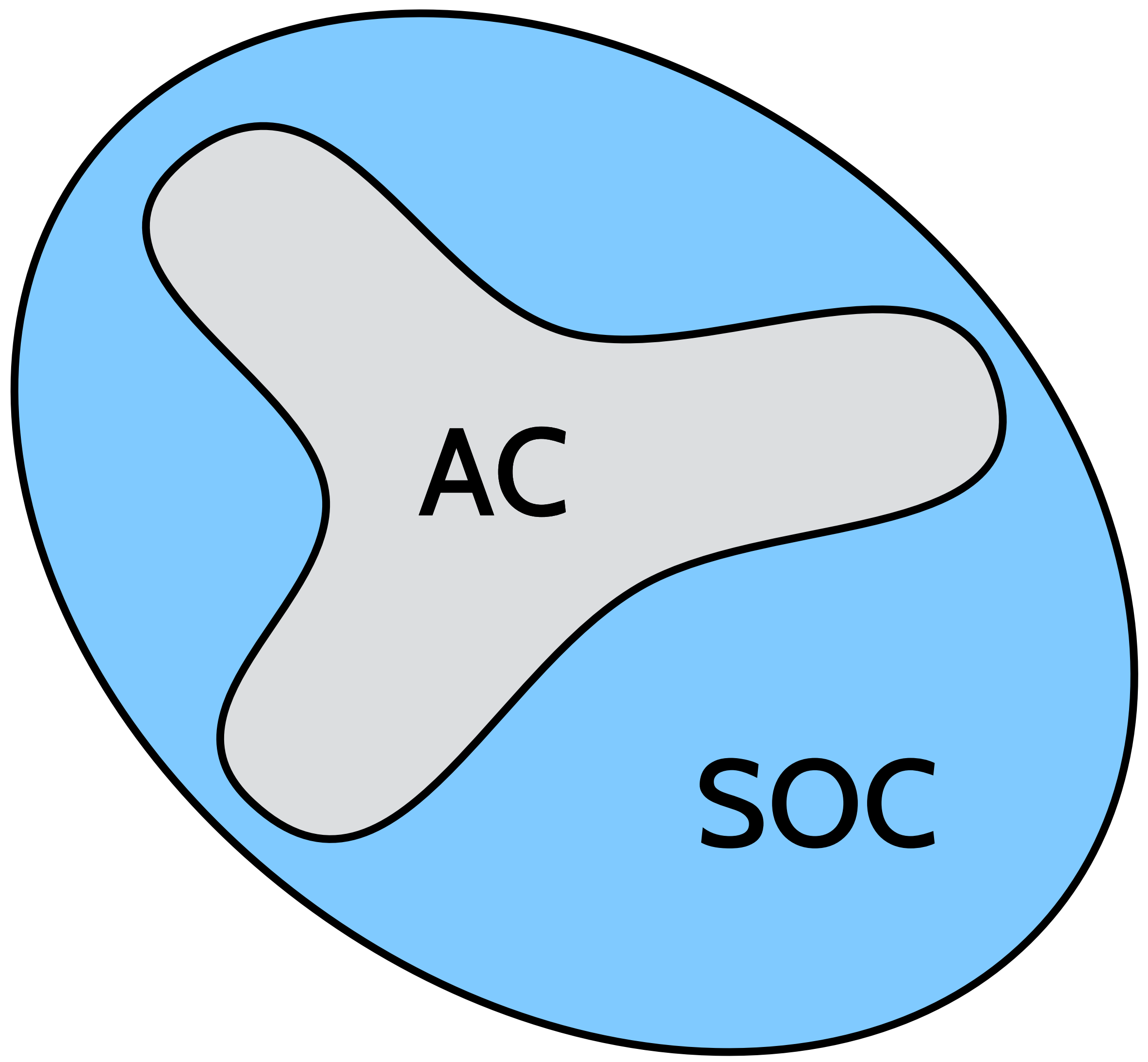
$$q_{ij} = -\boldsymbol{b}_{ij} w_i + \boldsymbol{b}_{ij} w_{ij}^R - \boldsymbol{g}_{ij} w_{ij}^I \quad (i,j) \in E \cup E^R$$

$$p_{ij}^2 + q_{ij}^2 \le (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \quad \forall (i,j) \in E \cup E^R$$

$$(w_{ij}^R)^2 + (w_{ij}^I)^2 \le w_i w_j \quad (i,j) \in E$$

$$\boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta l}} w_{ij}^R \le w_{ij}^I \le w_{ij}^R \boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta u}} \quad (i,j) \in E$$

Convex Constraints



convex

R&D Challenges

- Two Core Issues
  - Power Flow Formulations
  - Test Cases for Benchmarking

The Formulation Problem

- It is possible to publish a ***new*** **approximation or relaxation**, without comparing to many previous works

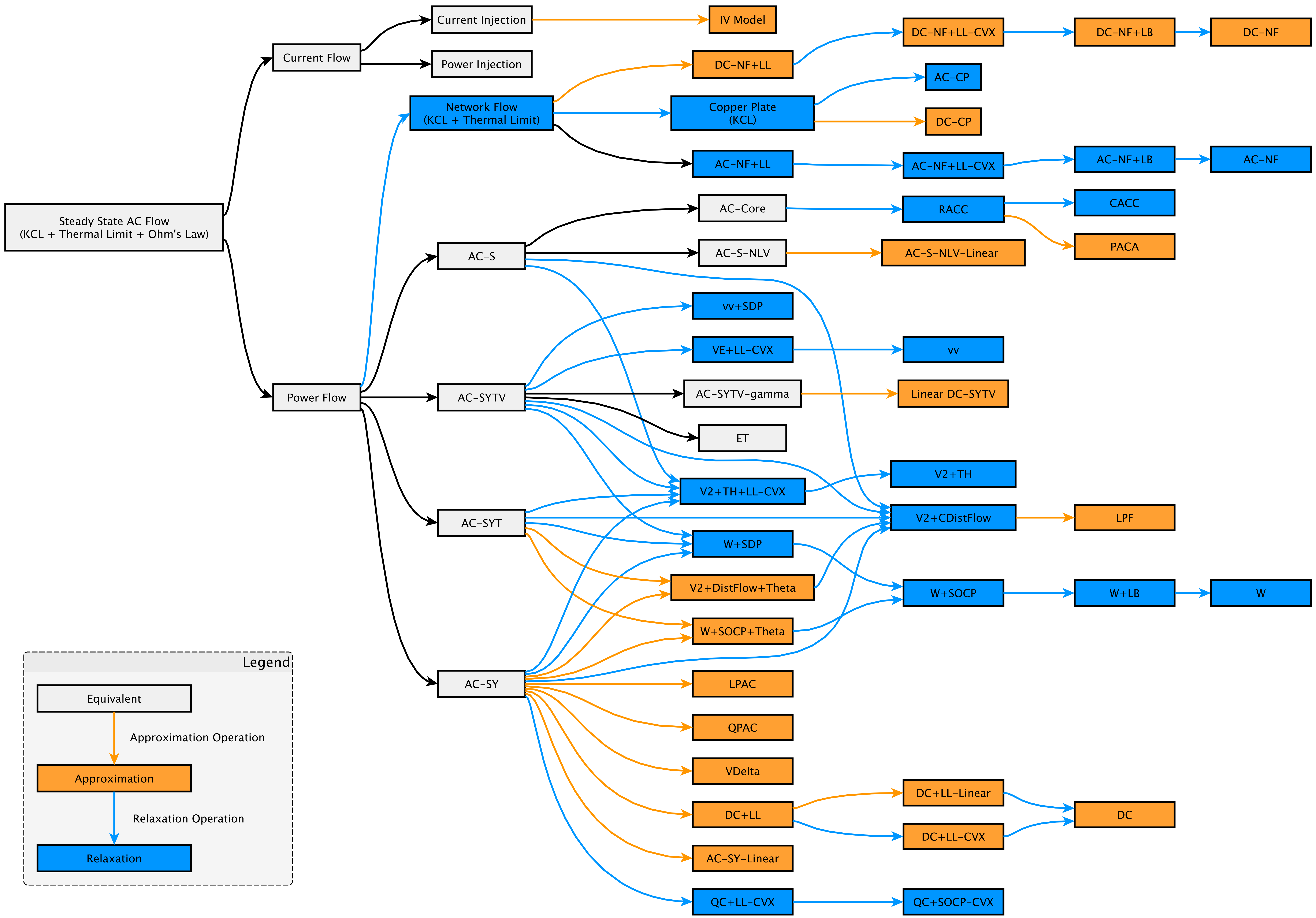
The Formulation Problem

- It is possible to publish a ***new*** **approximation or relaxation**, without comparing to many previous works
- There has been an **explosion** of proposed power flow alternatives (often hard to find)

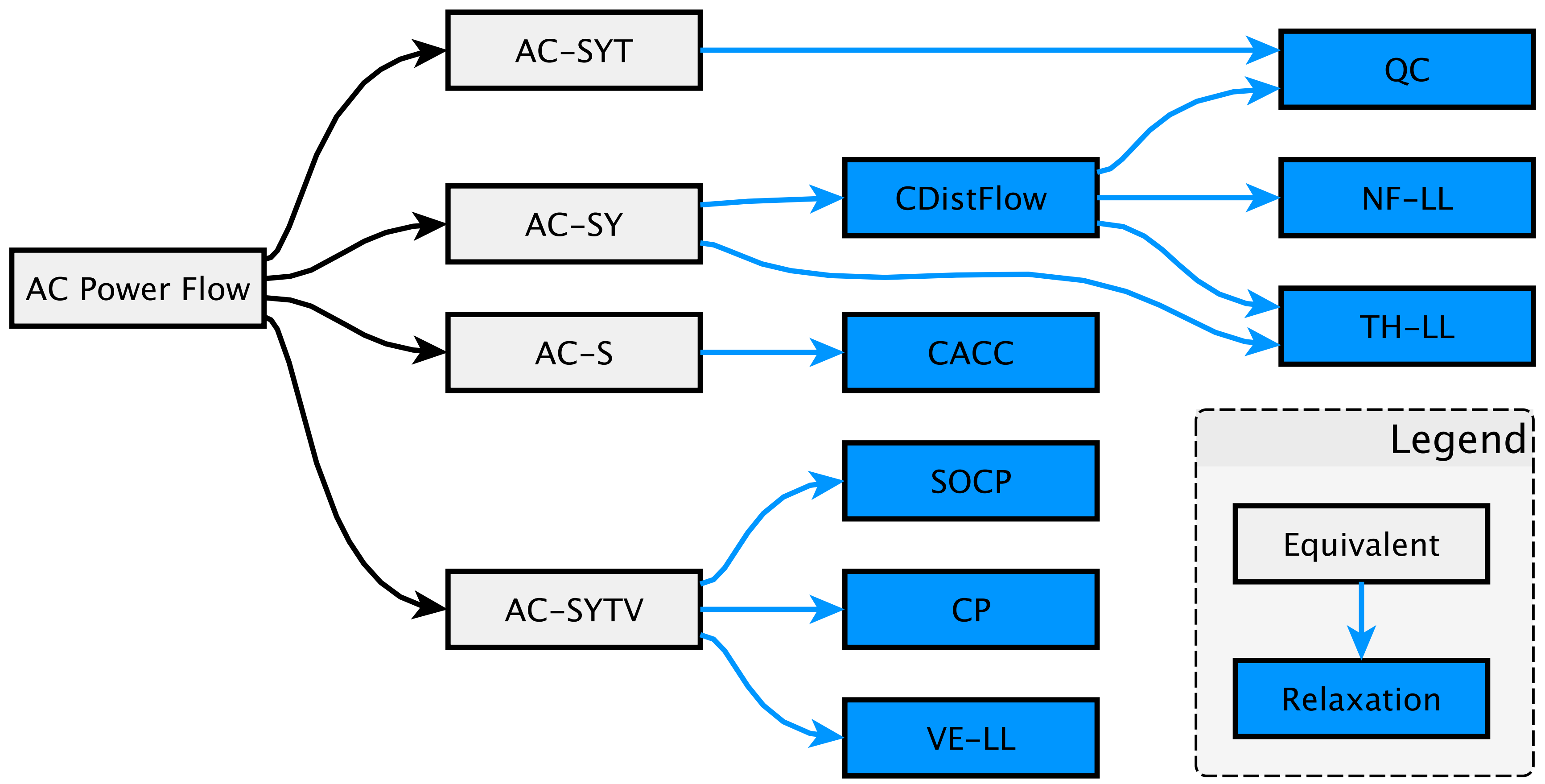
The Formulation Problem

- It is possible to publish a ***new*** **approximation or relaxation**, without comparing to many previous works
- There has been an **explosion** of proposed power flow alternatives (often hard to find)
- **No clear top performers**, in terms of citations at least…

Formulation Taxonomy (as of 2014)



Just Relaxations



The Instance Problem

- It is possible to publish a *new* method, by **only testing on a few (5-10)**

The Instance Problem

- It is possible to publish a *new* method, by **only testing on a few (5-10)**
- typically these are very-easy test cases
  - e.g. convex objective function with **no binding constraints**

The Instance Problem

- It is possible to publish a *new* method, by **only testing on a few (5-10)**
- typically these are very-easy test cases
  - e.g. convex objective function with **no binding constraints**
- **Industry** more-or-less **ignores** academic results
  - One reason is that the test cases are too easy

My Solution?

My Solution?

A novel scientific methodology

My Solution?

A novel scientific methodology

Brute-Force R&D

Run All Formulations on All Instances

"No clever ideas required!"

AMPL Implementation

- AC_b_only.mod
- AC_basic.mod
- AC_cb.mod
- AC_cb2.mod
- AC_cp.mod
- AC_current_inject.mod
- AC_current.mod
- AC_distflow_cvx.mod
- AC_first_order.mod
- AC_global_rect.mod
- AC_global_w_rect.mod
- AC_global_w.mod
- AC_global.mod
- AC_line_flex.mod
- AC_line_fp.mod
- AC_line_fp2.mod
- AC_ll_theta.mod
- AC_loss.mod
- AC_nf_lb_lin.mod
- AC_nf_lin.mod
- AC_nf_ll_cvx.mod
- AC_nf_ll.mod
- AC_pf_opf.mod
- AC_pf_soft.mod
- AC_pf.mod
- AC_polar.mod
- AC_poly_cvx.mod
- AC_poly_ll_cvx.mod
- AC_rect_cvx.mod
- AC_rect_gamma_lin.mod
- AC_rect_gamma.mod
- AC_rect_ll_cvx.mod
- AC_rect_nf_cvx.mod
- AC_rect_polar.mod
- AC_rect.mod
- DC_cp.mod
- DC_distflow_cvx.mod
- DC_ll_cvx.mod
- DC_ll.mod
- DC_nf_ll_cvx.mod
- DC_nf.mod
- DC.mod
- QC_bus_flex.mod
- QC_cs_cvx.mod
- QC_cut_cvx_fp.mod
- QC_cut_flex_nlp.mod
- QC_cvx_fp_qp.mod
- QC_cvx_fp.mod
- QC_cvx_init.mod
- QC_cvx_sym.mod
- QC_cvx.mod
- QC_dir_cvx.mod
- QC_flex_cvx_pre.mod
- QC_flex_cvx.mod
- QC_flex_nlp_pre.mod
- QC_flex_nlp.mod
- QC_line_flex_nlp.mod
- QC_line_flex.mod
- QC_line_fp2_nlp.mod
- QC_line_fp2.mod
- QC_ncvx.mod
- QC_nlp_old.mod
- QC_nlp.mod
- QC_tan_cvx.mod
- QC_w_cvx.mod
- QPAC.mod
- SOC_cut_flex_cvx.mod
- SOC_cvx_fp.mod
- SOC_flex_cvx.mod
- SOC_w_cs.mod
- SOC_w_cvx_er.mod
- SOC_w_cvx_er2.mod
- SOC_w_cvx_lp.mod
- SOC_w_cvx.mod
- SOC_w_cyc3.mod
- SOC_w_lnc_cvx.mod
- SOC_w_sdp3.mod
- SOC_w_tan_cvx.mod
- SOC_w_theta.mod
- SOC_w.mod
- SOC_wl_cvx.mod

Test Case Archive

NESTA
The NICTA Energy System Test Case Archive

Carleton Coffrin[1,2,3], Dan Gordon[1], and Paul Scott[1,2]

[1]Optimisation Research Group, NICTA
[2]College of Engineering and Computer Science, Australian National University
[3]Computing and Information Systems, University of Melbourne

August 12, 2016

Abstract

In recent years the power systems research community has seen an explosion of work applying operations research techniques to challenging power network optimization problems. Regardless of the application under consideration, all of these works rely on power system test cases for evaluation and validation. However, many of the well established power system test cases were developed as far back as the 1960s with the aim of testing AC power flow algorithms. It is unclear if these power flow test cases are suitable for power system optimization studies. This report surveys all of the publicly available AC transmission system test cases, to the best of our knowledge, and assess their suitability for optimization tasks. It finds that many of the traditional test cases are missing key network operation constraints, such as line thermal limits and generator capability curves. To incorporate these missing constraints, data driven models are developed from a variety of publicly available data sources. The resulting extended test cases form a compressive archive, NESTA, for the evaluation and validation of power system optimization algorithms.

https://arxiv.org/abs/1411.0359

Test Case Archive

NESTA

The NICTA Energy System Test Case Archive

Carleton Coffrin[1,2,3], Dan Gordon[1], and Paul Scott[1,2]

[1]Optimisation Research Group, NICTA
[2]College of Engineering and Computer Science, Australian National University
[3]Computing and Information Systems, University of Melbourne

August 12, 2016

Abstract

In recent years the power systems research community has seen an explosion of work applying operations research techniques to challenging power network optimization problems. Regardless of the application under consideration, all of these works rely on power system test cases for evaluation and validation. However, many of the well established power system test cases were developed as far back as the 1960s with the aim of testing AC power flow algorithms. It is unclear if these power flow test cases are suitable for power system optimization studies. This report surveys all of the publicly available AC transmission system test cases, to the best of our knowledge, and assess their suitability for optimization tasks. It finds that many of the traditional test cases are missing key network operation constraints, such as line thermal limits and generator capability curves. To incorporate these missing constraints, data driven models are developed from a variety of publicly available data sources. The resulting extended test cases form a compressive archive, NESTA, for the evaluation and validation of power system optimization algorithms.

https://arxiv.org/abs/1411.0359

35 base cases	API	SAD
nesta_case3_lmbd	nesta_case3_lmbd__api	nesta_case3_lmbd__sad
nesta_case4_gs	nesta_case4_gs__api	nesta_case4_gs__sad
nesta_case5_pjm	nesta_case5_pjm__api	nesta_case5_pjm__sad
nesta_case6_c	nesta_case6_c__api	nesta_case6_c__sad
nesta_case6_ww	nesta_case6_ww__api	nesta_case6_ww__sad
nesta_case9_wscc	nesta_case9_wscc__api	nesta_case9_wscc__sad
nesta_case14_ieee	nesta_case14_ieee__api	nesta_case14_ieee__sad
nesta_case24_ieee_rts	nesta_case24_ieee_rts__api	nesta_case24_ieee_rts__sad
nesta_case29_edin	nesta_case29_edin__api	nesta_case29_edin__sad
nesta_case30_as	nesta_case30_as__api	nesta_case30_as__sad
nesta_case30_fsr	nesta_case30_fsr__api	nesta_case30_fsr__sad
nesta_case30_ieee	nesta_case30_ieee__api	nesta_case30_ieee__sad
nesta_case39_epri	nesta_case39_epri__api	nesta_case39_epri__sad
nesta_case57_ieee	nesta_case57_ieee__api	nesta_case57_ieee__sad
nesta_case73_ieee_rts	nesta_case73_ieee_rts__api	nesta_case73_ieee_rts__sad
nesta_case89_pegase	nesta_case89_pegase__api	nesta_case89_pegase__sad
nesta_case118_ieee	nesta_case118_ieee__api	nesta_case118_ieee__sad
nesta_case162_ieee_dtc	nesta_case162_ieee_dtc__api	nesta_case162_ieee_dtc__sad
nesta_case189_edin	nesta_case189_edin__api	nesta_case189_edin__sad
nesta_case300_ieee	nesta_case300_ieee__api	nesta_case300_ieee__sad
nesta_case1354_pegase	nesta_case1354_pegase__api	nesta_case1354_pegase__sad
nesta_case1394sop_eir	nesta_case1394sop_eir__api	nesta_case1394sop_eir__sad
nesta_case1397sp_eir	nesta_case1397sp_eir__api	nesta_case1397sp_eir__sad
nesta_case1460wp_eir	nesta_case1460wp_eir__api	nesta_case1460wp_eir__sad
nesta_case2224_edin	nesta_case2224_edin__api	nesta_case2224_edin__sad
nesta_case2383wp_mp	nesta_case2383wp_mp__api	nesta_case2383wp_mp__sad
nesta_case2736sp_mp	nesta_case2736sp_mp__api	nesta_case2736sp_mp__sad
nesta_case2737sop_mp	nesta_case2737sop_mp__api	nesta_case2737sop_mp__sad
nesta_case2746wp_mp	nesta_case2746wp_mp__api	nesta_case2746wp_mp__sad
nesta_case2746wop_mp	nesta_case2746wop_mp__api	nesta_case2746wop_mp__sad
nesta_case2869_pegase	nesta_case2869_pegase__api	nesta_case2869_pegase__sad
nesta_case3012wp_mp	nesta_case3012wp_mp__api	nesta_case3012wp_mp__sad
nesta_case3120sp_mp	nesta_case3120sp_mp__api	nesta_case3120sp_mp__sad
nesta_case3375wp_mp	nesta_case3375wp_mp__api	nesta_case3375wp_mp__sad
nesta_case9241_pegase	nesta_case9241_pegase__api	nesta_case9241_pegase__sad

Brute Force R&D Example

The QC Relaxation: Theoretical and Computational Results on Optimal Power Flow

https://arxiv.org/abs/1502.07847

Brute Force R&D Example

The QC Relaxation: Theoretical and Computational Results on Optimal Power Flow

https://arxiv.org/abs/1502.07847

Power Formulations

Non-Trivial Instances

TABLE III

QUALITY AND RUNTIME RESULTS OF AC POWER FLOW RELAXATIONS

Test Case	\$/h AC	Optimality Gap (%) SDP	QC	SOC	CP	Runtime (seconds) AC	SDP	QC	SOC	CP
Typical Operating Conditions (TYP)										
nesta_case3_lmbd	5812.64	0.39	1.24	1.32	2.99	0.12	4.16	0.07	0.05	0.03
nesta_case5_pjm	17551.89	5.22	14.54	14.54	15.62	0.04	5.36	0.09	0.03	0.05
nesta_case30_ieee	204.97	**0.00**	15.64	15.88	27.91	0.09	8.38	0.17	0.07	0.06
nesta_case118_ieee	3718.64	0.06	1.72	2.07	7.87	0.41	12.62	0.87	0.43	0.05
nesta_case162_ieee_dtc	4230.23	1.08	4.00	4.03	15.44	0.61	35.20	1.48	0.31	0.04
nesta_case300_ieee	16891.28	0.08	1.17	1.18	n.a.	0.80	29.69	2.83	0.65	n.a.
nesta_case2224_edin	38127.69	1.22	6.03	6.09	8.45	11.42	690.16	65.59	45.99	0.33
nesta_case2383wp_mp	1868511.78	0.37	1.04	1.05	5.35	12.41	1966.10	57.87	12.91	0.80
nesta_case3012wp_mp	2600842.72	—	1.00	1.02	n.a.	12.40	14588.79†	53.59	19.15	n.a.
nesta_case9241_pegase	315913.26	—	1.67	—	n.a.	132.25	—	3064.42	—	n.a.
Congested Operating Conditions (API)										
nesta_case3_lmbd__api	367.74	1.26	1.83	3.30	14.79	0.18	4.41	0.09	0.05	0.23
nesta_case6_ww__api	273.76	0.00*	13.14	13.33	17.17	0.34	13.19	0.07	0.06	0.03
nesta_case14_ieee__api	325.56	**0.00**	1.34	1.34	8.89	0.19	5.64	0.11	0.08	0.94
nesta_case24_ieee_rts__api	6421.37	1.45	13.77	20.70	24.12	0.14	7.50	0.26	0.09	0.04
nesta_case30_as__api	571.13	0.00	4.76	4.76	8.01	0.38	6.12	0.17	0.11	1.11
nesta_case30_fsr__api	372.14	11.06	45.97	45.97	48.80	0.25	7.25	0.19	0.09	0.92
nesta_case30_ieee__api	415.53	0.00	1.01	1.01	12.75	0.07	6.60	0.19	0.09	0.03
nesta_case39_epri__api	7466.25	**0.00**	2.97	2.99	13.31	0.10	7.36	0.29	0.12	0.04
nesta_case73_ieee_rts__api	20123.98	4.29	12.01	14.34	17.83	0.48	10.03	0.66	0.20	0.06
nesta_case89_pegase__api	4288.02	18.11	20.39	20.43	22.60	1.16	21.58	1.29	0.81	0.04
nesta_case118_ieee__api	10325.27	31.50	43.93	44.08	49.69	0.46	12.59	0.84	0.25	0.05
nesta_case162_ieee_dtc__api	6111.68	0.85	1.33	1.34	19.39	0.50	36.85	1.53	0.39	0.05
nesta_case189_edin__api	1982.82	0.05	5.78	5.78	n.a.	1.07	16.10	1.14	0.33	n.a.
nesta_case2224_edin__api	46235.43	1.10	2.77	2.77	9.07	12.28	672.04	81.66	88.33	0.33
nesta_case2383wp_mp__api	23499.48	0.10	1.12	1.12	3.10	9.50	1421.39	28.37	10.25	0.34
nesta_case2736sp_mp__api	25437.70	0.07	1.32	1.33	3.89	9.21	2278.77	41.29	10.51	0.36
nesta_case2737sop_mp__api	21192.40	0.00	1.05	1.06	4.62	9.29	1887.22	30.94	9.91	0.32
nesta_case2869_pegase__api	96573.10	0.92*	1.49	1.49	5.16	21.03	1579.87	102.55	161.96	0.37
nesta_case3120sp_mp__api	22874.98	—	3.02	3.03	n.a.	14.92	15018.93†	41.72	12.19	n.a.
nesta_case9241_pegase__api	241975.18	—	2.45	2.59	n.a.	140.73	—	3511.60	8387.11	n.a.
Small Angle Difference Conditions (SAD)										
nesta_case3_lmbd__sad	5992.72	2.06	1.24*	4.28	5.90	0.19	4.39	0.10	0.05	0.03
nesta_case4_gs__sad	324.02	0.05	0.81	4.90	66.06	0.24	4.16	0.06	0.06	0.07
nesta_case5_pjm__sad	26423.32	**0.00**	1.10	3.61	43.95	0.08	5.35	0.11	0.05	0.03
nesta_case6_c__sad	24.43	0.00	0.40	1.36	6.79	0.26	5.32	0.11	0.05	0.02
nesta_case9_wscc__sad	5590.09	0.00	0.41	1.50	6.69	0.14	4.18	0.19	0.05	0.03
nesta_case24_ieee_rts__sad	79804.96	6.05	3.88	11.42	23.56	0.10	6.24	0.30	0.11	0.04
nesta_case29_edin__sad	46933.26	28.44	20.57	34.47	36.79	0.70	9.19	1.73	0.27	0.06
nesta_case30_as__sad	914.44	0.47	3.07	9.16	16.06	0.18	6.49	0.22	0.09	0.03
nesta_case30_ieee__sad	205.11	**0.00**	3.96	5.84	27.96	0.12	7.49	0.18	0.09	0.03
nesta_case73_ieee_rts__sad	235241.70	4.10	3.51	8.37	22.21	0.30	9.48	0.87	0.20	0.07
nesta_case118_ieee__sad	4324.17	7.57	8.32	12.89	20.77	0.56	14.14	0.98	0.31	0.06
nesta_case162_ieee_dtc__sad	4369.19	3.65	6.91	7.08	18.13	0.81	39.71	1.70	0.36	0.05
nesta_case189_edin__sad	914.61	1.20*	2.22	2.25	n.a.	0.65	14.83	1.27	0.46	n.a.
nesta_case300_ieee__sad	16910.23	0.13	1.16	1.26	n.a.	1.01	29.63	2.81	0.76	n.a.
nesta_case2224_edin__sad	38385.14	1.22	5.57	6.18	9.06	11.53	691.53	50.34	65.68	0.33
nesta_case2383wp_mp__sad	1935308.12	1.30	2.97	4.00	8.62	16.25	1785.26	40.71	12.57	0.80
nesta_case2736sp_mp__sad	1337042.77	2.18*	2.01	2.34	4.56	13.22	1737.25	35.42	11.31	0.48
nesta_case2737sop_mp__sad	795429.36	2.24*	2.21	2.42	3.95	13.01	2153.37	32.05	9.69	0.39
nesta_case2746wp_mp__sad	1672150.46	2.41*	1.83	2.44	5.43	14.01	2840.32	35.66	13.32	0.56
nesta_case2746wop_mp__sad	1241955.30	2.71*	2.48	2.94	5.14	14.51	2306.18	32.41	23.22	0.42
nesta_case3012wp_mp__sad	2635451.29	—	1.92	2.12	n.a.	15.79	13548.13†	46.59	28.41	n.a.
nesta_case3120sp_mp__sad	2203807.23	—	2.56	2.79	n.a.	30.01	16804.55†	53.81	15.69	n.a.
nesta_case9241_pegase__sad	315932.06	—	0.80	1.75	n.a.	80.30	—	3531.62	33437.86	n.a.

bold - the relaxation provided a feasible AC power flow, $\star$ - solver reported numerical accuracy warnings, —,$\dagger$ - iteration or memory limit

Brute Force R&D Example

The QC Relaxation: Theoretical and Computational Results on Optimal Power Flow

https://arxiv.org/abs/1502.07847

Power Formulations

Non-Trivial Instances

TABLE III
QUALITY AND RUNTIME RESULTS OF AC POWER FLOW RELAXATIONS

Test Case	$/h AC	Optimality Gap (%) SDP	QC	SOC	CP	Runtime (seconds) AC	SDP	QC	SOC	CP
Typical Operating Conditions (TYP)										
nesta_case3_lmbd	5812.64	0.39	1.24	1.32	2.99	0.12	4.16	0.07	0.05	0.03
nesta_case5_pjm	17551.89	5.22	14.54	14.54	15.62	0.04	5.36	0.09	0.03	0.05
nesta_case30_ieee	204.97	**0.00**	15.64	15.88	27.91	0.09	8.38	0.17	0.07	0.06
nesta_case118_ieee	3718.64	0.06	1.72	2.07	7.87	0.41	12.62	0.87	0.43	0.05
nesta_case162_ieee_dtc	4230.23	1.08	4.00	4.03	15.44	0.61	35.20	1.48	0.31	0.04
nesta_case300_ieee	16891.28	0.08	1.17	1.18	n.a.	0.80	29.69	2.83	0.65	n.a.
nesta_case2224_edin	38127.69	1.22	6.03	6.09	8.45	11.42	690.16	65.59	45.99	0.33
nesta_case2383wp_mp	1868511.78	0.37	1.04	1.05	5.35	12.41	1966.10	57.87	12.91	0.80
nesta_case3012wp_mp	2600842.72	—	1.00	1.02	n.a.	12.40	14588.79†	53.59	19.15	n.a.
nesta_case9241_pegase	315913.26	—	1.67	—	n.a.	132.25	—	3064.42	—	n.a.
Congested Operating Conditions (API)										
nesta_case3_lmbd__api	367.74	1.26	1.83	3.30	14.79	0.18	4.41	0.09	0.05	0.23
nesta_case6_ww__api	273.76	0.00⋆	13.14	13.33	17.17	0.34	13.19	0.07	0.06	0.03
nesta_case14_ieee__api	325.56	**0.00**	1.34	1.34	8.89	0.19	5.64	0.11	0.08	0.94
nesta_case24_ieee_rts__api	6421.37	1.45	13.77	20.70	24.12	0.14	7.50	0.26	0.09	0.04
nesta_case30_as__api	571.13	0.00	4.76	4.76	8.01	0.38	6.12	0.17	0.11	1.11
nesta_case30_fsr__api	372.14	11.06	45.97	45.97	48.80	0.25	7.25	0.19	0.09	0.92
nesta_case30_ieee__api	415.53	0.00	1.01	1.01	12.75	0.07	6.60	0.19	0.09	0.03
nesta_case39_epri__api	7466.25	**0.00**	2.97	2.99	13.31	0.10	7.36	0.29	0.12	0.04
nesta_case73_ieee_rts__api	20123.98	4.29	12.01	14.34	17.83	0.48	10.03	0.66	0.20	0.06
nesta_case89_pegase__api	4288.02	18.11	20.39	20.43	22.60	1.16	21.58	1.29	0.81	0.04
nesta_case118_ieee__api	10325.27	31.50	43.93	44.08	49.69	0.46	12.59	0.84	0.25	0.05
nesta_case162_ieee_dtc__api	6111.68	0.85	1.33	1.34	19.39	0.50	36.85	1.53	0.39	0.05
nesta_case189_edin__api	1982.82	0.05	5.78	5.78	n.a.	1.07	16.10	1.14	0.33	n.a.
nesta_case2224_edin__api	46235.43	1.10	2.77	2.77	9.07	12.28	672.04	81.66	88.33	0.33
nesta_case2383wp_mp__api	23499.48	0.10	1.12	1.12	3.10	9.50	1421.39	28.37	10.25	0.34
nesta_case2736sp_mp__api	25437.70	0.07	1.32	1.33	3.89	9.21	2278.77	41.29	10.51	0.36
nesta_case2737sop_mp__api	21192.40	0.00	1.05	1.06	4.62	9.29	1887.22	30.94	9.91	0.32
nesta_case2869_pegase__api	96573.10	0.92⋆	1.49	1.49	5.16	21.03	1579.87	102.55	161.96	0.37
nesta_case3120sp_mp__api	22874.98	—	3.02	3.03	n.a.	14.92	15018.93†	41.72	12.19	n.a.
nesta_case9241_pegase__api	241975.18	—	2.45	2.59	n.a.	140.73	—	3511.60	8387.11	n.a.
Small Angle Difference Conditions (SAD)										
nesta_case3_lmbd__sad	5992.72	2.06	1.24⋆	4.28	5.90	0.19	4.39	0.10	0.05	0.03
nesta_case4_gs__sad	324.02	0.05	0.81	4.90	66.06	0.24	4.16	0.06	0.06	0.07
nesta_case5_pjm__sad	26423.32	**0.00**	1.10	3.61	43.95	0.08	5.35	0.11	0.05	0.03
nesta_case6_c__sad	24.43	0.00	0.40	1.36	6.79	0.26	5.32	0.11	0.05	0.02
nesta_case9_wscc__sad	5590.09	0.00	0.41	1.50	6.69	0.14	4.18	0.19	0.05	0.03
nesta_case24_ieee_rts__sad	79804.96	6.05	3.88	11.42	23.56	0.10	6.24	0.30	0.11	0.04
nesta_case29_edin__sad	46933.26	28.44	20.57	34.47	36.79	0.70	9.19	1.73	0.27	0.06
nesta_case30_as__sad	914.44	0.47	3.07	9.16	16.06	0.18	6.49	0.22	0.09	0.03
nesta_case30_ieee__sad	205.11	**0.00**	3.96	5.84	27.96	0.12	7.49	0.18	0.09	0.03
nesta_case73_ieee_rts__sad	235241.70	4.10	3.51	8.37	22.21	0.30	9.48	0.87	0.20	0.07
nesta_case118_ieee__sad	4324.17	7.57	8.32	12.89	20.77	0.56	14.14	0.98	0.31	0.06
nesta_case162_ieee_dtc__sad	4369.19	3.65	6.91	7.08	18.13	0.81	39.71	1.70	0.36	0.05
nesta_case189_edin__sad	914.61	1.20⋆	2.22	2.25	n.a.	0.65	14.83	1.27	0.46	n.a.
nesta_case300_ieee__sad	16910.23	0.13	1.16	1.26	n.a.	1.01	29.63	2.81	0.76	n.a.
nesta_case2224_edin__sad	38385.14	1.22	5.57	6.18	9.06	11.53	691.53	50.34	65.68	0.33
nesta_case2383wp_mp__sad	1935308.12	1.30	2.97	4.00	8.62	16.25	1785.26	40.71	12.57	0.80
nesta_case2736sp_mp__sad	1337042.77	2.18⋆	2.01	2.34	4.56	13.22	1737.25	35.42	11.31	0.48
nesta_case2737sop_mp__sad	795429.36	2.24⋆	2.21	2.42	3.95	13.01	2153.37	32.05	9.69	0.39
nesta_case2746wp_mp__sad	1672150.46	2.41⋆	1.83	2.44	5.43	14.01	2840.32	35.66	13.32	0.56
nesta_case2746wop_mp__sad	1241955.30	2.71⋆	2.48	2.94	5.14	14.51	2306.18	32.41	23.22	0.42
nesta_case3012wp_mp__sad	2635451.29	—	1.92	2.12	n.a.	15.79	13548.13†	46.59	28.41	n.a.
nesta_case3120sp_mp__sad	2203807.23	—	2.56	2.79	n.a.	30.01	16804.55†	53.81	15.69	n.a.
nesta_case9241_pegase__sad	315932.06	—	0.80	1.75	n.a.	80.30	—	3531.62	33437.86	n.a.

bold - the relaxation provided a feasible AC power flow, ⋆ - solver reported numerical accuracy warnings, —,† - iteration or memory limit

Unexpected Insights!

UNCLASSIFIED

Brute Force R&D Lessons Learned

- **Reproducing previous works is challenging**
  - working from a base implementation is very helpful

Brute Force R&D Lessons Learned

- **Reproducing previous works is challenging**
  - working from a base implementation is very helpful
- **AMPL was not built for this…**
  - limited means to avoid excessive code replication
  - really hard to automate from the command line
  - limited licenses was the bottle neck in the All Formulations by All Instances Experiment

The Matpower Effect

- If a formulation is not implemented in Matpower, it **does not exist**
  - At least for the majority of Power System PhD students

Inception of PowerModels.jl

- A **baseline implementation** of Power Flow formulations from the literature
  - Hopefully, mitigates the Matpower effect

Inception of PowerModels.jl

- A **baseline implementation** of Power Flow formulations from the literature
  - Hopefully, mitigates the Matpower effect
- **Using Julia/JuMP** Resolves the AMPL Issues
  - Easy to automate at the command line
  - Fully open-source makes large-scale experiments easy
  - Julia enables advanced software design

My Dream

- I learn about a newly **proposed Power Flow** formulation
- It is **implemented** in PowerModels.jl and tested on all started test cases, **in 7 days or less**

My Dream

- I learn about a newly **proposed Power Flow** formulation
- It is **implemented** in PowerModels.jl and tested on all started test cases, **in 7 days or less**
- Lots of **code abstractions** in PowerModels.jl to enable this

The Value of Open-Source

TABLE III
QUALITY AND RUNTIME RESULTS OF AC POWER FLOW RELAXATIONS

	$/h	Optimality Gap (%)				Runtime (seconds)				
Test Case	AC	SDP	QC	SOC	CP	AC	SDP	QC	SOC	CP
Typical Operating Conditions (TYP)										
nesta_case3_lmbd	5812.64	0.39	1.24	1.32	2.99	0.12	4.16	0.07	0.05	0.03
nesta_case5_pjm	17551.89	5.22	14.54	14.54	15.62	0.04	5.36	0.09	0.03	0.05
nesta_case30_ieee	204.97	**0.00**	15.64	15.88	27.91	0.09	8.38	0.17	0.07	0.06
nesta_case118_ieee	3718.64	0.06	1.72	2.07	7.87	0.41	12.62	0.87	0.43	0.05
nesta_case162_ieee_dtc	4230.23	1.08	4.00	4.03	15.44	0.61	35.20	1.48	0.31	0.04
nesta_case300_ieee	16891.28	0.08	1.17	1.18	n.a.	0.80	29.69	2.83	0.65	n.a.
nesta_case2224_edin	38127.69	1.22	6.03	6.09	8.45	11.42	690.16	65.59	45.99	0.33
nesta_case2383wp_mp	1868511.78	0.37	1.04	1.05	5.35	12.41	1966.10	57.87	12.91	0.80
nesta_case3012wp_mp	2600842.72	—	1.00	1.02	n.a.	12.40	14588.79†	53.59	19.15	n.a.
nesta_case9241_pegase	315913.26	—	1.67	—	n.a.	132.25	—	3064.42	—	n.a.
Congested Operating Conditions (API)										
nesta_case3_lmbd__api	367.74	1.26	1.83	3.30	14.79	0.18	4.41	0.09	0.05	0.23
nesta_case6_ww__api	273.76	0.00⋆	13.14	13.33	17.17	0.34	13.19	0.07	0.06	0.03
nesta_case14_ieee__api	325.56	**0.00**	1.34	1.34	8.89	0.19	5.64	0.11	0.08	0.94
nesta_case24_ieee_rts__api	6421.37	1.45	13.77	20.70	24.12	0.14	7.50	0.26	0.09	0.04
nesta_case30_as__api	571.13	0.00	4.76	4.76	8.01	0.38	6.12	0.17	0.11	1.11
nesta_case30_fsr__api	372.14	11.06	45.97	45.97	48.80	0.25	7.25	0.19	0.09	0.92
nesta_case30_ieee__api	415.53	0.00	1.01	1.01	12.75	0.07	6.60	0.19	0.09	0.03
nesta_case39_epri__api	7466.25	**0.00**	2.97	2.99	13.31	0.10	7.36	0.29	0.12	0.04
nesta_case73_ieee_rts__api	20123.98	4.29	12.01	14.34	17.83	0.48	10.03	0.66	0.20	0.06
nesta_case89_pegase__api	4288.02	18.11	20.39	20.43	22.60	1.16	21.58	1.29	0.81	0.04
nesta_case118_ieee__api	10325.27	31.50	43.93	44.08	49.69	0.46	12.59	0.84	0.25	0.05
nesta_case162_ieee_dtc__api	6111.68	0.85	1.33	1.34	19.39	0.50	36.85	1.53	0.39	0.05
nesta_case189_edin__api	1982.82	0.05	5.78	5.78	n.a.	1.07	16.10	1.14	0.33	n.a.
nesta_case2224_edin__api	46235.43	1.10	2.77	2.77	9.07	12.28	672.04	81.66	88.33	0.33
nesta_case2383wp_mp__api	23499.48	0.10	1.12	1.12	3.10	9.50	1421.39	28.37	10.25	0.34
nesta_case2736sp_mp__api	25437.70	0.07	1.32	1.33	3.89	9.21	2278.77	41.29	10.51	0.36
nesta_case2737sop_mp__api	21192.40	0.00	1.05	1.06	4.62	9.29	1887.22	30.94	9.91	0.32
nesta_case2869_pegase__api	96573.10	0.92⋆	1.49	1.49	5.16	21.03	1579.87	102.55	161.96	0.37
nesta_case3120sp_mp__api	22874.98	—	3.02	3.03	n.a.	14.92	15018.93†	41.72	12.19	n.a.
nesta_case9241_pegase__api	241975.18	—	2.45	2.59	n.a.	140.73	—	3511.60	8387.11	n.a.
Small Angle Difference Conditions (SAD)										
nesta_case3_lmbd__sad	5992.72	2.06	1.24⋆	4.28	5.90	0.19	4.39	0.10	0.05	0.03
nesta_case4_gs__sad	324.02	0.05	0.81	4.90	66.06	0.24	4.16	0.06	0.06	0.07
nesta_case5_pjm__sad	26423.32	**0.00**	1.10	3.61	43.95	0.08	5.35	0.11	0.05	0.03
nesta_case6_c__sad	24.43	0.00	0.40	1.36	6.79	0.26	5.32	0.11	0.05	0.02
nesta_case9_wscc__sad	5590.09	0.00	0.41	1.50	6.69	0.14	4.18	0.19	0.05	0.03
nesta_case24_ieee_rts__sad	79804.96	6.05	3.88	11.42	23.56	0.10	6.24	0.30	0.11	0.04
nesta_case29_edin__sad	46933.26	28.44	20.57	34.47	36.79	0.70	9.19	1.73	0.27	0.06
nesta_case30_as__sad	914.44	0.47	3.07	9.16	16.06	0.18	6.49	0.22	0.09	0.03
nesta_case30_ieee__sad	205.11	**0.00**	3.96	5.84	27.96	0.12	7.49	0.18	0.09	0.03
nesta_case73_ieee_rts__sad	235241.70	4.10	3.51	8.37	22.21	0.30	9.48	0.87	0.20	0.07
nesta_case118_ieee__sad	4324.17	7.57	8.32	12.89	20.77	0.56	14.14	0.98	0.31	0.06
nesta_case162_ieee_dtc__sad	4369.19	3.65	6.91	7.08	18.13	0.81	39.71	1.70	0.36	0.05
nesta_case189_edin__sad	914.61	1.20⋆	2.22	2.25	n.a.	0.65	14.83	1.27	0.46	n.a.
nesta_case300_ieee__sad	16910.23	0.13	1.16	1.26	n.a.	1.01	29.63	2.81	0.76	n.a.
nesta_case2224_edin__sad	38385.14	1.22	5.57	6.18	9.06	11.53	691.53	50.34	65.68	0.33
nesta_case2383wp_mp__sad	1935308.12	1.30	2.97	4.00	8.62	16.25	1785.26	40.71	12.57	0.80
nesta_case2736sp_mp__sad	1337042.77	2.18⋆	2.01	2.34	4.56	13.22	1737.25	35.42	11.31	0.48
nesta_case2737sop_mp__sad	795429.36	2.24⋆	2.21	2.42	3.95	13.01	2153.37	32.05	9.69	0.39
nesta_case2746wp_mp__sad	1672150.46	2.41⋆	1.83	2.44	5.43	14.01	2840.32	35.66	13.32	0.56
nesta_case2746wop_mp__sad	1241955.30	2.71⋆	2.48	2.94	5.14	14.51	2306.18	32.41	23.22	0.42
nesta_case3012wp_mp__sad	2635451.29	—	1.92	2.12	n.a.	15.79	13548.13†	46.59	28.41	n.a.
nesta_case3120sp_mp__sad	2203807.23	—	2.56	2.79	n.a.	30.01	16804.55†	53.81	15.69	n.a.
nesta_case9241_pegase__sad	315932.06	—	0.80	1.75	n.a.	80.30	—	3531.62	33437.86	n.a.

bold - the relaxation provided a feasible AC power flow, ⋆ - solver reported numerical accuracy warnings, —,† - iteration or memory limit

UNCLASSIFIED

The Value of Open-Source

TABLE III

QUALITY AND RUNTIME RESULTS OF AC POWER FLOW RELAXATIONS

Test Case	$/h AC	Optimality Gap (%) SDP	QC	SOC	CP	Runtime (seconds) AC	SDP	QC	SOC	CP
Typical Operating Conditions (TYP)										
nesta_case3_lmbd	5812.64	0.39	1.24	1.32	2.99	0.12	4.16	0.07	0.05	0.03
nesta_case5_pjm	17551.89	5.22	14.54	14.54	15.62	0.04	5.36	0.09	0.03	0.05
nesta_case30_ieee	204.97	**0.00**	15.64	15.88	27.91	0.09	8.38	0.17	0.07	0.06
nesta_case118_ieee	3718.64	0.06	1.72	2.07	7.87	0.41	12.62	0.87	0.43	0.05
nesta_case162_ieee_dtc	4230.23	1.08	4.00	4.03	15.44	0.61	35.20	1.48	0.31	0.04
nesta_case300_ieee	16891.28	0.08	1.17	1.18	n.a.	0.80	29.69	2.83	0.65	n.a.
nesta_case2224_edin	38127.69	1.22	6.03	6.09	8.45	11.42	690.16	65.59	45.99	0.33
nesta_case2383wp_mp	1868511.78	0.37	1.04	1.05	5.35	12.41	1966.10	57.87	12.91	0.80
nesta_case3012wp_mp	2600842.72	—	1.00	1.02	n.a.	12.40	14588.79†	53.59	19.15	n.a.
nesta_case9241_pegase	315913.26	—	1.67	—	n.a.	132.25	—	3064.42	—	n.a.
Congested Operating Conditions (API)										
nesta_case3_lmbd__api	367.74	1.26	1.83	3.30	14.79	0.18	4.41	0.09	0.05	0.23
nesta_case6_ww__api	273.76	0.00*	13.14	13.33	17.17	0.34	13.19	0.07	0.06	0.03
nesta_case14_ieee__api	325.56	**0.00**	1.34	1.34	8.89	0.19	5.64	0.11	0.08	0.94
nesta_case24_ieee_rts__api	6421.37	1.45	13.77	20.70	24.12	0.14	7.50	0.26	0.09	0.04
nesta_case30_as__api	571.13	0.00	4.76	4.76	8.01	0.38	6.12	0.17	0.11	1.11
nesta_case30_fsr__api	372.14	11.06	45.97	45.97	48.80	0.25	7.25	0.19	0.09	0.92
nesta_case30_ieee__api	415.53	0.00	1.01	1.01	12.75	0.07	6.60	0.19	0.09	0.03
nesta_case39_epri__api	7466.25	**0.00**	2.97	2.99	13.31	0.10	7.36	0.29	0.12	0.04
nesta_case73_ieee_rts__api	20123.98	4.29	12.01	14.34	17.83	0.48	10.03	0.66	0.20	0.06
nesta_case89_pegase__api	4288.02	18.11	20.39	20.43	22.60	1.16	21.58	1.29	0.81	0.04
nesta_case118_ieee__api	10325.27	31.50	43.93	44.08	49.69	0.46	12.59	0.84	0.25	0.05
nesta_case162_ieee_dtc__api	6111.68	0.85	1.33	1.34	19.39	0.50	36.85	1.53	0.39	0.05
nesta_case189_edin__api	1982.82	0.05	5.78	5.78	n.a.	1.07	16.10	1.14	0.33	n.a.
nesta_case2224_edin__api	46235.43	1.10	2.77	2.77	9.07	12.28	672.04	81.66	88.33	0.33
nesta_case2383wp_mp__api	23499.48	0.10	1.12	1.12	3.10	9.50	1421.39	28.37	10.25	0.34
nesta_case2736sp_mp__api	25437.70	0.07	1.32	1.33	3.89	9.21	2278.77	41.29	10.51	0.36
nesta_case2737sop_mp__api	21192.40	0.00	1.05	1.06	4.62	9.29	1887.22	30.94	9.91	0.32
nesta_case2869_pegase__api	96573.10	0.92*	1.49	1.49	5.16	21.03	1579.87	102.55	161.96	0.37
nesta_case3120sp_mp__api	22874.98	—	3.02	3.03	n.a.	14.92	15018.93†	41.72	12.19	n.a.
nesta_case9241_pegase__api	241975.18	—	2.45	2.59	n.a.	140.73	—	3511.60	8387.11	n.a.
Small Angle Difference Conditions (SAD)										
nesta_case3_lmbd__sad	5992.72	2.06	1.24*	4.28	5.90	0.19	4.39	0.10	0.05	0.03
nesta_case4_gs__sad	324.02	0.05	0.81	4.90	66.06	0.24	4.16	0.06	0.06	0.07
nesta_case5_pjm__sad	26423.32	**0.00**	1.10	3.61	43.95	0.08	5.35	0.11	0.05	0.03
nesta_case6_c__sad	24.43	0.00	0.40	1.36	6.79	0.26	5.32	0.11	0.05	0.02
nesta_case9_wscc__sad	5590.09	0.00	0.41	1.50	6.69	0.14	4.18	0.19	0.05	0.03
nesta_case24_ieee_rts__sad	79804.96	6.05	3.88	11.42	23.56	0.10	6.24	0.30	0.11	0.04
nesta_case29_edin__sad	46933.26	28.44	20.57	34.47	36.79	0.70	9.19	1.73	0.27	0.06
nesta_case30_as__sad	914.44	0.47	3.07	9.16	16.06	0.18	6.49	0.22	0.09	0.03
nesta_case30_ieee__sad	205.11	**0.00**	3.96	5.84	27.96	0.12	7.49	0.18	0.09	0.03
nesta_case73_ieee_rts__sad	235241.70	4.10	3.51	8.37	22.21	0.30	9.48	0.87	0.20	0.07
nesta_case118_ieee__sad	4324.17	7.57	8.32	12.89	20.77	0.56	14.14	0.98	0.31	0.06
nesta_case162_ieee_dtc__sad	4369.19	3.65	6.91	7.08	18.13	0.81	39.71	1.70	0.36	0.05
nesta_case189_edin__sad	914.61	1.20*	2.22	2.25	n.a.	0.65	14.83	1.27	0.46	n.a.
nesta_case300_ieee__sad	16910.23	0.13	1.16	1.26	n.a.	1.01	29.63	2.81	0.76	n.a.
nesta_case2224_edin__sad	38385.14	1.22	5.57	6.18	9.06	11.53	691.53	50.34	65.68	0.33
nesta_case2383wp_mp__sad	1935308.12	1.30	2.97	4.00	8.62	16.25	1785.26	40.71	12.57	0.80
nesta_case2736sp_mp__sad	1337042.77	2.18*	2.01	2.34	4.56	13.22	1737.25	35.42	11.31	0.48
nesta_case2737sop_mp__sad	795429.36	2.24*	2.21	2.42	3.95	13.01	2153.37	32.05	9.69	0.39
nesta_case2746wp_mp__sad	1672150.46	2.41*	1.83	2.44	5.43	14.01	2840.32	35.66	13.32	0.56
nesta_case2746wop_mp__sad	1241955.30	2.71*	2.48	2.94	5.14	14.51	2306.18	32.41	23.22	0.42
nesta_case3012wp_mp__sad	2635451.29	—	1.92	2.12	n.a.	15.79	13548.13†	46.59	28.41	n.a.
nesta_case3120sp_mp__sad	2203807.23	—	2.56	2.79	n.a.	30.01	16804.55†	53.81	15.69	n.a.
nesta_case9241_pegase__sad	315932.06	—	0.80	1.75	n.a.	80.30	—	3531.62	33437.86	n.a.

bold - the relaxation provided a feasible AC power flow, $\star$ - solver reported numerical accuracy warnings, —,† - iteration or memory limit

https://lanl-ansi.github.io/PowerModels.jl/latest/

Software Versions

PowerModels.jl: v0.3.1-18-ga0785a2, a0785a28341986f92cebeee9a4be3482a6dd4d2e

Ipopt.jl: v0.2.6, 959b9c67e396a6e2307fc022d26b0d95692ee6a4

NESTA: v0.6.1, 466cd045d852c8c2cd86167b91ad8fa842ddf3da

Hardware: Dual Intel 2.10GHz CPUs, 128GB RAM

Typical Operating Conditions (TYP)

Case Name	Nodes	Edges	AC ($/h)	QC Gap (%)	SOC Gap (%)	AC Time (sec.)	QC Time (sec.)	SOC Time (sec.)
nesta_case3_cc	3	3	2.0756e+02	1.55	1.62	5	2	2
nesta_case3_cgs	3	3	1.0171e+02	1.69	1.69	5	2	2
nesta_case3_lmbd	3	3	5.8126e+03	1.22	1.32	5	2	2
nesta_case3_ch	3	5	9.8740e+01	100.01	100.01	5	2	2
nesta_case4_gs	4	4	1.5643e+02	0.01	0.01	5	2	2
nesta_case5_pjm	5	6	1.7552e+04	14.55	14.55	5	2	2

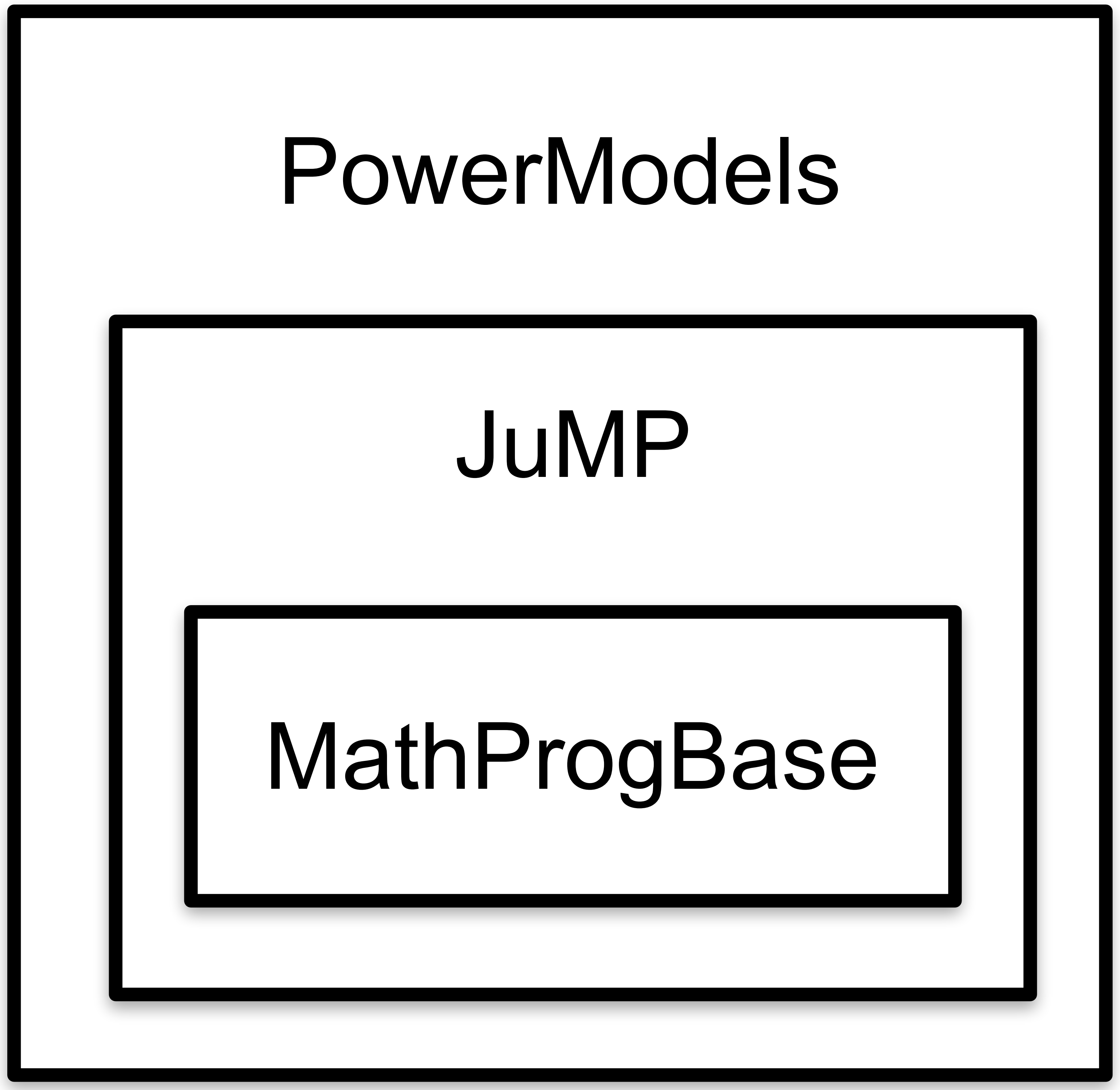
PowerModels.jl Core Features

PowerModels.jl

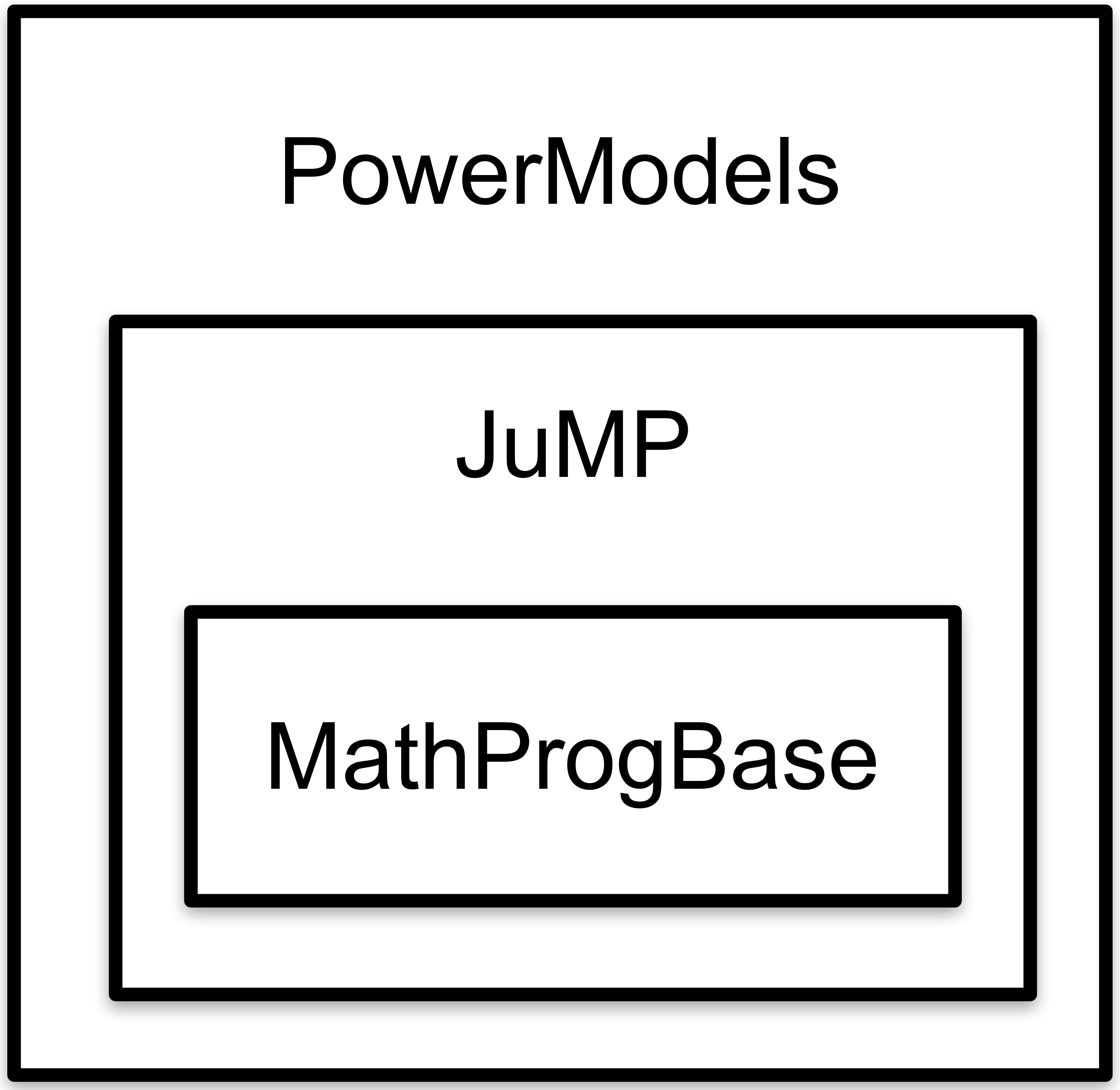
CAUTION

Under Construction

PowerModels.jl Structure



PowerModels.jl Structure



Average user not interested in the modeling details, just wants it to work.

Matpower Data is the R&D Standard

```
function mpc = nesta_case3_lmbd
mpc.version = '2';
mpc.baseMVA = 100.0;

mpc.bus = [
	1	3	110.0	40.0	0.0	0.0	1	1.10000	-0.00000	240.0	1	1.10000	0.90000;
	2	2	110.0	40.0	0.0	0.0	1	0.92617	7.25883	240.0	1	1.10000	0.90000;
	3	2	95.0	50.0	0.0	0.0	1	0.90000	-17.26710	240.0	2	1.10000	0.90000;
];

mpc.gen = [
	1	148.067		54.697	1000.0	-1000.0		1.1	100.0	1	2000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	170.006		-8.791	1000.0	-1000.0		0.92617		100.0	1	2000.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	3	0.0	-4.843	1000.0	-1000.0		0.9	100.0	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
];

mpc.gencost = [
	2	0.0	0.0	3	0.110000	5.000000	0.000000;
	2	0.0	0.0	3	0.085000	1.200000	0.000000;
	2	0.0	0.0	3	0.000000	0.000000	0.000000;
];

mpc.branch = [
	1	3	0.065	0.62	0.45	9000.0	0.0	0.0	0.0	0.0	1	-30.0	30.0;
	3	2	0.025	0.75	0.7	50.0	0.0	0.0	0.0	0.0	1	-30.0	30.0;
	1	2	0.042	0.9	0.3	9000.0	0.0	0.0	0.0	0.0	1	-30.0	30.0;
];
```

Parsing Matpower Files

```
using PowerModels

network_data = PowerModels.parse_file(“nesta_case3_lmbd.m”)
```

Parsing Matpower Files

raw text

```
using PowerModels

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")
```

Parsing Matpower Files

julia dictionary

raw text

```
using PowerModels

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")
```

UNCLASSIFIED

Parsing Matpower Files

julia dictionary

raw text

```
using PowerModels

network_data = PowerModels.parse_file(“nesta_case3_lmbd.m”)

println(network_data[“bus”][“1”][“pd”])
> 1.1
```

Parsing Matpower Files

julia dictionary

raw text

```
using PowerModels

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")

println(network_data["bus"]["1"]["pd"])
> 1.1
```

Parser supports user-defined extensions to the Matpower format

https://lanl-ansi.github.io/PowerModels.jl/latest/data.html

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()
```

UNCLASSIFIED

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf(“nesta_case3_lmbd.m”, solver)
```

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf(“nesta_case3_lmbd.m”, solver)

result = run_dc_opf(“nesta_case3_lmbd.m”, solver)
```

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf("nesta_case3_lmbd.m", solver)

result = run_dc_opf("nesta_case3_lmbd.m", solver)


run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
```

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf(“nesta_case3_lmbd.m”, solver)

result = run_dc_opf(“nesta_case3_lmbd.m”, solver)

run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
```

Non-Convex Form

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf("nesta_case3_lmbd.m", solver)

result = run_dc_opf("nesta_case3_lmbd.m", solver)


run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)
```

Non-Convex Form

Linear Approximation

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf(“nesta_case3_lmbd.m”, solver)

result = run_dc_opf(“nesta_case3_lmbd.m”, solver)


run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)

run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
```

Non-Convex Form

Linear Approximation

Convex Relaxation

Your First PowerModel (OPF)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_ac_opf(“nesta_case3_lmbd.m”, solver)

result = run_dc_opf(“nesta_case3_lmbd.m”, solver)

run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)

run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
```

Non-Convex Form

Linear Approximation

Convex Relaxation

Inspecting the Results

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
```

Inspecting the Results

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
```

julia dictionary (standard structure)

Inspecting the Results

julia dictionary (standard structure)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

println(result["objective"])
> 5812.64293503618
```

Inspecting the Results

julia dictionary (standard structure)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

println(result[“objective”])
> 5812.64293503618

println(result[“solve_time”])
> 0.009732971
```

Inspecting the Results

julia dictionary (standard structure)

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

println(result[“objective”])
> 5812.64293503618

println(result[“solve_time”])
> 0.009732971

println(result[“solution”])
> Dict{String,Any}(Pair{String,Any}(“baseMVA”,100.0),Pair{String,Any}
("gen",Dict{String,Any}(Pair{String,Any}("1",Dict{String,Any}…
```

Modifying Network Data

```
using PowerModels; using Ipopt
solver = IpoptSolver()

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")
```

Modifying Network Data

```
using PowerModels; using Ipopt
solver = IpoptSolver()

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")

network_data["bus"]["3"]["pd"] = 0.0
network_data["bus"]["3"]["qd"] = 0.0

result_1 = run_ac_opf(network_data, solver)
```

Modifying Network Data

```
using PowerModels; using Ipopt
solver = IpoptSolver()

network_data = PowerModels.parse_file("nesta_case3_lmbd.m")

network_data["bus"]["3"]["pd"] = 0.0
network_data["bus"]["3"]["qd"] = 0.0

result_1 = run_ac_opf(network_data, solver)

network_data["bus"]["3"]["pd"] = 1.0
network_data["bus"]["3"]["qd"] = 0.5

result_2 = run_ac_opf(network_data, solver)
```

Solving Different Problems

```
using PowerModels; using Ipopt
solver = IpoptSolver()
```

Solving Different Problems

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)
```

Solving Different Problems

Problem Class

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)
```

Solving Different Problems

Problem Class

Problem Formulation

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)
```

Solving Different Problems

Problem Class

Problem Formulation

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)

# Linear Approximation
run_pf(“case5_pjm_tnep.m", DCPPowerModel, solver)
run_opf("case5_pjm_tnep.m", DCPPowerModel, solver)
run_ots("case5_pjm_tnep.m", DCPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", DCPPowerModel, solver)
```

Solving Different Problems

Problem Class

Problem Formulation

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)

# Linear Approximation
run_pf(“case5_pjm_tnep.m", DCPPowerModel, solver)
run_opf("case5_pjm_tnep.m", DCPPowerModel, solver)
run_ots("case5_pjm_tnep.m", DCPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", DCPPowerModel, solver)
```

Solving Different Problems

Problem Class

Problem Formulation

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)
```

Linear Formulation

```
# Linear Approximation
run_pf(“case5_pjm_tnep.m", DCPPowerModel, solver)
run_opf("case5_pjm_tnep.m", DCPPowerModel, solver)
run_ots("case5_pjm_tnep.m", DCPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", DCPPowerModel, solver)
```

Solving Different Problems

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)

# Convex Relaxation
run_pf(“case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_opf("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_ots("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_tnep("case5_pjm_tnep.m", SOCWRPowerModel, solver)
```

Solving Different Problems

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)

# Convex Relaxation
run_pf(“case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_opf("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_ots("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_tnep("case5_pjm_tnep.m", SOCWRPowerModel, solver)
```

Convex Formulation

Solving Different Problems

This software design helps to organize 100s of possible Problem / Formulation combinations

```
using PowerModels; using Ipopt
solver = IpoptSolver()

# Base Non-Convex Model
run_pf(“case5_pjm_tnep.m", ACPPowerModel, solver)
run_opf("case5_pjm_tnep.m", ACPPowerModel, solver)
run_ots("case5_pjm_tnep.m", ACPPowerModel, solver)
run_tnep("case5_pjm_tnep.m", ACPPowerModel, solver)

# Convex Relaxation
run_pf(“case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_opf("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_ots("case5_pjm_tnep.m", SOCWRPowerModel, solver)
run_tnep("case5_pjm_tnep.m", SOCWRPowerModel, solver)
```

Convex Formulation

Where is JuMP?

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
```

Where is JuMP?

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)
result = solve_generic_model(pm, solver)
```

Where is JuMP?

PowerModels Internal Data Structure

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)
result = solve_generic_model(pm, solver)
```

Where is JuMP?

PowerModels Internal Data Structure

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)
result = solve_generic_model(pm, solver)
```

```
pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)

println(pm.model) # show / modify the JuMP model

result = solve_generic_model(pm, solver)
```

Where is JuMP?

PowerModels Internal Data Structure

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)
result = solve_generic_model(pm, solver)
```

```
pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)

println(pm.model) # show / modify the JuMP model

result = solve_generic_model(pm, solver)
```

JuMP Model

Where is JuMP?

```
using PowerModels; using Ipopt
solver = IpoptSolver()

result = run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)

pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)
result = solve_generic_model(pm, solver)
```

PowerModels Internal Data Structure

```
pm = build_generic_model("nesta_case3_lmbd.m", ACPPowerModel, PowerModels.post_opf)

println(pm.model) # show / modify the JuMP model

result = solve_generic_model(pm, solver)
```

JuMP Model

PowerModels Problem Definition (OPF)

```
function post_opf(pm::GenericPowerModel)
    variable_voltage(pm)
    variable_generation(pm)
    variable_line_flow(pm)

    objective_min_fuel_cost(pm)

    constraint_theta_ref(pm)
    constraint_voltage(pm)

    for (i,bus) in pm.ref[:bus]
        constraint_kcl_shunt(pm, bus)
    end

    for (i,branch) in pm.ref[:branch]
        constraint_ohms_yt_from(pm, branch)
        constraint_ohms_yt_to(pm, branch)

        constraint_phase_angle_difference(pm, branch)

        constraint_thermal_limit_from(pm, branch)
        constraint_thermal_limit_to(pm, branch)
    end
end
```

PowerModels Problem Definition (OPF)

```
function post_opf(pm::GenericPowerModel)
    variable_voltage(pm)
    variable_generation(pm)
    variable_line_flow(pm)

    objective_min_fuel_cost(pm)

    constraint_theta_ref(pm)
    constraint_voltage(pm)

    for (i,bus) in pm.ref[:bus]
        constraint_kcl_shunt(pm, bus)
    end

    for (i,branch) in pm.ref[:branch]
        constraint_ohms_yt_from(pm, branch)
        constraint_ohms_yt_to(pm, branch)

        constraint_phase_angle_difference(pm, branch)

        constraint_thermal_limit_from(pm, branch)
        constraint_thermal_limit_to(pm, branch)
    end
end
```

variables:

$$S_i^g \ \ \forall i \in N$$
$$V_i \ \ \forall i \in N$$

minimize:

$$\sum_{i \in N} f(S_i^g)$$

subject to:

$$(\boldsymbol{v}_i^{\boldsymbol{l}})^2 \leq V_i V_i^* \leq (\boldsymbol{v}_i^{\boldsymbol{u}})^2 \ \ \forall i \in N$$
$$\boldsymbol{S}_i^{\boldsymbol{gl}} \leq S_i^g \leq \boldsymbol{S}_i^{\boldsymbol{gu}} \ \ \forall i \in N$$
$$S_i^g - \boldsymbol{S}_i^d = \sum_{(i,j) \in E \cup E^R} S_{ij} \ \ \forall i \in N$$
$$S_{ij} = \boldsymbol{Y}_{ij}^* V_i V_i^* - \boldsymbol{Y}_{ij}^* V_i V_j^* \ \ (i,j) \in E \cup E^R$$
$$|S_{ij}|^2 \leq (\boldsymbol{s}_{ij}^{\boldsymbol{u}})^2 \ \ \forall (i,j) \in E \cup E^R$$
$$-\boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta}} \leq \angle(V_i V_j^*) \leq \boldsymbol{\theta}_{ij}^{\boldsymbol{\Delta}} \ \ \forall (i,j) \in E$$

PowerModels Problem Definition (OPF)

```
function post_opf(pm::GenericPowerModel)
    variable_voltage(pm)
    variable_generation(pm)
    variable_line_flow(pm)

    objective_min_fuel_cost(pm)

    constraint_theta_ref(pm)
    constraint_voltage(pm)

    for (i,bus) in pm.ref[:bus]
        constraint_kcl_shunt(pm, bus)
    end

    for (i,branch) in pm.ref[:branch]
        constraint_ohms_yt_from(pm, branch)
        constraint_ohms_yt_to(pm, branch)

        constraint_phase_angle_difference(pm, branch)

        constraint_thermal_limit_from(pm, branch)
        constraint_thermal_limit_to(pm, branch)
    end
end
```

Implicit

variables:

$$S^g_i \quad \forall i \in N$$

$$V_i \quad \forall i \in N$$

minimize:

$$\sum_{i \in N} f(S^g_i)$$

subject to:

$$(\boldsymbol{v}^{\boldsymbol{l}}_i)^2 \le V_i V_i^* \le (\boldsymbol{v}^{\boldsymbol{u}}_i)^2 \quad \forall i \in N$$

$$\boldsymbol{S}^{\boldsymbol{gl}}_i \le S^g_i \le \boldsymbol{S}^{\boldsymbol{gu}}_i \quad \forall i \in N$$

$$S^g_i - \boldsymbol{S}^d_i = \sum_{(i,j) \in E \cup E^R} S_{ij} \quad \forall i \in N$$

$$S_{ij} = \boldsymbol{Y}^*_{ij} V_i V_i^* - \boldsymbol{Y}^*_{ij} V_i V_j^* \quad (i,j) \in E \cup E^R$$

$$|S_{ij}|^2 \le (\boldsymbol{s}^{\boldsymbol{u}}_{ij})^2 \quad \forall (i,j) \in E \cup E^R$$

$$-\boldsymbol{\theta}^{\boldsymbol{\Delta}}_{ij} \le \angle(V_i V_j^*) \le \boldsymbol{\theta}^{\boldsymbol{\Delta}}_{ij} \quad \forall (i,j) \in E$$

PowerModels.jl Road Map

Versions Convention

vX.Y.Z

Versions Convention

Will be zero for some time

vX.Y.Z

Versions Convention

Will be zero for some time

vX.Y.Z

breaking changes

Versions Convention

Will be zero for some time

vX.Y.Z

breaking changes

Non-breaking changes

Versions Past and Planned

- **v0.1.0** (2016 Q2-Q3)
  - First draft (basically learning Julia / JuMP)

Versions Past and Planned

- **v0.1.0** (2016 Q2-Q3)
  - First draft (basically learning Julia / JuMP)
- **v0.2.0** (2016 Q3-Q4)
  - First public version, Thanks to Miles

Versions Past and Planned

- **v0.1.0** (2016 Q2-Q3)
  - First draft (basically learning Julia / JuMP)
- **v0.2.0** (2016 Q3-Q4)
  - First public version, Thanks to Miles
- **v0.3.0** (2017 Q1-**Present**)
  - Significant engineering improvements

Versions Past and Planned

- **v0.1.0** (2016 Q2-Q3)
  - First draft (basically learning Julia / JuMP)
- **v0.2.0** (2016 Q3-Q4)
  - First public version, Thanks to Miles
- **v0.3.0** (2017 Q1-**Present**)
  - Significant engineering improvements
- **v0.4.0** (2017, I hope)
  - Massive renaming of stuff
  - Adding many more formulations from the literature

Contributions Welcome!

Contributions Welcome!

- This is a community resource for ***established*** problems and formulations

Contributions Welcome!

- This is a community resource for ***established*** problems and formulations
- Excited to add,
  - New **problem classes**
  - New **formulations** (especially complex ones, e.g. moment-based relaxations)

Contributions Welcome!

- This is a community resource for ***established*** problems and formulations
- Excited to add,
  - New **problem classes**
  - New **formulations** (especially complex ones, e.g. moment-based relaxations)
- Addressing anything in the **github issues**

Questions / Comments?

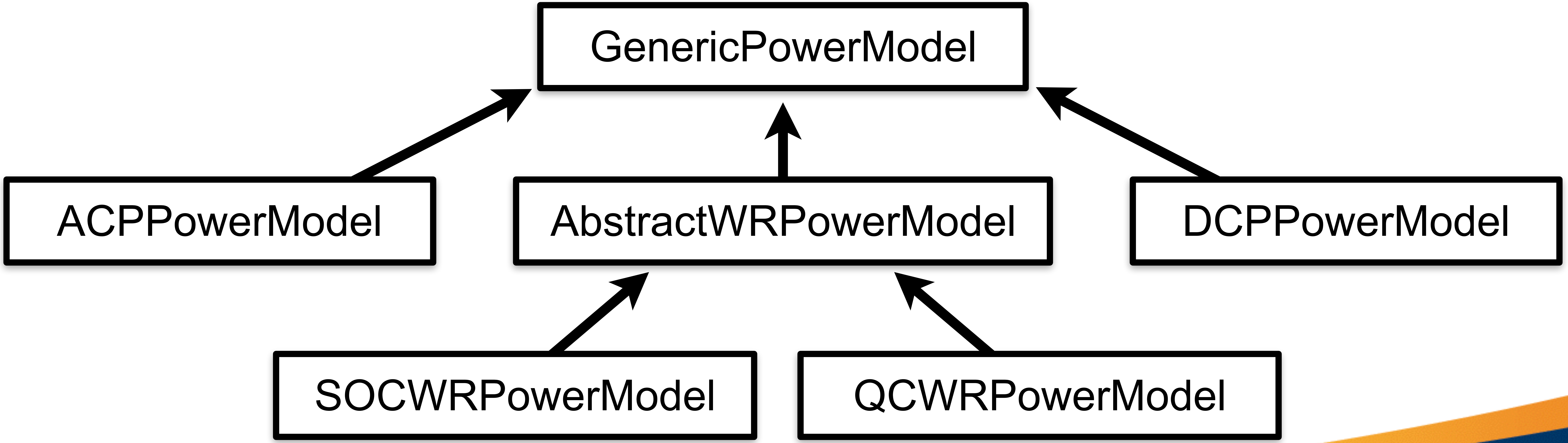
cjc@lanl.gov

Why have these ***PowerModel Things?

```
run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
...
```

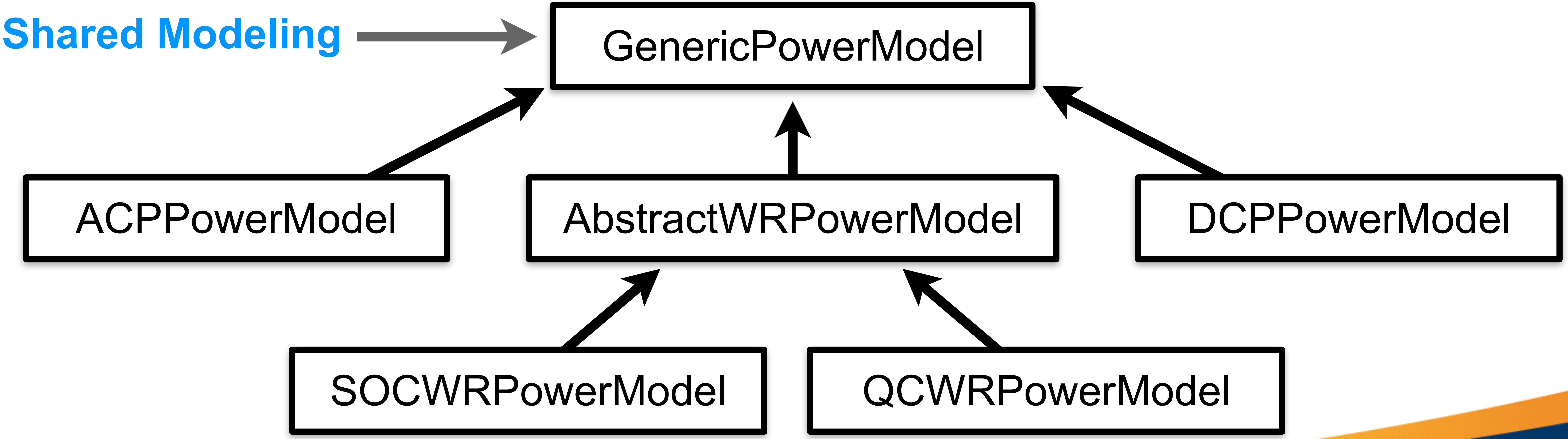
Why have these ***PowerModel Things?

```
run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
...
```



Why have these ***PowerModel Things?

```
run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
...
```



Why have these ***PowerModel Things?

```
run_opf("nesta_case3_lmbd.m", ACPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", DCPPowerModel, solver)
run_opf("nesta_case3_lmbd.m", SOCWRPowerModel, solver)
...
```