Computational Performance of Solution Techniques Applied to the ACOPF

Optimal Power Flow Paper 5

A staff paper by
Anya Castillo
Richard P. O'Neill

January 2013

The views presented are the personal views of the authors and not the Federal Energy Regulatory Commission or any of its Commissioners.

Computational Performance of Solution Techniques Applied to the ACOPF

Optimal Power Flow Paper 5

Anya Castillo and Richard P O'Neill

anya.castillo@ferc.gov; richard.oneill@ferc.gov

February 2013

Abstract:

In the over fifty years of studying and testing the Alternating Current Optimal Power Flow (ACOPF) there has been a lack of rigorous experimental design for benchmarking competing approaches. In an effort to motivate better ACOPF benchmarking and reporting standards for replicable and reliable analysis, we present an experimental framework and statistical methods that are an improvement on current practices. We report numerical results from testing the nonlinear solvers Conopt, Ipopt, Knitro, Minos, and Snopt on various sized test problems in which we apply various mathematically equivalent AC-OPF formulations. We run simulations on numerous recorded starting points that include starting from a previous solution (i.e. hot starts), randomized starting points, and the solution to a linearized model as an initialization. Our experimental results indicate a clear advantage to employing a multistart strategy, which leverages parallel processing in order to solve the ACOPF on large-scale networks for time-sensitive applications.

Disclaimer: The views presented are the personal views of the authors and not the Federal Energy Regulatory Commission or any of its Commissioners.

Table of Contents

1. Introduction 4

2. Experimental Design 5

3. Performance Benchmarking 12

4. Numerical Results 14

5. Discussion 29

Acknowledgements 30

Appendix 31

References 32

1. Introduction

After fifty years of improving solution techniques to solving the ACOPF, it is still considered an extremely difficult nonconvex optimization problem and is still not applied in most market operations and planning. Although the ACOPF simultaneously manages real power demand with voltage and reactive power management, the direct current optimal power flow (DCOPF) which is an linearized approximation of the real power injections has been instrumental in calculating the locational marginal prices (LMPs) for market operations. These clearing prices are therefore based on the shadow price for the real power balancing constraint and are adjusted if the revenues fail to cover bid costs. Even with losses accounted for in the DCOPF, this optimization approach fundamentally neglects voltage and reactive power management. As smarter grid controls and technologies are integrated into the power system, the added value is more quantifiable using the ACOPF. Therefore the ACOPF can lead to better utilization of resources and support price-based competition and welfare maximization for a broad class of market activities.

In the over fifty years of studying and testing on formulations, approximation techniques, algorithmic methods and other modeling aspects of the ACOPF, there has been a lack of rigorous experimental design for benchmarking competing approaches. ACOPF studies published to-date often present a solution technique and claim that the proposed approach is faster, or converges more robustly, compared to alternatives. These claims are made on limited testing and limited reporting of the results. Yet, theoretical results are not sufficient proxies of actual algorithm performance because such general principles often fail to explain real world outcomes (Johnson, 2011).

Prior studies have been difficult to replicate and compare due to inconsistencies in reported metrics and lack of publicly available problem sets; see Castillo and O'Neill (2013) for further details. Therefore future studies could be more reliable in reporting metrics and replicable in benchmarking standards. In this study we present a framework for controlled experimental design and performance benchmarking of solution techniques applied to the ACOPF. We utilize the same hardware and software platforms to test and measure a sample of simulations. We use Dolan-Moré performance profiles for ACOPF benchmarking in order to assess the robustness of a solution technique and the relative speed at which it converges. This approach has been proven to be insensitive to outliers and slight error in the results across many samples (Dolan, 2002). The influencing factors on performance that we consider in this study include: the test problem, formulation, solver, and initialization.

The remainder of this paper is organized as follows: Section 2 addresses design choices in the computational study; Section 3 discusses performance benchmarking techniques that are appropriate for analyzing and comparing computational performance of varying ACOPF solution techniques; Section 4 presents numerical results, and is followed by a brief discussion in Section 5.

2. Experimental Design

In this study we generated a sample of simulations to represent ACOPF solution techniques that are combinatorial in the formulation, test problem, solver, and initialization. The type of formulation, solver, and initialization defines a solution technique. As illustrated in Table 1, the combination of a starting point, which is determined by the initialization type for a given test problem, along with the solution technique, uniquely identifies a simulation.

Table 1. Characteristics of a Solution Technique and Simulation Record

Solver (s)	Formulation (f)	Initialization (i)	Test Problem (p)	Starting Point (n(i,p))
Conopt	PSV	Bθ	118-bus	1 Sample (Bθ, per test problem)
Ipopt	RSV	Uniform	300-bus	100 Samples (Uniform, per test problem)
Knitro	RIV	Hot	2383-bus	100 Samples (Hot, per test problem)
Minos			2736-bus	
Snopt			2746-bus	
			3012-bus	
Solution Technique			3120-bus	

Through this factorial approach, we can observe the significance of each factor, and the interactions among these factors. Therefore we recorded the optimal solution value, the variable state, the convergence status, and the CPU time for each simulation. With this experimental design, we addressed common questions about the performance of a given approach, and fixed the factors irrelevant to the question at hand.

We reduced variance through:

- Applying common tolerances on the commercial solvers
- Underutilizing system processor and memory resources to circumvent bottlenecks
- Testing the solution techniques with the same set of starting points for the sample of simulations

In the remaining subsections, we discuss in detail the types of formulations, test problems, solvers, and initializations applied in this study.

Formulations. The ACOPF originated from Carpentier's reformulation of the economic dispatch problem based upon the Kuhn and Tucker theorem, otherwise known today as the Karush-Kuhn-Tucker (KKT) conditions (Carpentier, 1962). The KKT conditions are necessary and sufficient for a locally optimal solution, but are not sufficient for a solution to the ACOPF to be globally optimal since the problem is nonconvex. We present three mathematically equivalent formulations, the polar power-voltage power flow formulation (PSV), the rectangular power-voltage power flow formulation (RSV), and the rectangular current injection formulation (RIV), of the ACOPF with the following notation.

Notation

Sets		Indices
N	Set of nodes in the network	n,m
Parameters		
g_{nm}	Conductance between nodes n and m	
b_{nm}	Susceptance between nodes n and m	
p^{d}_{n}	Real power load at node n	
q^{d}_{n}	Reactive power load at node n	
p^{min}_{n}	Minimum active power for generation at node n	
p^{max}_{n}	Maximum active power for generation at node n	
q^{min}_{n}	Minimum reactive power for generation at node n	
q^{max}_{n}	Maximum reactive power for generation at node n	
v^{min}_{n}	Minimum voltage magnitude at node n	
v^{max}_{n}	Maximum voltage magnitude at node n	
$v^{r,min}_{n}$	Minimum real part of complex voltage at node n	
$v^{r,max}_{n}$	Maximum real part of complex voltage at node n	
$v^{j,min}_{n}$	Minimum imaginary part of complex voltage at node n	
$v^{j,max}_{n}$	Maximum imaginary part of complex voltage at node n	
Variables		
p_n	Active power generation at node n	
q_n	Reactive power generation at node n	
v_n	Voltage magnitude at node n	
θ_n	Voltage phase angle at node n, where $\theta_{nm} = \theta_n - \theta_m$	
v^{r}_{n}	Real part of complex voltage at node n	
v^{j}_{n}	Imaginary part of complex voltage at node n	

i^r_n	Real part of complex current at node *n*
i^j_n	Imaginary part of complex current at node *n*

The ACOPF is an extension of the power flow problem and is balanced at each bus *n*,*m* of the network *N* through power flows or current injections that constitute Kirchhoff's Laws. The set of phasors representing the complex bus voltages of the network is the steady-state condition and can be expressed in polar or rectangular coordinates, as (v_n, θ_n) or $v^r_n + \boldsymbol{j}\, v^j_n$, respectively. The objective of the ACOPF problem can vary, but is often set to minimize total system cost subject to operational constraints. We express the objective as the sum of quadratic cost functions with cost coefficients c_{2n}, c_{1n}, and c_{0n} for the real power injections p_n at all nodes *n* in *N*:

$$\min \sum_{n \in N} [c_{2n}(p_n)^2 + c_{1n}p_n + c_{0n}] \quad (1)$$

The complex power at each bus is defined as $s_n = p_n + \boldsymbol{j} q_n$, where the reactive power q_n is typically much cheaper than real power p_n and therefore not priced. This objective in equation (1) is equivalent to maximizing market surplus when the real and reactive power demanded (p^d_n, q^d_n) is inelastic.

The constraint set for this study is formulated in polar power-voltage power flows (2a-6a), rectangular power-voltage power flows (2b-6b), or rectangular current injections (2c-6c). The polar power-voltage power flow formulation (PSV) is the approach most commonly applied in the literature and has the following constraint set:

$$v_n \sum_{m \in N} v_m (g_{nm}\cos\theta_{nm} + b_{nm}\sin\theta_{nm}) - p_n + p^d_n = 0,\ n \in N \quad (2a)$$
$$v_n \sum_{m \in N} v_m (g_{nm}\sin\theta_{nm} - b_{nm}\cos\theta_{nm}) - q_n + q^d_n = 0,\ n \in N \quad (3a)$$
$$p^{min}_n \le p_n \le p^{max}_n,\ n \in N \quad (4a)$$
$$q^{min}_n \le q_n \le q^{max}_n,\ n \in N \quad (5a)$$
$$v^{min}_n \le v_n \le v^{max}_n,\ n \in N \quad (6a)$$

The rectangular power-voltage power flow formulation (RSV):

$$v^r_n \sum_{m \in N} (g_{nm}v^r_m - b_{nm}v^j_m) + v^j_n \sum_{m \in N} (g_{nm}v^j_m + b_{nm}v^r_m) - p_n + p^d_n = 0,\ n \in N \quad (2b)$$
$$v^j_n \sum_{m \in N} (g_{nm}v^r_m - b_{nm}v^j_m) - v^r_n \sum_{m \in N} (g_{nm}v^j_m + b_{nm}v^r_m) - q_n + q^d_n = 0,\ n \in N \quad (3b)$$
$$p^{min}_n \le p_n \le p^{max}_n,\ n \in N \quad (4b)$$
$$q^{min}_n \le q_n \le q^{max}_n,\ n \in N \quad (5b)$$
$$(v^{min}_n)^2 \le (v^r_n)^2 + (v^j_n)^2 \le (v^{max}_n)^2,\ n \in N \quad (6b)$$

The rectangular current injection formulation (RIV):

$$i^r_n = \sum_{m \in N} (g_{nm}v^r_m - b_{nm}v^j_m),\ n \in N \quad (2c)$$

$i^j_n = \sum_{m \in N} (g_{nm} v^j_m + b_{nm} v^r_m),\ n \in N$ (3c)
$p^{min}_n \leq (v^r_n i^r_n + v^j_n i^j_n) + p^d_n \leq p^{max}_n,\ n \in N$ (4c)
$q^{min}_n \leq (v^j_n i^r_n - v^r_n i^j_n) + q^d_n \leq q^{max}_n,\ n \in N$ (5c)
$(v^{min}_n)^2 \leq (v^r_n)^2 + (v^j_n)^2 \leq (v^{max}_n)^2,\ n \in N$ (6c)

For each formulation, the equivalent constraints are numbered accordingly; for a more detailed discussion on ACOPF formulations, see (Cain, 2012). Control elements such as phase shifters or variable transformer tap ratios are typically incorporated into the power balancing constraints through modifications to the admittance matrix, also known as the Y-Matrix $Y = G + jB$, where g_{nm} is the conductance and b_{nm} is the susceptance from node n to m. In the PSV approach, (2a) and (3a) are trigonmetric functions. The RSV approach instead has quadratic functions in (2b) and (3b) which results in a finite and constant Hessian for derivative-based solvers. The RIV approach solves for the real (i^r_n) and imaginary parts (i^j_n) of the complex current through linear equality constraints (2c) and (3c). In constraints (4c) and (5c), the relationships $v^r_n i^r_n + v^j_n i^j_n = p_n$ and $v^j_n i^r_n - v^r_n i^j_n = q_n$ are derived similarly to (2a), (3a), (2b), and (3b) through a power-current model. The real and reactive power injection/withdrawal limits are in constraints (4) and (5), respectively. The voltage magnitude at the bus is constrained by lower (v^{min}_n) and upper (v^{max}_n) bounds in (6); the approaches in rectangular coordinates (i.e. (6b) and (6c)) include a non-convex lower bound and convex upper bound.

Test Problems. Table 2 lists the publicly available test problems we use for this study and the corresponding minimal costs according to the MATPOWER parameterizations (Zimmerman, 2011). The MATPOWER parameterizations have no angle difference or thermal line constraints, and furthermore do not include time-varying demand profiles. The models range in the number of generators and loads, but are unsophisticated in generation representation. Generators are parameterized with operational 'box-constraints' having scalar power limits and a

Table 2. Characteristics of Test Problems

Test Problem	Buses	Branches	Generators	Loads	Y-Matrix Density	Observed Minimum Cost ($)
I	118	186	54	99	3.519E-02	129,661
II	300	411	69	199	1.247E-02	719,725
III	2383	2896	323	1822	1.440E-03	1,922,928
IV	2736	3269	206	2011	1.239E-03	1,307,832
V	2746	3279	342	1993	1.234E-03	1,505,109
VI	3012	3572	292	2260	1.119E-03	2,584,227
VII	3120	3693	229	2277	1.079E-03	2,232,988

quadratic cost function on the real power, but do not include further characteristics such as ramp rates, startup/shutdown costs, and detailed reactive power capability curves. Thus we simplify the test problems with multiple generators at a given bus to a single generator per bus by aggregating the generation and determining the weighted average cost curve (i.e. the weight is set by the generator nameplate capacity).

The 118-bus test problem represents portions of the early 1960's American Electric Power System in the US Midwest; the 300-bus test problem was developed in 1993 by the IEEE Test Systems Task Force under Mike Adibi; and the 2736-bus (summer-peak), 2746-bus (winter-peak and winter-offpeak), 3012-bus (winter-peak), and 3120-bus (summer-peak) test problems represent the Polish Network during time periods between 2004-2008. The models vary in connectivity, but are all extremely sparse according to the Y-matrix density. Although these MATPOWER parameterized test problems are inadequate compared to real power market models, they are widely used in academic studies and published on in peer-review journals; see Castillo and O'Neill (2013) for a detailed summary of the test problems used in prior studies.

Solvers. We test five commercial solvers using the General Algebraic Modeling System (GAMS). Without linearizations or convexifications of the ACOPF, nonlinear commercial solvers can only guarantee local optimality and feasibility. Furthermore, algorithms behave differently due to numerous factors, including the size, complexity, and representation of the underlying problem. Below in Table 3 we provide a sense of the representation of each test problem where the varying formulations require a different number of nonlinear instructions and nonzero coefficients.

Table 3. Characteristics of Formulations

Test Problem	Number of	
Formulation	Nonzero Coefficients	Nonlinear Instructions
118-bus		
PSV	2,903	11,391
RIV	4,681	6,158
RSV	3,139	11,447
300-bus		
PSV	6,942	24,671
RIV	11,045	14,411
RSV	7,542	26,227
2383-bus		
PSV	51,750	180,009
RIV	84,387	105,375
RSV	56,516	198,290
2736-bus		
PSV	59,003	209,802
RIV	97,449	120,915
RSV	64,475	230,048
2746-bus		
PSV	59,211	210,462
RIV	97,821	121,419
RSV	64,703	230,828
3012-bus		
PSV	64,748	229,533
RIV	107,089	133,131
RSV	70,772	252,456
3120-bus		
PSV	66,979	237,194
RIV	110,801	137,811
RSV	73,219	261,184

(Table 3 continued)

The number of nonlinear instructions and nonzero coefficients for the different formulations and test problems.

The five commercial solvers included in this study are listed in Table 4. In general, default parameterization was adopted. For Minos, the minor and major damping parameters were set to 0.5 (GAMS, 2012). For both Conopt and Minos, the default memory settings in GAMS were increased in order to solve the larger network problems without compilation errors. The parameterizations are available upon request. In a companion paper, Castillo and O'Neill (2013) performed a more detailed discussion on nonlinear optimization, these commercial solvers, and a historical survey of published algorithmic methods applied to the ACOPF.

Table 4. Characteristics of Commercial Solvers

NLP Solver	Version	Algorithm	Convergence Strategy	Second-Order (Hessian) Info
Conopt	3	Generalized Reduced Gradient	Line Search	Exact & Quasi-Newton
Ipopt	3.10	Interior-Point	Line Search	Exact
Knitro	8.0	Interior-Point/Active-Set	Trust Region	Exact & Quasi-Newton
Minos	5.51	Augmented Lagrangian	Line Search	Quasi-Newton
Snopt	7	Sequential Quadratic Program	Line Search	Quasi-Newton

Initializations. Effective initialization methods can improve convergence. There is a fairly large body of literature on ACOPF problems where the authors only report results from using a flat start[1]. Instead we consider the BTheta (Bθ) start, which is a natural extension to the current injection approach for optimal dispatch. We also consider statistical initialization methods to assess the robustness of a given solution technique to solve the problem from varying starting points. In one of the statistical methods included in this study we determine a set of starting points through uniformly randomizing the voltage magnitude profile. In the other we determine a set of starting points using the optimal solution for a given uncertain demand profile (i.e. hot-start). For each starting point, the power flow constraints solve for $[v_n, \theta_n, v^r{}_n, v^j{}_n, p_n, q_n, i^r{}_n, i^j{}_n]$ in order to initialize the PSV, RSV, and RIV formulations similarly. Therefore the initialization is feasible to the equality constraints (2) and (3) but may be infeasible for the remaining constraint set.

The linearized BTheta start, a type of DCOPF, dispatches real power by assuming unitary voltage magnitude (i.e. $v_n = v_m = 1$ p.u.), negligible resistance (i.e. $g_{nm} = 0$), and $sin\theta_{nm} \approx \theta_{nm}$ approximation in the real power flow constraint (2); therefore in the PSV formulation, constraints (3), (5), and (6) are ignored and a linearized cost function is applied as follows:

$\min \sum_{n \in N} c_{1n} p_n$ (1-lin)
subject to
$\sum_{m \in N} b_{nm}\theta_{nm} - p_n + p^d{}_n = 0,\ n \in N$ (2a-lin)
$p^{min}{}_n \leq p_n \leq p^{max}{}_n,\ n \in N$ (4a)
$\theta^{min}{}_{nm} \leq \theta_{nm} \leq \theta^{max}{}_{nm},\ n,m \in N$ for $k \in K$ (7)

Constraint (7) is included, which limits the angle difference between interconnected nodes to be less than or equal to an absolute difference of 30-degrees. Losses are negligible in this formulation. Since the BTheta formulation is a linear program, the optimal solution to the above problem is the global minimum. Therefore only a single starting point is tested for the BTheta initialization; the solution time for solving the BTheta is negligible.

The uniformly randomized start randomizes the initial voltage magnitude at each node, $v^0{}_n$, via a uniform distribution function $U(.)$ on the lower- and upper-bounds

$v^0{}_n = U(v^{min}{}_n, v^{max}{}_n)\ n \in N.$ (8)

The phase angle at each node is set to zero. Then the real and reactive power injections are determined through the power balancing equations (2) and (3).

[1] In a flat start, the voltage magnitude is initialized to 1 p.u. with zero phasor angle, and the real and reactive power injections from generators are at half-output of the given lower and upper bounds.

Therefore constraints (2), (3), and (6) are satisfied, but the real and reactive power injection constraints (4) and (5) may be violated. We determine 100 starting points using the uniformly randomized initialization.

Lastly, the hot-start is essentially a converged solution of a perturbation to the original problem, which could be comparable to a sequential dispatch in real-time. Therefore this approach initializes the variables with the optimal solution to a perturbed test problem where the real power load at each node is shifted via a uniform distribution function within ± 10% of the parameterized load

$p^{d,0}{}_{n} = U(0.9p^{d}{}_{n}, 1.1p^{d}{}_{n})\ n \in N.$ (9)

The solution at the real power load $p^{d,0}{}_{n}$ determines the variable initializations to the original problem which solves for real power load parameter $p^{d}{}_{n}$. We determine 100 starting points using the hot-start initialization.

3. Performance Benchmarking

We apply statistical measurements that are widely accepted in the optimization community to benchmark ACOPF solution techniques. Prior studies in the optimization community on general nonlinear algorithms have shown that comparing the number of iterations per solution technique is not an effective measure of performance; an iteration of one approach is not necessarily comparable computationally to an iteration of another approach (Bongartz, 1997). Instead, we determine a performance ratio for each simulation, which is the ratio of the simulation CPU time in seconds to the best-recorded reference solution. Since relative performance metrics are normalized, these results are robust against changes in software platform, operating system, and hardware (as long as the same environment is utilized across all simulations). These performance ratios are then analyzed to determine the performance profile and geometric mean of a solution technique in order to relate design choices in the solution techniques to the performance measures.

Performance Ratios*.* Performance ratios are useful in determining relative performance of a given solution technique compared to another, where a solution technique is defined as the combination of solver, formulation, and initialization. In this study we define the following performance ratio

$\tau^{n}{}_{sfip} = t^{n}{}_{sfip}/t^{*}{}_{p}$ (10)

for each simulation as a baseline for comparing results, where the subscripts refer to the solver (*s*), formulation (*f*), initialization (*i*) and test problem (*p*), and the superscripts refer to the starting point n which is a function of the test problem and initialization (i.e. $n = n(i, p)$). The *solution technique* is denoted as (*s*, *f*, *i*). There are 100 uniform starting points, 100 hot starting points and one BTheta starting point for each test problem. This is summarized in Table 1, in the beginning of section II.

For each simulation, we define $t^{n}{}_{sfip}$ as the CPU time in seconds required for it to converge. If the simulation does not converge, then $t^{n}{}_{sfip}$ is set to the maximum convergence runtime (i.e. worst-recorded). The minimum recorded CPU time on a given test problem p is

$t^{*}{}_{p} = \min\{t^{n}{}_{sfip}\!: s \text{ in } S, f \in F, i \in I, n \in N\}$. (11)

This simulation must obtain the minimal cost presented in Table 1. We also refer to $t^{*}{}_{p}$ as the *reference solution* for each test problem, and the performance ratio $\tau^{n}{}_{sfip}$ as the *τ-Ratio* for each simulation instance.

Geometric and Arithmetic Means. We use geometric means of the performance ratios as a way to assess the solution techniques, and the arithmetic means for the raw CPU times in seconds. The geometric mean log-transforms the data, which results in statistical terms that are unaffected by normalization. Also, this approach dampens the effects of outliers. The geometric mean for solution technique (*s*, *f*, *i*) is:

$\underline{\tau}_{sfi} = \exp(\sum_{\mathrm{n,p}} w_p \ln(\tau^{n}{}_{sfip}) / \sum_{\mathrm{p}} w_p)$ (12)

Weights, denoted w_p, are included to weight simulation results by test problem since the test problems vary in size and difficulty; assume $w_p = 1$ on all test problems to determine unweighted metrics. We also calculate the geometric standard deviation for solution technique *s*, *f*, *i* as follows:

$\underline{\sigma}_{sfi} = \exp(\sum_{\mathrm{n,p}} w_p (\ln(\tau^{n}{}_{sfip}) - \ln(\underline{\tau}_{sfi}))^2 / \sum_{\mathrm{p}} w_p)^{1/2}$ (13)

The weights are applied similarly for the arithmetic means. We report the geometric means and standard deviations on the performance ratios, and the arithmetic means and standard deviations on the CPU times in seconds.

Performance Profiles. The performance profiles were initially proposed by Dolan and Moré to evaluate the performance of optimization software on a set of test problems; the authors claim that performance profiling reveals all the major performance characteristics of a given approach (Dolan, 2002). We extend this approach to evaluate varying solution techniques to the ACOPF problem on a test set of medium- to large-scale networks. The performance profile is a distribution function of the performance ratios. For the set of test problems in this study, the performance profile provides insight as to the reliability of the solution technique to find an optimal solution, and the relative speed at which it converges. Dolan and Moré prove that performance profiles are insensitive to changes in results on a small number of problems, as well as being robust against small changes in results across many problems (Dolan, 2002).

Therefore, the performance profile for a given solution technique (s, f, i) is defined as the distribution function ρ_{sfi}: R→[0,1] for

$$\rho_{sfi}(\tau) = \left(\sum_{\mathrm{p}} w_p \,\mathrm{card}\{n \in N : \tau^{n}_{sfip} \leq \tau,\, p \in P\}\right) / \left(\sum_{\mathrm{p}} w_p \,\mathrm{card}\{n \in N : p \in P\}\right) \quad (14)$$

where card{.} is the size of the set and the starting point $n = n(i, p)$ is dependent of the initialization and test problem. The test problems are weighted by w_p, where setting $w_p = 1$ for all test problems results in the unweighted performance profile. At $\rho_{sfi}(\tau)$, all the performance ratios under consideration are within a τ-factor of the best possible ratio. For example, the value $\rho_{sfi}(1)$ is the probability that the solution technique (s, f, i) will "win" over all other approaches on the set of test problems.

Interactions. We consider various levels of interactions amongst the solution techniques (i.e. solver, formulation, initialization) to effectively evaluate the simulation results. We can then relate the solution techniques to the performance measures and assess the performance of a given approach. Therefore, in calculating the means, standard deviations, and performance profiles for solution technique (s, f, i), we consider the full interaction amongst the factors solver, formulation, and initialization; we also analyze the results as partial interactions and no interactions. Partial interactions hold two factors fixed, and aggregates on the third factor. No interactions holds a single factor fixed, and aggregates on the other two factors.

4. Numerical Results

The simulation instances were implemented and solved on the GAMS 23.6.2 platform, Intel Xeon E7458 2.40GHz, 64.GB RAM, 64-bit Windows Server 2008 Enterprise. Each simulation instance was solved sequentially, with a time limit of 20 minutes. We define a simulation instance uniquely by a given starting point and the solution technique, which is the combination of solver, formulation, and initialization method for this study[2]. Likewise, a given starting point is defined by the initialization method (i.e. hot start, BTheta start, uniform start) and various combinations of solver and formulation can be compared on this starting point.

[2] In section 5 we discuss other factors that influence a solution technique, such as solver parameterization.

Below we initially discuss the effects of placing a higher emphasis (through the weighting functions presented in the previous section) on successful convergences for the larger test problems. Then we briefly review the results for both the nonconverging and converging simulations. We then present numerical results on the performance of simulation instances on the set of unique starting points. Since the simulation instances were executed sequentially on a given core and therefore did not contend for computing resources, we assess the potential of a parallel processing strategy. In conclusion, we further detail solution technique results where we aggregate the simulation results based on:

(i) solver, formulation, or initialization ("without interactions")
(ii) solver, formulation, and initialization ("full interaction")
(iii) solver and formulation, solver and initialization, or formulation and initialization ("partial interactions")

N-Squared Weighting Effect. We introduce test problem weights, w_p, to serve as a proxy for the comparative dimension of each test problem. We consider both weights based on:

(1) the number of nodes squared (N-Squared), and
(2) the number of nodes plus branches (N+K).

However, we only report results for the N-squared weights since both methods trend consistently. This is because the weighting methodologies are fairly similar, as shown in Table 5 by the proportional percentage shown in parentheses for each test problem.

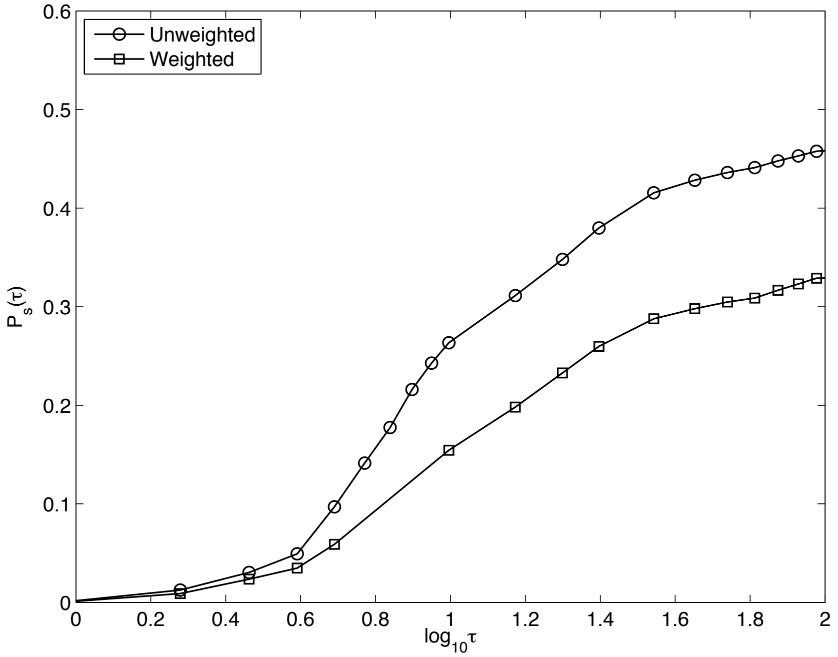
Table 5. Test Problem Weights

Test Problem	N-Squared	N+K	Observed Minimum Cost ($)
118-bus	13924 (0.04%)	304 (0.96%)	129,661
300-bus	90000 (0.23%)	711 (2.24%)	719,725
2383-bus	5678689 (14.33%)	5279 (16.64%)	1,922,928
2736-bus	7485696 (18.90%)	6005 (18.93%)	1,307,832
2746-bus	7540516 (19.03%)	6025 (18.99%)	1,505,109
3012-bus	9072144 (22.90%)	6584 (20.76%)	2,584,227
3120-bus	9734400 (24.57%)	6813 (21.48%)	2,232,988

The value in parentheses is the proportional percentage of the aggregated N-Squared or N+K term, respectively, across all test problems.

Figure 1 displays the aggregated performance profiles across all simulations. For distribution functions with cumulative performances less than 100% (i.e. 1 on the vertical axis), there were nonconverging simulations unaccounted for. The unweighted and weighted performance profiles show that the unweighted profile is optimistically biased towards the smaller test problems. From the weighted performance profile, we see that many of the nonconverging simulations are those that were computed on the larger test problems. The larger networks are closer in size to representing more realistic and practical ACOPF problems. Therefore, in the remainder of our numerical analysis, we will consider only the N-squared weighted statistics, but we will also report the unweighted convergences.

Figure 1. Unweighted versus Weighted Performance Profiles



Overall performance profiles for the aggregated simulations, displayed in terms of both unweighted and N-squared weighted performance ratios.

Overall Nonconvergence Summary. Simulation instances that solved to optimality are reported as successfully converged; no other local optimal solutions on the test problems were found by any of the solvers. The remaining simulations are reported as nonconverged (see Table 6). For determining the performance profiles and τ-ratios in later analysis, the CPU time of the nonconverged instances is set to the maximum time limit of 20 minutes.

The results in Table 6 demonstrate that the solvers generally encounter different nonconvergence issues. Early termination with a feasible solution is due to insignificant improvement in the optimal objective function value. Solvers that terminate with a feasible (practical) solution could be useful in actual operations, while solvers that terminated with an infeasible (impractical) solution may be less useful in actual operations.

Table 6. Simulation Results

Termination	Conopt	Ipopt	Knitro	Minos	Snopt
Exceeded time limit with an infeasible solution	7.9%	22.2%	38.4%	0.0%	23.0%
Exceeded time limit with a feasible solution	6.5%	0.9%	6.7%	0.0%	14.4%
Early termination with an infeasible solution	0.0%	0.4%	0.6%	11.4%	0.0%
Early termination with a feasible solution	31.2%	0.6%	1.8%	2.3%	0.2%
Normal Termination with an infeasible solution	10.5%	3.4%	0.0%	55.9%	1.6%
Lowest Observed Minimal Cost Solution	43.8%	72.5%	52.5%	30.3%	60.8%
Total Percentage of Feasible	81.6%	74.1%	61.0%	32.7%	75.4%

The percentage of the simulations that reported convergent, nonconvergent feasible, and nonconvergent infeasible results for each solver across the test problems.

Overall Convergence Summary. Given an initialization method, Tables 7-9 report the minimum CPU time to convergence across the simulations for each test problem; the solver and formulation that achieved this time is also reported.

According to Table 9 the hot starts achieve lower CPU times out as compared to uniform starts and BTheta starts on each test problem. Therefore the hot starting points set the *reference solution* t^*_p; this *reference solution* indicates the minimum CPU time attained to solve the test problem to optimality whereas nonconverging solutions have a CPU time set to the time limit of 20 minutes.

This reference solution is the basis for constructing the performance ratios (i.e. "τ-Ratios") that are aggregated into *performance profiles* in this study (see Section III). The *performance profiles*, which are essentially cumulative distribution functions, and the weighted geometric mean τ-Ratios are both unit-less measurements and indicate performance that is relative to the *reference solution* reported below in Table 9. Per test problem, the *reference solution* is the fastest recorded CPU time in this study.

Table 7. Uniform Starting Points

Test Problem	Minimum CPU Time (s)	Solver	Formulation
118-bus	0.3	Knitro	RIV
300-bus	0.7	Knitro	RSV
2383-bus	46.2	Minos	PSV
2736-bus	53.8	Ipopt	RIV
2746-bus	41.6	Ipopt	PSV
3012-bus	59.6	Ipopt	PSV
3120-bus	49.5	Knitro	RSV

The minimum CPU time across uniform starts for each test problem.

Table 8. BTheta Starting Points

Test Problem	Minimum CPU Time (s)	Solver	Formulation
118-bus	0.5	Ipopt	PSV
300-bus	1.6	Ipopt	PSV
2383-bus	33.7	Minos	RSV
2736-bus	33.8	Minos	RSV
2746-bus	24.8	Ipopt	PSV
3012-bus	79.1	Minos	RSV
3120-bus	58.7	Ipopt	PSV

The minimum CPU time across BTheta starts for each test problem.

Table 9. Hot Starting Points

Test Problem	t*p, Minimum CPU Time (s)	Solver	Formulation
118-bus	0.1	Conopt	RSV
300-bus	0.3	Conopt	PSV
2383-bus	10.3	Minos	RIV
2736-bus	6.5	Minos	RIV
2746-bus	11.3	Conopt	RSV
3012-bus	8.4	Knitro	RIV
3120-bus	10.1	Conopt	RSV

The minimum CPU time across hot starts for each test problem. Note that the recorded minimum CPU time for hot start is lower than the uniform and BTheta minimum CPU time. Therefore the minimum CPU times by hot start set the *reference solution* t^*_p.

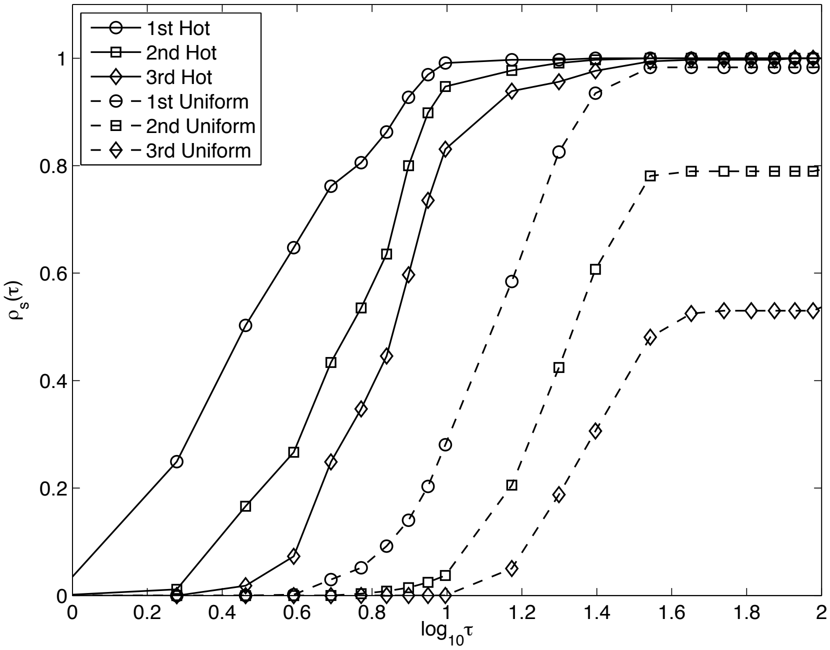
Parallel Processing Model. We first analyze the performance of simulation instances on a given starting point. Overall, there are 100 unique hot starts, 100 unique uniform starts, and 1 unique BTheta start[3] for each test problem; we exclude the BTheta start from this analysis. Thus for each unique hot and uniform starting

[3] The BTheta model is a linear program, where if the linear program has a bounded, feasible region, then the optimal solution is the global solution. Therefore the BTheta start results in a single starting point which is at the global solution to the linear program.

point we record the three lowest CPU times observed for each test problem. Figure 2 aggregates the τ-Ratio of these simulations across the set of test problems.

Figure 2 illustrates that for all unique hot starting points, *at least* three of the fifteen combinations of solver (i.e. Conopt, Ipopt, Knitro, Minos, Snopt) and formulation (i.e. RSV, RIV, PSV) converge to optimality. This result indicates that the 100 unique hot starts for each test problem were solved successfully. On the contrary, the uniform starts indicate the lack of robustness across all combinations of solver and formulation. The performance profiles in Figure 2 indicate that regardless of the solver and formulation combination, all simulations failed on certain uniform starting points.

Figure 2. Lowest CPU Time by Starting Point



The three minimum CPU times on each starting point for both hot and uniform initializations. These performance profiles are N-squared on all test problems.

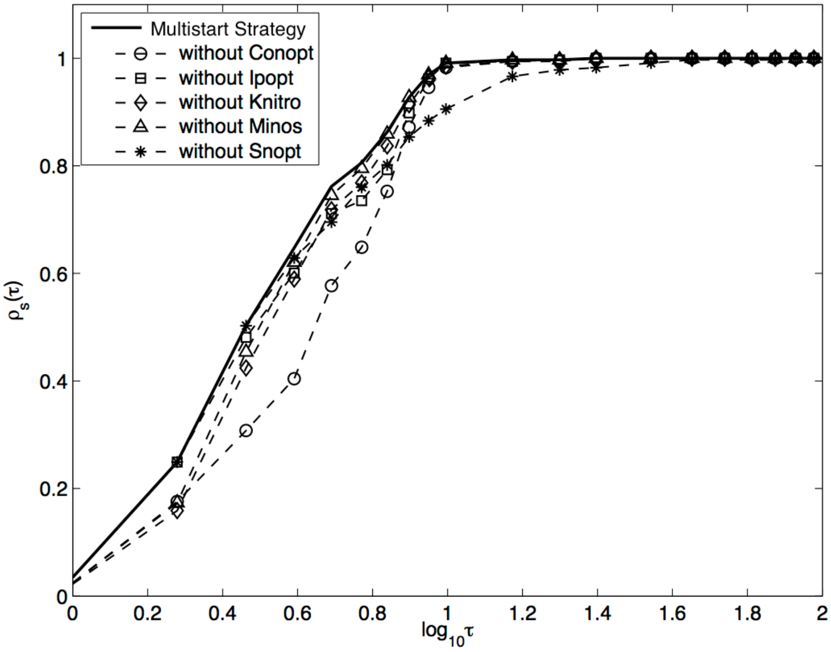
Given that the formulation and initialization are practically supplied for free by the user, we next analyzed whether one solver proved to be more beneficial than another. This is of interest since most commercial solvers require licensing and purchasing agreements. Therefore we assess a multistart strategy[4] across the simulation instances where we run the solvers in parallel to solve a given test problem.

[4] A global optimization procedure that applies a solution technique for numerous starting points on parallel threads/processes and then terminates with increasing confidence in a unique optimal solution.

Since the hot and uniform initializations provide various and numerous starting points, we assess how each solver would perform with the three different approaches to formulating the ACOPF as a nonlinear program. For each solver, all the hot starts across the various formulations would entail a single multistart approach on a given test problem[5]. Thus the multistart strategy is assessed across all available solvers; this strategy could certainly be implemented in a cloud or grid computing environment where computing resources can be scaled ad-hoc.

In Figure 3 the *solid*, *bold* performance profile (i.e. 'Multistart Strategy') illustrates the relative performance to the reference solution (see Table 9) when the overall multistart strategy is applied. Note that the performance profile of the multistart strategy in Figure 3 is equivalent to the performance profile 'Multistart Strategy' in Figure 2; this profile consists of hot starts only since the hot starts resulted in faster times than the uniform starts. The remaining performance profiles in Figure 3 are for the multistart strategy where a single multistart approach for a given solver is excluded. By excluding one solve at a time, we are able to gain some intuition on the marginal impact on the multistart strategy of a particular solver.

Figure 3. Impact of Solver on Parallel Processing Performance



The multistart strategy performance is illustrated above in bold. The remaining performance profiles demonstrate the overall performance when a given solver is excluded

[5] Note that we do not include uniform starting points when considering practical operations due to its inferior performance, as shown in Figure 2.

from the parallel processing. These performance profiles are N-squared on all test problems.

The results in Figure 3 indicate that the most substantial increase in CPU time occurs when Conopt is excluded from the set. Furthermore, Snopt provides significant benefits in decreasing CPU time for achieving higher than 80% convergence probability (i.e. above 0.8 on the vertical-axis). Thus the removal of either Conopt or Snopt from the parallel processing set would dramatically decrease the performance of a multistart strategy. However, note that the removal of any single solver did not result in complete nonconvergence on a starting point; although there may be nonconvergence when excluding multiple solvers from the proposed strategy. Thus the proposed multistart strategy is robust for the given hot starting points included in this study, where certain solvers provide an edge in decreasing execution time on the set of test problems.

This result suggests that each solver provides added benefit to a parallel processing strategy. For example, Table 10 indicates the percentage of hot starting points on which a particular solver attained one of the three minimum CPU times. These percentages are N-squared weighted to emphasize the increased value in simulations that converge to optimality on larger test problems. Given that each solver shared in achieving a minimum CPU time, these statistics evidently indicate that no solver is strictly inferior to the other solvers for the given hot starting points in this study.

Table 10. Overall Weighted Percentage of Best Performance by Solver

	Conopt	Ipopt	Knitro	Minos	Snopt
1st	13%	25%	12%	1%	49%
2nd	32%	18%	17%	12%	21%
3rd	10%	27%	11%	1%	52%

The highest percentage of best performances across all starting points with hot initialization. These percentages are N-squared weighted for each test problem.

Results without Interactions. We analyze results where we aggregate simulation instances by solver, formulation, or initialization method independently. For example, we can assess the influence of the solver on a given simulation instance without considering other factors of the solution technique. Therefore Figures 4 through 6 and Table 11 analyze the solution techniques where the results are aggregated by the factor of interest within the solution technique.

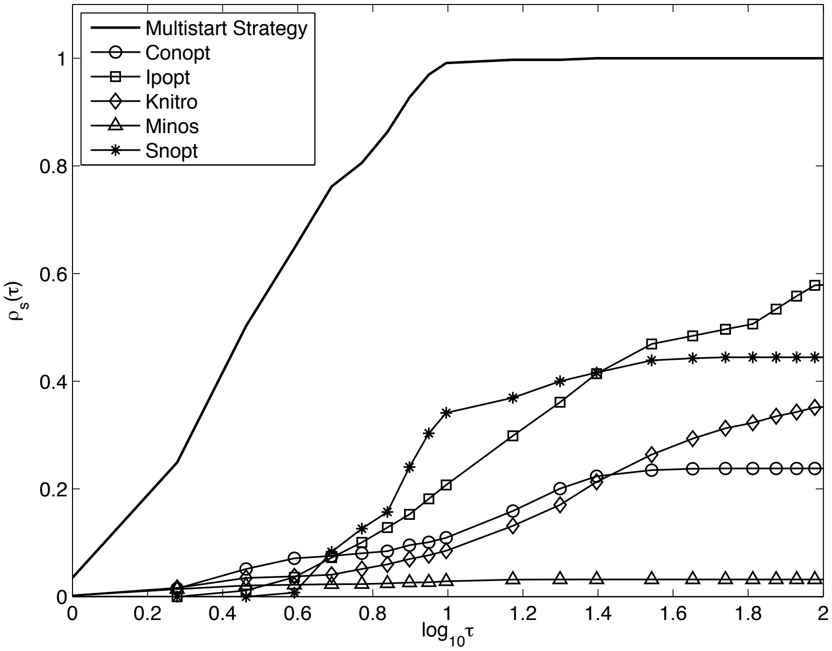
The weighted geometric means in Table 11 indicate that Ipopt is on average approximately 39 times slower than the reference solution (see Table 9), but is still faster overall than the other solvers. Furthermore, Ipopt and Snopt are fairly comparable, both being over three times faster than Minos. Ipopt and Snopt also have approximately three-fourths of the simulations terminating with a feasible, if not optimal, solution. Conopt has both a low weighted arithmetic mean and a low convergence percentage which indicates that the solver has comparably fast nonconvergence times. As opposed to slow-running nonconverging simulations, faster nonconvergence could be useful in freeing up computational resources for

Table 11. No Interactions: Overall Summary

Ranked Solver	Weighted Geometric Mean τ-Ratio (Standard Deviation)	Relative τ-Ratio	Weighted Arithmetic Mean CPU Time in seconds (Standard Deviation)	Unweighted Convergence
Ipopt	38.9 (3.7)	1.00	365.4 (351.0)	3062/4221 (72.5%)
Snopt	40.6 (4.4)	1.05	358.7 (484.0)	2567/4221 (60.8%)
Knitro	67.9 (3.3)	1.75	479.6 (404.4)	2214/4221 (52.5%)
Conopt	73.2 (3.7)	1.88	156.0 (110.0)	1850/4221 (43.8%)
Minos	127.3 (2.2)	3.28	231.0 (395.9)	482/4221 (11.4%)
Ranked Formulation				
RSV	55.8 (4.0)	1.00	249.8 (320.3)	3707/7035 (52.7%)
RIV	56.6 (3.9)	1.01	258.1 (316.4)	3734/7035 (53.1%)
PSV	79.5 (3.2)	1.42	446.5 (474.0)	2734/7035 (38.9%)
Ranked Initialization				
Btheta	31.4 (3.9)	1.00	652.3 (1393.9)	78/105 (74.3%)
Hot	40.9 (4.6)	1.30	399.2 (503.7)	6164/10500 (58.7%)
Uniform	97.9 (2.4)	3.12	233.7 (122.3)	3933/10500 (37.5%)

The relative τ-Ratio is determined as relative to the lowest weighted geometric mean for a given category. The weighted arithmetic mean CPU time includes both converging and nonconverging simulations where nonconverging simulations are assigned a CPU time equal to the time limit of 20 minutes.

Figure 4. Solve Performance Profiles



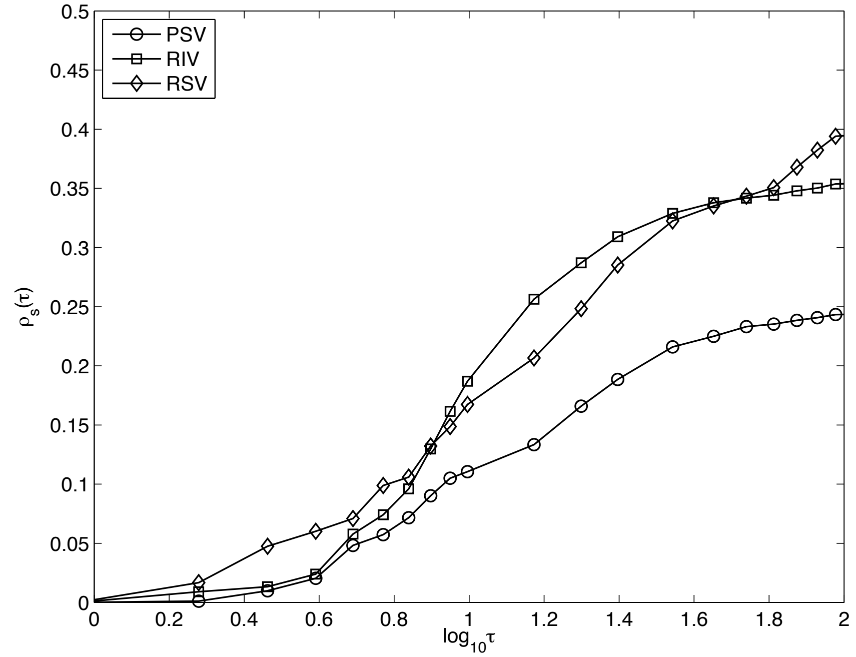
A comparison of the multistart strategy to the solvers, regardless of formulation and initialization used in the solution technique.

queued simulation instances or also could be useful when terminating with a feasible solution. For example although Conopt has a low convergence rate (see Table 11), Conopt has the highest rate for nonconverging with a feasible solution (see Table 6).

Figure 4 compares the multistart strategy performance profile (see Figure 3) to those of each solver regardless of formulation and initialization method applied; there is a clear advantage of running the solution techniques in parallel.

Other interesting results to note: although Conopt and Minos set the reference solution for most test problems (see Table 9), Snopt and Ipopt have more robust performance. Snopt has a 40% probability of converging within 10 times the speed of the reference solution (i.e. $\log_{10}\tau = 1$), and up to a 45% probability of converging within 100 times the speed of the reference solution (i.e. $\log_{10}\tau = 2$). Ipopt has a higher convergence rate than Snopt during the longer runtimes; that is, when Ipopt has been running more than 25 times slower than the reference solution.

Figure 5. Formulation Performance Profiles.

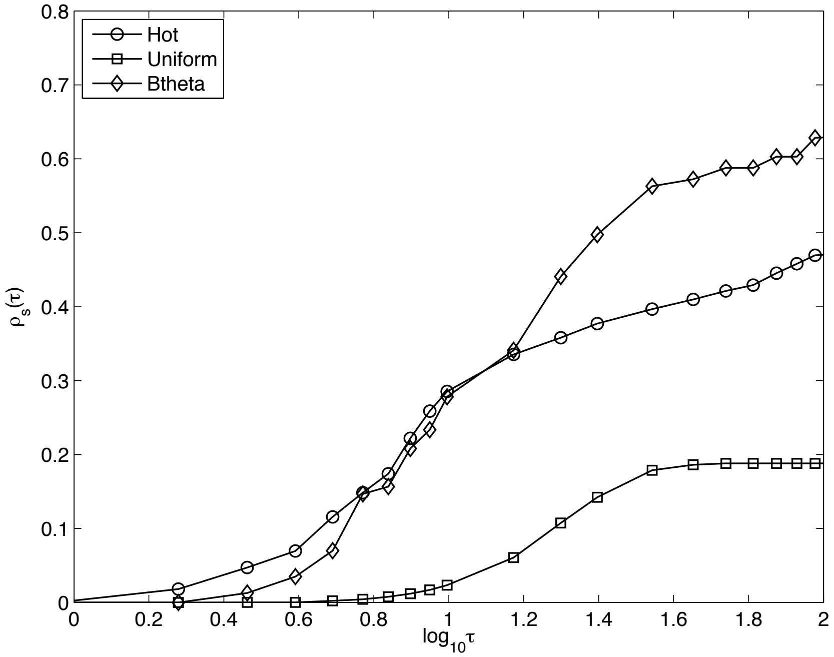


Performance profiles of formulation methods assessed independently of the solver and initialization method used in the solution technique.

Figure 5 indicates that the rectangular formulations (RIV and RSV) have higher convergence and lower CPU time as compared to the polar formulation (PSV) across all solvers and initialization techniques for the set of test problems. Accordingly, table 11 reports that the rectangular formulations are nearly comparable on the test problems, regardless of the solver and initialization used, and also that the rectangular formulations are approximately 40% faster in convergence time compared to the polar.

Figure 6 illustrates that the BTheta and hot initializations out-perform the uniformly randomized starting points. This highlights the significance of selecting appropriate initialization methods for this problem. Furthermore, these results also indicate that even commercial solvers are not robust in solving the given set of test problems when randomized initializations are applied.

Figure 6. Initialization Performance Profiles



Performance profiles of initialization methods aggregated regardless of the solver and formulation in the solution technique.

Results with Full Interactions. We analyze the influence of a given solution technique on the set of starting points. Therefore we aggregate the simulation results based on the solver, formulation, and initialization method. This analysis determines whether certain combinations of solver, formulation, and initialization perform better than others. Table 12 reports the statistics for the top twenty solution techniques with the lowest τ-ratios; these τ-ratios are calculated relative to the reference solution (see Table 9).

Table 12. Full Interactions: Overall Summary

		Weighted Geometric Mean	Relative	Weighted Arithmetic Mean	Unweighted
		τ-Ratio	τ-Ratio	CPU Time in seconds	Convergence
	Ranked Combination	(Standard Deviation)		(Standard Deviation)	
1	Minos, RSV, Btheta	5.5 (1.5)	1.0	53.7 (20.2)	7/7 (100.0%)
2	Snopt, RIV, Hot	7.5 (1.6)	1.4	75.3 (56.8)	700/700 (100.0%)
3	Ipopt, PSV, Btheta	9.0 (2.3)	1.6	102.1 (60.6)	7/7 (100.0%)
4	Ipopt, RIV, Btheta	10.7 (1.9)	1.9	125.3 (83.9)	7/7 (100.0%)
5	Ipopt, RIV, Hot	10.7 (2.5)	2.0	180.3 (215.2)	680/700 (97.1%)
6	Knitro, RSV, Hot	11.3 (3.0)	2.1	181.4 (197.7)	684/700 (97.7%)
7	Minos, PSV, Btheta	13.4 (4.0)	2.4	353.6 (477.9)	6/7 (85.7%)
8	Snopt, RSV, Hot	14.6 (3.8)	2.7	381.2 (531.7)	587/700 (83.9%)
9	Snopt, RSV, Btheta	16.4 (3.5)	3.0	387.0 (2455.7)	6/7 (85.7%)
10	Conopt, RSV, Hot	20.7 (7.4)	3.8	686.3 (25.1)	440/700 (62.9%)
11	Snopt, RIV, Btheta	23.1 (2.9)	4.2	421.9 (2436.0)	6/7 (85.7%)
12	Ipopt, RIV, Uniform	23.3 (2.7)	4.2	384.7 (82.8)	600/700 (85.7%)
13	Conopt, RSV, Btheta	25.4 (2.8)	4.6	377.5 (20.6)	6/7 (85.7%)
14	Snopt, PSV, Btheta	28.5 (4.7)	5.2	662.5 (2858.5)	5/7 (71.4%)
15	Knitro, RSV, Btheta	28.5 (1.7)	5.2	298.2 (172.6)	7/7 (100.0%)
16	Ipopt, PSV, Hot	28.6 (5.0)	5.2	663.3 (489.3)	452/700 (64.6%)
17	Snopt, PSV, Hot	29.5 (5.0)	5.4	687.1 (744.5)	430/700 (61.4%)
18	Knitro, Polar, Btheta	46.3 (2.0)	8.4	569.0 (403.7)	5/7 (71.4%)
19	Ipopt, Polar, Uniform	49.4 (3.4)	9.0	780.7 (181.1)	443/700 (63.3%)
20	Conopt, RSV, Uniform	54.1 (2.9)	9.8	786.3 (40.3)	426/700 (60.9%)

The top twenty solution techniques according to the τ-ratios are reported in Table 12; the remaining ranks are reported in Appendix A.

Figure 7. Solution Technique Performance Profiles

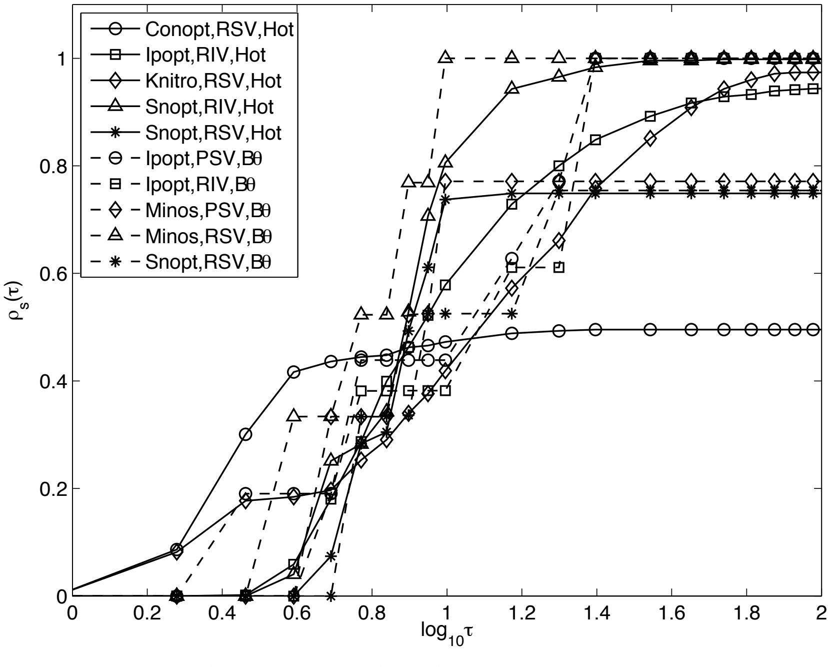


Figure 7 illustrates the performance profiles for the top ten solution techniques that are listed in Table 12.

We observe that four techniques exhibit perfect convergence: Minos-RSV-BTheta, Snopt-RIV-Hot, Ipopt-PSV-BTheta, and Ipopt-RIV-BTheta solve to optimality for the given starting points on all seven of the test problems. The Minos-RSV-BTheta solution technique performed on average within 5.5 times the speed of the reference solution. Comparably, the next in rank, Snopt-RIV-Hot, is 30% slower.

Notably, the Snopt-RIV-Hot solution technique converged perfectly on 700 unique simulation instances (i.e. 100 unique runs per test problem) and performs on average within 7.5 times slower than the reference solution. According to Figure 7, the Snopt-RIV-Hot solution technique has the probability to converge in 80% of the simulation instances where the CPU time is within 10 times the speed of the reference solution; the remaining 20% of simulation instances within 30 times the speed of the reference solution for each test problem. Interestingly, by switching the formulation in Snopt-RIV-Hot from RIV to RSV (with all else being equal), the convergence rate declines by 16% as the execution time doubles.

Table 12 also reports that Ipopt-PSV-BTheta, Ipopt-RIV-BTheta and Ipopt-RSV-BTheta converge to optimality for all recorded simulation instances, but Ipopt-RSV-BTheta had much higher performance ratios and therefore did not make the top twenty list (see Appendix A). Notably Knitro-RSV-Hot and Ipopt-RIV-Hot solution techniques converge 97.7% and 97.1%, respectively; but, the Ipopt-RIV-Hot approach is slightly faster.

Results with Partial Interactions. The analysis in this section aggregates the simulation results based on two of the three factors comprising a solution technique. Tables 13-15 present additional results when the simulations are analyzed on two factors (i.e. solver and formulation; solver and initialization; formulation and initialization). We include these tables to demonstrate the difference in performance and convergence characteristics when the results are partially aggregated.

Note that the lowest τ-Ratios occur with full interactions, as illustrated in Table 12. In fact, the lowest τ-Ratios reported in Table 12 is at least 10 times faster than the τ-Ratios reported in Tables 13-15. This result corroborates the assessment that a multistart strategy that includes the varying solution techniques would out perform strategies where only one solver and one formulation, as an example, is applied to solve the test problem.

Table 13. Solution Technique Aggregated by Solver and Initialization

	Weighted Geometric Mean	Relative	Weighted Arithmetic Mean	Unweighted
	τ-Ratio	τ-Ratio	CPU Time in seconds	Convergence
Ranked Combinations	(Standard Deviation)		(Standard Deviation)	
Snopt, Hot	14.8 (3.9)	1.0	404.2 (601.1)	1717/2100 (81.8%)
Minos, Btheta	17.9 (4.6)	1.2	148.3 (303.1)	16/21 (76.2%)
Ipopt, Btheta	18.7 (3.2)	1.3	319.2 (366.7)	21/21 (100.0%)
Snopt, Btheta	22.1 (3.8)	1.5	1924.7 (2659.9)	17/21 (81.0%)
Ipopt, Hot	29.2 (4.0)	2.0	475.2 (444.2)	1748/2100 (83.2%)
Knitro, Hot	44.0 (3.9)	3.0	675.1 (496.6)	1405/2100 (66.9%)
Ipopt, Uniform	52.2 (3.3)	3.5	256.1 (158.0)	1293/2100 (61.6%)
Conopt, Hot	53.4 (5.0)	3.6	99.6 (112.7)	992/2100 (47.2%)
Knitro, Btheta	57.8 (2.4)	3.9	683.4 (458.6)	13/21 (61.9%)
Conopt, Btheta	71.6 (3.0)	4.9	18598 (31.9)	11/21 (52.4%)
Conopt, Uniform	100.4 (2.3)	6.8	212.1 (72.7)	847/2100 (40.3%)
Knitro, Uniform	105.0 (2.2)	7.1	282.0 (49.0)	796/2100 (37.9%)
Snopt, Uniform	112.4 (2.0)	7.6	297.5 (91.2)	833/2100 (39.7%)
Minos, Hot	113.6 (2.7)	7.7	341.9 (525.2)	302/2100 (14.4%)
Minos, Uniform	145.6 (1.4)	9.9	121.0 (117.3)	164/2100 (7.8%)

Performance metrics of the simulation instances aggregated regardless of the formulation.

Table 14. Solution Technique Aggregated by Solver and Formulation

	Weighted Geometric Mean	Relative	Weighted Arithmetic Mean	Unweighted
	τ-Ratio	τ-Ratio	CPU Time in seconds	Convergence
Ranked Combinations	(Standard Deviation)		(Standard Deviation)	
Ipopt, RIV	15.8 (2.8)	1.0	163.7 (163.2)	1287/1407 (91.5%)
Knitro, PSV	24.8 (3.7)	1.6	211.4 (152.9)	1152/1407 (81.9%)
Snopt, RIV	32.9 (4.6)	2.1	195.5 (230.1)	906/1407 (64.4%)
Conopt, RSV	33.4 (5.3)	2.1	93.6 (70.4)	872/1407 (62.0%)
Ipopt, PSV	37.3 (4.3)	2.4	365.2 (383.6)	902/1407 (64.1%)
Snopt, PSV	44.4 (4.0)	2.8	541.7 (640.8)	868/1407 (61.7%)
Snopt, RSV	45.9 (4.4)	2.9	338.9 (422.4)	793/1407 (56.4%)
Conopt, RIV	86.2 (2.9)	5.5	199.2 (98.8)	609/1407 (43.3%)
Ipopt, RSV	99.7 (1.8)	6.3	567.3 (338.2)	873/1407 (62.1%)
Knitro, PSV	100.4 (2.3)	6.4	562.2 (411.1)	570/1407 (40.5%)
Minos, RIV	103.3 (3.2)	6.6	66.7 (85.7)	440/1407 (31.3%)
Knitro, RIV	125.5 (1.7)	8.0	665.2 (430.3)	492/1407 (35.0%)
Conopt, PSV	136.0 (1.7)	8.6	175.2 (124.4)	369/1407 (26.2%)
Minos, PSV	140.0 (1.6)	8.9	588.4 (508.3)	25/1407 (1.8%)
Minos, RSV	142.8 (1.5)	9.1	37.9 (111.3)	17/1407 (1.2%)

Performance metrics of the simulation instances aggregated regardless of the initialization method.

Table 15. Solution Technique Aggregated by Formulation and Initialization

Ranked Combinations	Weighted Geometric Mean τ-Ratio (Standard Deviation)	Relative τ-Ratio	Weighted Arithmetic Mean CPU Time in seconds (Standard Deviation)	Unweighted Convergence
RSV, Btheta	21.4 (3.2)	1.0	543.9 (1230.0)	33/35 (94.3%)
PSV, Btheta	29.7 (4.2)	1.4	788.6 (1650.5)	25/35 (71.4%)
RIV, Hot	32.3 (4.5)	1.5	273.6 (419.6)	2363/3500 (67.5%)
RSV, Hot	33.1 (4.8)	1.6	283.1 (410.8)	2335/3500 (66.7%)
RIV, Btheta	49.0 (3.7)	2.3	624.4 (1249.4)	20/35 (57.1%)
PSV, Hot	64.1 (3.9)	3.0	641.0 (573.3)	1466/3500 (41.9%)
RSV, Uniform	95.0 (2.4)	4.5	213.6 (138.0)	1339/3500 (38.3%)
RIV, Uniform	99.2 (2.4)	4.7	238.8 (86.9)	1351/3500 (38.6%)
PSV, Uniform	99.5 (2.4)	4.7	248.7 (132.9)	1243/3500 (35.5%)

Performance metrics of the simulation instances aggregated regardless of the solver applied.

5. Discussion

We argue that future experimental work needs to be more replicable and reliable. For example, this study presents an experimental framework and statistical methods that are an improvement on current practices for ACOPF benchmarking and reporting standards. The computational results are often problem dependent and therefore it is difficult to extrapolate the reported performance metrics to larger test sets without further testing.

The experiments indicate that when solving the ACOPF, formulating the mathematical problem in rectangular coordinates versus polar coordinates results in faster computational performance and a higher rate of simulations instances that converge to an optimal solution. Furthermore, the BTheta and hot initialization methods outperformed the uniform initialization and are most likely to be available in actual operations. The uniform initialization provided information on whether the solution technique was robust, that is, converges to a solution from the starting point.

Out of the commercial solvers tested, the experiments suggest that Ipopt and Snopt are both robust and relatively fast in determining the lowest system cost across converged simulations. We adopt mostly default settings for the commercial solvers; therefore we acknowledge that customizing the solver options instead of using the default parameters could result in a very different outcome. For example, setting "bar_directinterval=0" in Knitro enforces direct conjugate gradient steps, and therefore improves how the algorithm handles nonconvexities; for a more detailed discussion on algorithmic methods (see Castillo and O'Neill, 2013). Furthermore, other parameterizations, such as providing the solvers with the active-set information from the hot initialization, would result in increased performance. Therefore, aspects on how the solvers are parameterized and initialized should be considered carefully when customizing general nonlinear solvers for specific applications of the ACOPF.

A useful conclusion is the clear advantage of employing a multistart strategy. In a multistart strategy, various solution techniques are executed in parallel processes in order to solve the ACOPF on a given network. In a cloud or grid computing environment where computing resources can be scaled ad-hoc, this approach would enable an implementation strategy that involves various solvers, solver parameterizations, initialization approaches, and mathematical representations or formulations of the ACOPF problem. Our experimental results show that the lowest system cost could be attained in less than 20 minutes on networks with as many as 3,120 buses from any hot starting point. Therefore a multistart strategy could ensure the robustness in determining the lowest system cost and also increase the probability of determining this operating point, or a feasible and near-optimal operating point, in much faster than 20 minutes.

Therefore our findings can provide some initial indication of useful multistart strategies to solving the ACOPF on large-scale networks. A follow-up study would be needed to address how much solver customization affects performance, why certain solution techniques behave better than others, and whether such experimental results can provide more guidance in honing current ACOPF research endeavors.

Acknowledgements

We would like to acknowledge developers Arne Stolbjerg Drud, Michael Saunders, and Richard Waltz of the tested nonlinear solvers for their insights and recommendations on custom parameterization for solving the ACOPF in future work.

Appendix

Table A. Bottom Twenty-Five: Full Interactions

		Weighted Geometric Mean	Relative	Weighted Arithmetic Mean	Unweighted
		τ-Ratio	τ-Ratio	CPU Time in seconds	Convergence
	Ranked Combination	(Standard Deviation)		(Standard Deviation)	
21	Knitro, RSV, Uniform	54.3 (2.8)	9.9	765.9 (70.9)	461/700 (65.9%)
22	Conopt, RIV, Hot	55.6 (3.6)	10.1	883.3 (26.0)	375/700 (53.6%)
23	Snopt, PSV, Uniform	67.3 (2.6)	12.2	868.8 (155.2)	433/700 (61.9%)
24	Ipopt, RSV, Btheta	68.0 (1.8)	12.4	730.3 (372.7)	7/7 (100.0%)
25	Knitro, PSV, Hot	70.0 (2.9)	12.7	893.6 (447.3)	394/700 (56.3%)
26	Minos, RIV, Hot	73.4 (4.6)	13.3	1131.4 (90.0)	281/700 (40.1%)
27	Minos, RIV, Btheta	78.1 (3.1)	14.2	1084.9 (20.8)	3/7 (42.9%)
28	Ipopt, RSV, Hot	81.0 (1.7)	14.7	817.2 (309.7)	616/700 (88.0%)
29	Conopt, RIV, Btheta	99.6 (2.5)	18.1	1108.4 (29.9)	3/7 (42.9%)
30	Knitro, RIV, Hot	107.6 (2.0)	19.6	1110.2 (325.8)	327/700 (46.7%)
31	Ipopt, RSV, Uniform	123.2 (1.8)	22.4	1223.8 (151.5)	250/700 (35.7%)
32	Conopt, PSV, Hot	131.9 (1.9)	24.0	1291.3 (167.1)	177/700 (25.3%)
33	Conopt, RIV, Uniform	133.5 (1.5)	24.3	1270.4 (19.3)	231/700 (33.0%)
34	Minos, PSV, Hot	139.5 (1.6)	25.4	1305.5 (454.7)	13/700 (1.9%)
35	Conopt, PSV, Uniform	140.1 (1.4)	25.5	1307.1 (54.3)	190/700 (27.1%)
36	Minos, RSV, Hot	143.1 (1.5)	26.0	1317.6 (132.7)	8/700 (1.1%)
37	Minos, PSV, Uniform	143.7 (1.5)	26.1	1320.3 (134.6)	6/700 (0.9%)
38	Conopt, PSV, Btheta	145.1 (1.3)	26.4	1326.6 (14.6)	2/7 (28.6%)
39	Snopt, RSV, Uniform	145.1 (1.2)	26.4	1326.6 (15.2)	200/700 (28.6%)
40	Snopt, RIV, Uniform	145.3 (1.2)	26.4	1326.6 (23.1)	200/700 (28.6%)
41	Minos, RIV, Uniform	145.8 (1.3)	26.5	1327.9 (29.8)	156/700 (22.3%)
42	Knitro, PSV, Uniform	145.8 (1.2)	26.5	1327.2 (13.0)	171/700 (24.4%)
43	Knitro, RIV, Uniform	146.2 (1.2)	26.6	1327.4 (10.6)	164/700 (23.4%)
44	Knitro, RIV, Btheta	146.5 (1.2)	26.6	1327.4 (53.2)	1/7 (14.3%)
45	Minos, RSV, Uniform	147.2 (1.3)	26.8	1330.1 (85.3)	2/700 (0.3%)

References

Mary B. Cain, Richard P. O'Neill, Anya Castillo, "History of Optimal Power Flow and Formulations," Federal Energy Regulatory Commission Staff Technical Paper, December 2012 http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-1-history-formulation-testing.pdf

A. Castillo, R.P. O'Neill. "Survey of Approaches to Solving the ACOPF." Federal Energy Regulatory Commission Staff Technical Paper, 2013. http://www.ferc.gov/industries/electric/indus-act/market-planning/opf-papers/acopf-4-solution-techniques-survey.pdf

I. Bongartz, A.R. Conn, N.I.M. Gould, M.A. Saunders, and P.L. Toint. "A Numerical Comparison Between the LANCELOT and Minos Packages for Large-Scale Constrained Optimization," Council for the Central Laboratory of the Research Councils, 1997.

J. Carpentier. "Contribution a. 'l'etude du dispatching economique." Bull. Soc. Francaise Elect., Vol. 3, Pp. 431-447, 1962.

E.D. Dolan and J.J. Moré. "Benchmarking optimization software with performance profiles." Mathematical Programming, vol. 91, no. 2, pp. 201-213, 2002.

GAMS, "Minos Solver Manual." 2012 http://www.gams.com/dd/docs/solvers/minos.pdf,.

D.S. Johnson. "A Theoreticians guide for experimental analysis of algorithms." in Proceedings of the 5th and 6th DIMACS Implementation Challenges, 2002.

J.N. Hooker. "Testing heuristics: We have it all wrong." Journal of Heuristics, Vol. 1, Pp. 33-42, 1996.

B. Mourrain, S. Schaefer, G. Xu. "Advances in Geometric Modeling and Processing." In Proceedings of the 6th International Conference on Geometric Modeling and Processing, 2010.

C.C. McGeoch. "Toward an Experimental Method for Algorithm Simulation." INFORMS Journal on Computing. Vol. 8, No. 1, Winter 1996.

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.