Benchmarking Large-Scale ACOPF Solutions and Optimality Bounds

Smitha Gopinath
Theoretical Division
Los Alamos National Laboratory
Los Alamos, USA
smitha@lanl.gov

Hassan L. Hijazi
Theoretical Division
Los Alamos National Laboratory
Los Alamos, USA
hlh@lanl.gov

Abstract—We present the results of a comprehensive benchmarking effort aimed at evaluating and comparing state-of-the-art open-source tools for solving the Alternating-Current Optimal Power Flow (ACOPF) problem. Our numerical experiments include all instances found in the public library PGLIB with network sizes up to 30,000 nodes. The benchmarked tools span a number of programming languages (Python, Julia, Matlab/Octave, and C++) and nonlinear optimization solvers (Ipopt, MIPS, and INLP) as well as different mathematical modeling tools (JuMP and Gravity). We also present state-of-the-art optimality bounds obtained using sparsity-exploiting semidefinite programming approaches and corresponding computational times.

Index Terms—Large-scale, ACOPF, Global optimization

I. INTRODUCTION

With the revenues of the energy sector reaching close to $400 billion per year [1], power systems optimization plays a central role in maximizing system reliability while minimizing costs. The Alternating-Current Optimal Power Flow (ACOPF) problem lies at the heart of all power systems optimization. Very recently, the Advanced Research Projects Agency-Energy (ARPA-E) hosted the Grid Optimization Competition putting the ACOPF at the forefront of impactful research in this field with $2.4 million in prizes at stake.

In this paper, we address the performance of state-of-the-art techniques in providing feasible solutions along with optimality bounds for the ACOPF problem. We compare four open-source implementations capable of producing feasible solutions fast: Gravity (C++), PowerModels.jl (Julia), Matpower (Matlab/Octave) and GridOpt (Python). To get lower bounds and compute optimality gaps, we revisit the polynomial semidefinite programming (SDP)-Reformulation-Linearization Technique (RLT) relaxation for ACOPF developed in [2]. While the relaxation has so far been used to prove optimality on all networks up to 300 buses, we present new results on networks up to 30,000 buses. To the best of our knowledge, this is the first published global optimality proof for a number of these instances. We benchmark the performance of this approach against a recent SDP-based moment relaxation introduced by Wang et al. in [3].

II. ACOPF

A. Notations

Grid Parameters:

- $\bar{S}^g_i = (p^g_i, q^g_i)$ Complex, active and reactive power generation at node $i$
- $c_{ij}, c_x, c_{2x}$ Generation cost coefficients at node $i$
- $t_{ij}$ Thermal limit along line $(i,j)$
- $Y_{ij}$ Complex admittance of line $(i,j)$
- $Y^+_{ij}$ Line charging admittance of line $(i,j)$
- $Y^*_{ij}$ Bus shunt admittance of bus $i$
- $S^0_{ij}$ Transformer properties (tap change and phase shift)
- $p^0_i + i q^0_i$ Voltage magnitude bounds at node $i$
- $\theta_{ij} \geq -\pi/2, \theta_{ij} \leq \pi/2$ Phase angle difference bounds along line $(i,j)$

Grid Variables:

- $\bar{V}_i$ Complex, active and reactive power flow from $i$ to $j$
- $I_{ij}$ Complex voltage at node $i$
- $I_{ij}$ Complex current flowing from $i$ to $j$ at end $i$
- $I_{ij}$ Squared magnitude of current flowing from $i$ to $j$ at end $i$
- $W_{ij}$ Lifted voltage product $V_i^\dagger V_j$

Other notations:

- $\mathcal{G} = (N,E)$ Graph with nodes $N$ and edges $E$
- $\mathcal{R}(\cdot)$ Real component of complex number
- $V^\dagger$ Imaginary component of complex number
- $V^*$ Conjugate of complex number
- $|V|^2$ Square Magnitude of complex number $|V|^2 = \mathcal{R}(V)^2 + \mathcal{I}(V)^2$
- $\# , \#$ Lower and upper bounds on $x$

B. Formulations

The extended (non-convex) ACOPF problem (with line charging, bus shunts, transformers) is presented in Model 1. Note that complex inequalities correspond to component-wise constraints on the respective real and imaginary parts.

Model 1 Extended ACOPF

minimize:

$$\sum_{i \in N} c_{0i} + c_{1i} \mathcal{R}(S^g_i) + (c_{2i} \mathcal{R}(S^x_i))^2$$

subject to:

$$S_{ij} = (Y^*_{ij} + Y^0_{ij}) \frac{|V_i|^2 |V_j|^2 - Y_{ij} \bar{V}_i^\dagger \bar{V}_j^\dagger}{|T_{ij}|^2}, \ \forall (i,j) \in E$$

(1a)

$$S_{ij} = (Y^*_{ij} + Y^0_{ij}) |V_j|^2 - Y_{ij} \bar{V}_i^\dagger \bar{V}_j^\dagger, \ \forall (i,j) \in E$$

(1b)

$$S^x_i - S^0_i - (V^\dagger_i |V_i|^2) = \sum_{(i,j) \in E} S_{ij}, \ \forall i \in N$$

(1c)

$$|V_i| \leq \bar{V}_i, \ \forall i \in N$$

(1d)

$$\theta_{ij} \mathcal{I}(V_i^\dagger V_j^\dagger) \leq \mathcal{R}(V_i^\dagger V_j^\dagger) \leq \theta_{ij} \mathcal{I}(V_i^\dagger V_j^\dagger), \ \forall (i,j) \in E$$

(1e)

$$S^x_i \leq S^0_i \leq S^0_1, \ \forall i \in N$$

(1f)

$$|S_{ij}|^2 \leq t_{ij}, \ \forall (i,j), (j,i) \in E.$$  

(1g)
We next present the convex relaxation of Model 1, developed in [2]. In Model 2, we introduce the variable $W_{ij}$ to denote the lifted variable $V_i V_j^*$ and $W$ the corresponding matrix. The real component of power flow $S_{ij}$ is denoted by $p_{ij}$, whereas $q_{ij}$ denotes the imaginary component. Variables $\tilde{p}_{ij}$ and $\tilde{q}_{ij}$ represent squares of real and imaginary power flows on each edge $(i, j)$. Variables $\hat{W}_{ij}$ and $\hat{W}_{ji}$ are introduced to represent products of current magnitude and the squared voltage at each end of an edge.

In (2i), $T(G)$ is a tree decomposition of a chordal completion of the network graph $G$ [4]. For each clique (fully-connected sub-graph) $c$ in $T(G)$, $W_c$ is a submatrix of $W$, contains elements of $W$ that correspond to the nodes in $c$. $M(W_c)$ is the set of all the $2 \times 2$ and $3 \times 3$ principal minors of $W_c$. Sets $Sec(c, x, y)$ and $MC(c, x, y, z)$ are defined in [2].

Power flow constraints, KCL and voltage-angle constraints are given by (2a), (2b), (2c) and (2d). The thermal limit constraint in lifted variables is given by (2e). RLT-based valid inequalities on current magnitude are given by (2f), (2h), (2g), (2i). Polynomial determinant cuts are given in (2j). Bound constraints are given in (2k) - (2q) and concave and convex envelopes for $\hat{W}_{ij}$, $\hat{W}_{ji}$, $\tilde{p}_{ij}$, $\tilde{q}_{ij}$ are given in (2r) and (2s).

\textbf{Model 2 DSDP-RLT}

\begin{align*}
\text{minimize:} & \quad \sum_{c \in S} c_{ij} + c_{ij}R(S^c) + (c_{ai}R(S^a))^2 \\
\text{subject to:} & \quad S_{ij} = (Y_{ij}^* + Y_{ij}^*)W_{ij}T_{ij}^{-2} - Y_{ij}^*W_{ij}
\text{, } \forall (i, j) \in E \quad (2a) \\
S_{ij} = (Y_{ij}^* + Y_{ij}^*)W_{ij} - Y_{ij}^*W_{ij}
\text{, } \forall (i, j) \in E \quad (2b) \\
S_i^a - S_i^d - Y_i^*W_{ii} = \sum_{(i,j) \in E \cap \bar{V}_i} S_{ij}, \quad \forall i \in N \quad (2c) \\
I(W_{ij})\tan\theta_{ij} \leq R(W_{ij}) \leq I(W_{ij})\tan\theta_{ij}, \quad \forall (i, j) \in E \quad (2d) \\
l_{ij}|\mathbf{T}_{ij}|^2 = |(Y_{ij} + Y_{ij}^*)|T_{ij}^2W_{ii} - Y_{ij}(Y_{ij} + Y_{ij}^*)T_{ij}^2W_{ij}
\text{, } \forall (i, j) \in E \quad (2e) \\
\hat{W}_{ij} = |\mathbf{T}_{ij}|^2(\tilde{p}_{ij} + \tilde{q}_{ij}), \quad \forall (i, j) \in E \quad (2f) \\
l_{ij}|\mathbf{T}_{ij}|^2 = |(Y_{ij} + Y_{ij}^*)|T_{ij}^2W_{ij} - Y_{ij}(Y_{ij} + Y_{ij}^*)T_{ij}^2W_{ij}
\text{, } \forall (i, j) \in E \quad (2g) \\
\text{det}(W_{ij}) = 0 \quad \forall W_{ij} \in M(W_c), \forall c \in T(G) \quad (2h) \\
S_i^a \leq S_i^d \leq S_i^a, \quad \forall i \in N \quad (2i) \\
S_i^d \leq W_{ii} \leq S_i^d, \quad \forall i \in N \quad (2j) \\
\theta_i \geq R(W_{ij}) \geq \theta_i, \quad \forall (i, j) \in E \quad (2k) \\
-\theta_i \leq \theta_i - I(W_{ij}) \leq \theta_i, \quad \forall (i, j) \in E \quad (2l) \\
0 \leq l_{ij} = |T_{ij}|/\sqrt{l_{ij}^2}, \quad \forall (i, j) \in E \quad (2m) \\
0 \leq l_{ij} = |T_{ij}|^2/l_{ij}^2, \quad \forall (i, j) \in E \quad (2n) \\
-\sqrt{l_{ij}^2} \leq p_{ij}, q_{ij} \leq \sqrt{l_{ij}^2}, \quad \forall (i, j), (j, i) \in E \quad (2o) \\
MC(l_{ij}, W_{ij}, \tilde{W}_{ij}), \quad \forall (i, j), (j, i) \in E \quad (2p) \\
Sec(\tilde{p}_{ij}, p_{ij}), \quad \forall (i, j), (j, i) \in E \quad (2q)
\end{align*}

\textbf{C. Scaling Up the Determinant-Based Relaxations}

Below are the steps we took to make sure Model (2) can still converge on large-scale instances:

1. The total number of determinant cuts (2j) generated from $3 \times 3$ determinants are capped at 30,000.
2. The RLT constraints on current-magnitudes (2f), (2g), (2h), (2i) are applied on the edge set $E'$ where: $E' = \{(i, j) \in E : |g_{ij}| \leq 10^3, |b_{ij}| \leq 10^3\}$
3. Model (2) is initialized using the solution of Model (1) and the non-convex relationships between the variables.
4. Concave envelopes for $\tilde{p}_{ij}, \tilde{q}_{ij}$, $\forall (i, j), (j, i) \in E$ were not added to the formulation (secant constraints in [2]).
5. Thermal limit constraints (1g), (2e) and the objective function are scaled by $10^{-3}$.  

\textbf{III. Computational Experiments}

Upper bounds were computed for all instances in PGLIB v21.07 [5]. We tested Gravity [6] and PowerModels v0.18.3 [7] with Ipopt 3.12 [8] and linear solver MA27 [9], GridOpt v1.3.7 [10] with the default interior-point solver INLP from the python-package OPTALG [11], and MATPOWER v7.1 [12] which uses the default interior-point solver MIPs. We used a Mac 3.5 GHz 6-Core Intel Xeon E5 with 32 GB of RAM for computing all ACOPF solutions.

Lower bounds for all networks in PGLIB v21.07, with a second order cone relaxation gap greater than $1\%$, were tested. Lower bounds from Model (2) were computed in Gravity [6], using Ipopt 3.12 [8] with linear solver MA27 [9]. The optimality and constraint satisfaction tolerances are set to $10^{-6}$ and bound relax factor to $10^{-8}$. Times reported include lower bound model build time, initialization time and solution time. Lower bounds corresponding to the moment-hierarchy TSSOS [13] were tested in Julia 1.6 using Mosek 9.3 for solving the underlying SDPs. Times include model build time, prepossessing and solution time. Lower bounds are run on a 3.5 GHz Dual-Core Intel Core i7 with 16 GB RAM. In all experiments, we exclude Julia’s just-in-time compile time. All experiments presented here can be reproduced by cloning the following repositories: PES22 and PES22_Largescale.

\textbf{IV. Numerical Results}

\textbf{A. Upper Bounds}

Table I presents the wall-clock computational time for solving the ACOPF problem on the original (typical operating conditions) PGLIB v21.07 library [5]. Figure 1 shows the slow-down of all solvers compared to the Gravity-Ipopt approach. The figure shows that the Python-based tool GridOpt can be up to $13 \times$ slower than an average $3.5 \times$ slow-down. Powermodels, using the same Ipopt solver can be up to $10 \times$ slower than an average $4 \times$ slow-down. Finally, note that Matpower fails on 11 out of 59 instances. When it does converge, Matpower is on average $2.2 \times$ slower with a worst-case slow-down of $6 \times$. Performance profiles depicting the number of instances solved as a function of time is presented on all PGLIB instances in Figures 2, 4 and 5. Note that
scaling the objective function and the thermal limit constraints by $10^{-3}$ helps speed-up Ipopt's convergence by 28% on average as illustrated in Figure 3. All the upper-bound results presented here (except for Figure 3) are without scaling. All the semidefinite programming relaxations used to compute lower bounds use this scaling as discussed in Section II-C.

### B. Lower Bounds

Results of the lower bound runs are given in Table II. We compute the optimality gap using the optimal solution of Model (2) with and without the RLT cuts. For networks with an optimality gap greater than 1% and less than a 1000 buses, we also compute the gap from the optimization-based bound tightening (OBBT) algorithm (SDP-BT) using a small cluster of 8 machines, with a time limit of 3h, as described in [2].

Columns 1 and 2 give the root node gap and computation time in seconds, respectively, for the Determinant-SDP (DSDP) relaxation. Columns 3 and 4 give the root node gap and computation time in seconds, respectively, for the Determinant-SDP relaxation with the addition of the RLT constraints. If OBBT is called, Columns 5 and 6 give the gap and time after bound-tightening.

F denotes that Ipopt failed to converge. T.L. denotes that Ipopt failed to obtain a solution within the maximum time of 1.5h for the relaxation. * denotes that Moser reported problem status as feasible (not optimal). # denotes that Moser reported problem status as unknown. MEM denotes insufficient memory. Gaps in bold indicate instances in which the DSDP approaches close the gap, but the moment hierarchy does not.

For small and medium-scale instances ($\leq 3000$ buses), the DSDP-RLT relaxation closes the gap on 61% of all tested instances. We ran OBBT on 17 out of the remaining 24 open cases, which closed the gap on all instances except two. Thus, the DSDP-RLT, in conjunction with OBBT, can successfully close the gap on 85% on all instances up to 3000 buses in PGLiB.

The DSDP relaxation, without the RLT cuts, is more robust in solving large-scale instances (> 3000 buses). Together, the two DSDP relaxations converge to a lower bound for all instances, except one (case6515_rte_api). In total, the DSDP and DSDP-RLT relaxations, close the gap for 53 out of the 118 open cases in PGLiB.

We were able to test 57 small to medium-scale PGLiB cases with the moment-hierarchy approach, without running into memory issues on our hardware. For these 57 instances, the DSDP-RLT relaxation and the SDP-BT close the gap for 32 and 49 cases, respectively. In contrast, the level 1 and the two levels combined close the gap for 11 and 33 cases, respectively. However, the solution status for several of these problems is “feasible”, and not certified to be optimal.

### V. Conclusion

In this paper, we present a comprehensive computational study on the ACOFP problem, benchmarking state-of-the-art open-source tools. We also present methods and software solutions to compute lower bounds and optimality gaps on large-scale instances. The experiments show that using the general-purpose solver Ipopt along with the modeling language Gravity yields the best performance overall.

### ACKNOWLEDGMENT

We would like to thank Carleton Coffrin for providing us with scripts to run all the Julia experiments.
Fig. 1. Relative Slowdown Factor Compared to Gravity on ACOPF Typical Operating Conditions (TYP) PGLIB instances.

Fig. 2. Performance Profile on Typical Operating Conditions (TYP) PGLIB Instances.

REFERENCES

[1] “Revenue from Sales of Electricity to Ultimate Customers by the Energy Information Administration,” 2020. [Online]. Available: https://www.eia.gov/electricity/annual/html/epa02_03.html

[2] S. Gopinath, H. Hijazi, T. Weisser, H. Nagarajan, M. Yekin, K. Sundar, and R. Bent, “Proving global optimality of acopf solutions,” Electric Power Systems Research, vol. 189, p. 106688, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779620304910

[3] J. Wang, V. Magron, and J. B. Lasserre, “Certifying global optimality of ac-opf solutions via the cs-tssos hierarchy,” 2021.

[4] L. Vandenberghe and M. S. Andersen, “Chordal graphs and semidefinite optimization,” Foundations and Trends® in Optimization, vol. 1, no. 4, pp. 241–433, 2015. [Online]. Available: http://dx.doi.org/10.1561/2400000006

[5] IEEE, “The IEEE PES Task Force on Benchmarks for Validation of Emerging Power System: PGLib optimal power flow benchmarks.” [Online]. Available: https://github.com/power-grid-lib/pglib-opf

[6] H. Hijazi, G. Wang, and C. Coffrin, “Gravity: A mathematical modeling language for optimization and machine learning,” Machine Learning Open Source Workshop (MLOSS) at NeurIPS, 2018.

[7] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, “Powermodels.jl: An open-source framework for exploring power flow formulations,” in Power Systems Computation Conference (PSCC), June 2018, pp. 1–8.

[8] A. Wächter and L. T. Biegler, “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming,” Math. Programming, vol. 106, no. 1, pp. 25–57, 2006.

[9] “HSL, a collection of Fortran codes for large-scale scientific computation. See http://www.hsl.rl.ac.uk/.”

[10] “Gridopt.” [Online]. Available: https://github.com/ttinoco/GRIDOPT

[11] “Optalg.” [Online]. Available: https://optalg.readthedocs.io/en/latest

[12] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 12–19, 2011.

[13] “Tssos.” [Online]. Available: https://github.com/wangjie212/TSSOS
TABLE II

| Instance | Spark | DSDP-RLT | BLPI | RLT-DBL | Sym-DRT | Sym-RT |
|----------|-------|----------|------|---------|---------|--------|
| case1a   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1b   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1c   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1d   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1e   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1f   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1g   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1h   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1i   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1j   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1k   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1l   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1m   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1n   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1o   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1p   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1q   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1r   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1s   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1t   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1u   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1v   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1w   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1x   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1y   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |
| case1z   | 9.8   | 9.8      | 9.8  | 9.8     | 9.8     | 9.8    |

Fig. 3. Scaling the Objective and Thermal Limits Constraints by $10^{-3}$.}

Fig. 4. Performance Profile on Congested (API) PGLIB Instances.}

Fig. 5. Performance Profile on Small Angle Difference (SAD) PGLIB Instances.