ACOPF models: extending data, formulations, and solution methodology

Michael C. Ferris
Zev Friedman, Dan Molzahn, Chris DeMarco and Bernie Lesieutre

Computer Sciences and Wisconsin Institutes for Discovery
University of Wisconsin, Madison

Increasing Real-Time and Day-Ahead Market Efficiency through Improved Software
June 26, 2013

Quotes from Wisconsin

- “Since all models are wrong the scientist cannot obtain a correct one by excessive elaboration”, Box, 1976.
- “Essentially, all models are wrong, but some are useful”, Box, 1987.
- “Using the wrong algorithm on the wrong model”, Alvarado, June 25, 2013.

Quotes from Wisconsin

- "Since all models are wrong the scientist cannot obtain a correct one by excessive elaboration", Box, 1976.
- "Essentially, all models are wrong, but some are useful", Box, 1987.
- "Using the wrong algorithm on the wrong model", Alvarado, June 25, 2013.

- Industry: problems abound, data is vast, models are adequate, (folklore of) tricks and heuristics
- Academia: theory is strong, models are rich, data is poor, solving the wrong problem

- How do we bridge this gap? We aim to build a collection of authentic (simpler/focussed) models (existing and new formulations) tied to optimization solver technology

Not another call for data

- Optimization appears central to many design and operational models in power flow - our data format is optimization centric
- Problems are at the engineering and economic interface - data has both elements in same location
- Harness existing datasets and provide conversion to/from PSSE, MatPower, etc
- Provide tools and examples that demonstrate and enable collections of models, solvable by both commercial and academic solution engines

Not another call for data

- Optimization appears central to many design and operational models in power flow - our data format is optimization centric
- Problems are at the engineering and economic interface - data has both elements in same location
- Harness existing datasets and provide conversion to/from PSSE, MatPower, etc
- Provide tools and examples that demonstrate and enable collections of models, solvable by both commercial and academic solution engines
- Use to answer questions like:
  - Why use ACOPF?
  - Why use off the shelf NLP?
  - What to use our CPU for?

Why use ACOPF? Or not?

- Better physics: ACOPF provides information on voltage magnitudes and reactive power that are not available from DCOPF.
- How bad is DCOPF: Overbye concludes they are close (engineering), but a 5% error in LMPs corresponds to a LOT of money. Gaming opportunities, picking the wrong winner or loser
- Are constraints binding/violated in one and not the other? (Atypical operating conditions)
- Local solutions? Difficulty in solving quickly, reliably, accurately.
- Can add (proxy) constraints to DCOPF that do well enough

Conversion utilities/extending data (using GAMS)

- To/from: Matpower, psse, xls, gdx
- Raw format: just basic data
- Add features to create case (gdx data):

```
calc_S_matrix.gms
calc_active_limits.gms
calc_cost_curves.gms
calc_line_limits.gms
calc_reactive_limits.gms
```

- Process data, save solutions

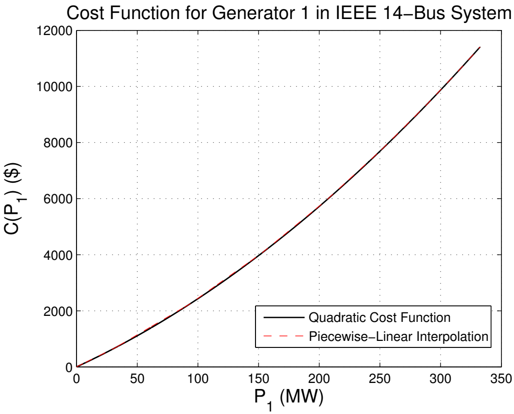
```
extract_data.gms
calc_Ybus.gms
dcopf_shift.gms
piecewise_costs.gms
reactive_limits.gms

save_solution.gms
```

Linear Interpolation of Quadratic Cost Functions

Piecewise-linear interpolations of quadratic cost functions: use the roots of Legendre polynomials

0
0.095012509837637
0.281603550779259
0.458016777657227
0.617876244402644
0.755404408355003
0.865631202387832
0.944575023073233
0.989400934991650
1



Estimating Line-Flow Limits

Specifying reasonable line-flow limits requires two quantities: the surge impedance loading for the line and an estimate of the line length. We approximate these quantities using power flow data and assumptions of line geometry and material properties.

$$Z_{base} = \frac{V_{base}^2}{S_{base}} \qquad [\Omega]$$

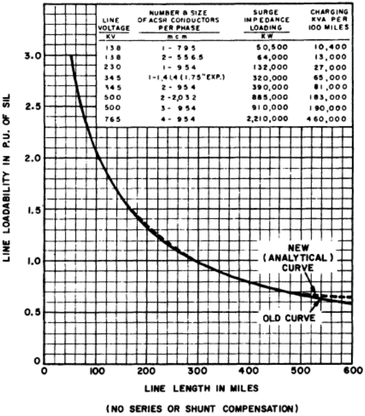
$$L = \frac{X_{pu}}{2\pi 60} Z_{base} \qquad [\mathrm{H}]$$

$$R = R_{pu} Z_{base} \qquad [\Omega]$$

$$C = \frac{B_{pu}}{2\pi 60} \frac{1}{Z_{base}} \qquad [\mathrm{F}]$$

$$Z_c = \sqrt{\frac{R + j2\pi 60 L}{j2\pi 60 C}} \qquad [\Omega]$$

$$SIL = \frac{V_{rated}^2}{|Z_c|} \qquad [\mathrm{W}]$$



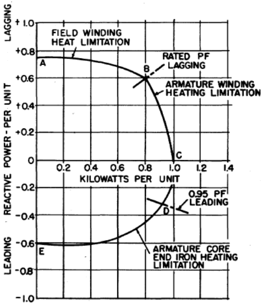
LINE VOLTAGE KV	NUMBER & SIZE OF ACSR CONDUCTORS PER PHASE mcm	SURGE IMPEDANCE LOADING KW	CHARGING KVA PER 100 MILES
138	1 - 795	50,500	10,400
138	2 - 556.5	64,000	13,000
230	1 - 954	132,000	27,000
345	1-1,414 (1.75" EXP.)	320,000	65,000
345	2 - 954	390,000	81,000
500	2 -2,032	885,000	183,000
500	3 - 954	910,000	190,000
765	4 - 954	2,210,000	460,000

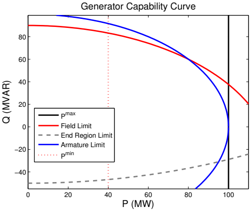
Figure 7. Comparison between "analytical" and "old" curves

Estimating Generator Capability Curves

Simplistic generator models often use "rectangle constraints" for active and reactive output limits. More detailed modeling: "D-curves."

- Reactive power of a synchronous generator constrained by several factors: armature current limit, field current limit, end region heating limit
- Each limit modeled as circle
- The machine must operate within the intersection of these circles.
- The generator must also operate within maximum and minimum active power limits imposed by the prime mover





Active Power Ramp Rates and other features

- Estimate generator active power ramp up and ramp down rates, respectively, as functions of nameplate capacity.
  (RTO Unit Commitment Test System, Federal Energy Regulatory Commission, Staff Report, July 2012)
- DC lines
- Adjustable Transformers (phase shifting and voltage tap changing)
- Transformer Impedance Correction Data
- Switched Shunt Devices
- FACTs Devices
- Multisection Lines, loop flows
- Demand bids
- Startup costs
- Scenarios, uncertainty, external influences

Model and solution examples

```
dcopf.gms
polar_acopf.gms
iv_acopf.gms
rect_acopf.gms

feasibility_reactive_limits.gms
feasibility_dcopf.gms
feasibility_*.gms

ybus_*_acopf.gms

condensed_dcopf.gms
condensed_*_acopf.gms
```

Why use optimization modeling software?

- Allows interplay between models - use dcopf for starting point, pass onto acopf - automatic setting of multipliers
- Easy to switch solvers
- Has many more “standard” model types: MINLP, MPEC, SDP, EMP
- EMP: Scalar quadratic penalties, soft limit penalties, multi-stage stochastic programs, risk measures
- Transparency: “dirty tricks” are explicit
- Portability: models can run on multiple architectures
- Interaction with optimization community
- Special structure: pros and cons
- Grid, GUSS, Dynamics - AMPL extensions
- Higher level definition of logical (e.g. up-to) constraints
- Ability to compare solvers (e.g. Castillo)

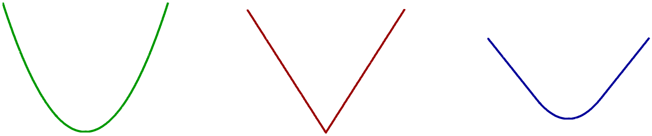
Comparisons

Same data, solver, and host (other solvers do much better on some)

Test Case	Objective	Matpwr	GAMS	Best GAMS mo
IEEE 14 Bus	8.08152e+03	0.25	0.090	Polar YBus 2 (F
IEEE 24 Bus	6.33522e+04	0.33	0.096	Polar YBus 0 (M
IEEE 30 Bus	5.76892e+02	0.10	0.119	Polar Condense
IEEE 39 Bus	4.18641e+04	0.29	0.111	IV Full 2 (Flat s
IEEE 57 Bus	4.17377e+04	0.14	0.106	Polar Full 0 (Mi
IEEE 118 Bus	1.29660e+05	0.33	0.176	Polar YBus 0 (M
IEEE 300 Bus	7.19725e+05	0.32	0.362	Polar YBus 0 (M
Polish 2383wp	1.86851e+06	3.03	3.457	Polar YBus 0 (M
Polish 2736sp	1.30788e+06	3.02	4.356	Polar YBus 0 (M
Polish 2737sop	7.77629e+05	2.68	4.267	Polar Full 0 (Mi
Polish 2746wp	1.63177e+06	3.42	4.871	Polar YBus 0 (M
Polish 3012wp	2.591706565e+06	4.02	5.937	Polar YBus 0 (M
Polish 3120sp	2.142703764e+06	4.28	6.839	Polar Condense
Polish 3375wp*	7.412030674e+06*	180.24*	9.480	Polar YBus 4 (P

ENLP: Primal problem

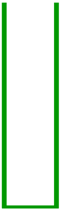
$$\min_{x \in X} f_0(x) + \theta(f_1(x), \ldots, f_m(x))$$



$$\theta(u) = \begin{cases} \gamma u - \frac{1}{2}\gamma^2 & \text{if } u \geq \gamma \\ \frac{1}{2}u^2 & \text{if } u \in [-\gamma, \gamma] \\ -\gamma u - \frac{1}{2}\gamma^2 & \text{else} \end{cases}$$

Huber function used in robust statistics.

More general θ functions



In general any piecewise linear penalty function can be used (different upside/downside costs)
General form:

$$\theta(u) = \sup_{y \in Y} \{ y'u - k(y) \}$$

θ can take on ∞ and may be nonsmooth; it is convex.

Elegant Duality

For these θ (defined by $k(\cdot)$, Y), duality is derived from the Lagrangian:

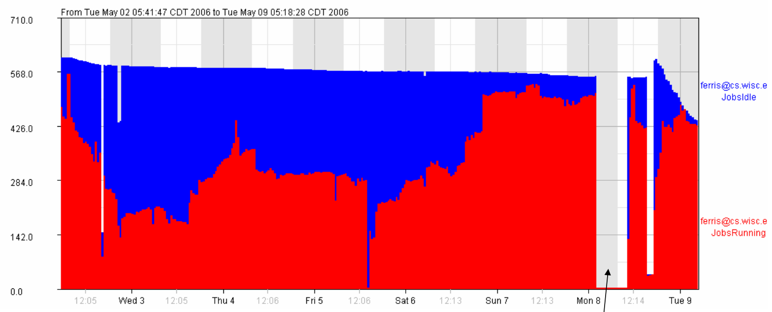
$$\mathcal{L}(x,y) = f_0(x) + \sum_{i=1}^{m} y_i f_i(x) - k(y)$$
$$x \in X, y \in Y$$

- Several ways to reformulate.
- EMP automatically creates an MCP:

```
model enlp /    gradLx.x,
               -gradLy.y /;
solve enlp using ecp;
```

How to solve: Gams/Grid

- solvelink = 3;
- solve mod using minlp min obj;
- execute_loadhandle mod;
- Multiple jobs spawned to grid, collectable asynchronously
- Computation configurable (e.g. Condor, OS process, Amazon)



main submitting machine died, jobs not lost

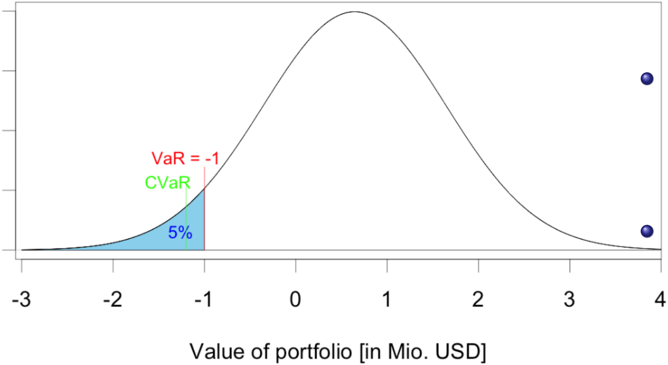
Partitioned into 1000 subproblems, over 300 machines running for multiple days

What to use our CPU for?

- Transmission line switching
  - CPLEX and Gurobi can be run multi-threaded
  - Options can perform significantly better than defaults
  - Reformulations (using SOS1 variables) work better
- 2 scenarios better than 1
  - Can generate models by sampling (SAA), more distributions
  - Different risk measures
  - Benders decomposition, importance sampling as solver option
  - Can validate solutions using different samples (reproduceable over different machines)
- Switch to SOCP or SDP relaxations - Mosek solver

Optimization of risk measures

- Determine portfolio weights w_j for each of a collection of assets
- Asset returns v are random, but jointly distributed
- Portfolio return $r(w, v)$



- Value at Risk (VaR) can be viewed as a chance constraint (hard):
- CVaR gives rise to a convex optimization problem (easy)

- Chance constraints (implemented using mixed integer programming):

$$\min_x c^T x \text{ s.t. } Pr(Ax \leq b) \geq \pi$$

Example: Portfolio Model

- Maximize the mean of the lower tail (mean tail loss):

$$\begin{array}{ll} \max & \underline{CVaR}_{\alpha}(r) \\ \text{s.t.} & r = \sum_j v_j * w_j \\ & \sum_j w_j = 1, \;\; w \geq 0 \end{array}$$

- Jointly distributed random variables v, realized at stage 2
- Variables: portfolio weights w in stage 1, returns r in stage 2
- Coherent risk measures $\mathbb{E}$ and $\underline{CVaR}$ (or convex combination)

Example: Portfolio Model

- Maximize the mean of the lower tail (mean tail loss):

$$\begin{array}{ll} \max & \underline{CVaR}_\alpha(r) \\ \text{s.t.} & r = \sum_j v_j * w_j \\ & \sum_j w_j = 1, \; w \geq 0 \end{array}$$

- Jointly distributed random variables v, realized at stage 2
- Variables: portfolio weights w in stage 1, returns r in stage 2
- Coherent risk measures $\mathbb{E}$ and $\underline{CVaR}$ (or convex combination)
- Optimization modeling systems have new tools for sampling, risk measures and solution of stochastic programs
- Classical: mean-variance model (Markowitz)

$$\begin{array}{ll} \min & w^T \Sigma w - q \sum_j v_j * w_j \\ & \sum_j w_j = 1, \; w \geq 0 \end{array}$$

Conclusions

- Collections of, and interactions between, models are critical
- Uncertainty is present everywhere: we need to hedge/control/ameliorate it
- Modern computational optimization tools can be very fast, deal with large amounts of data and variables, address non-convex and discrete issues, interact with dynamics
- Modeling systems allow quick prototyping, switching between formats, state-of-the-art solvers, portability and transparency
- Will be testable using NEOS system (online without purchase of GAMS)
- FERC contract: will make available at Wisconsin after approval