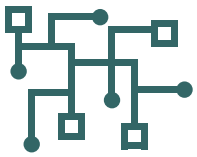

ISE

Industrial and
Systems Engineering

Hybrid Methods in Solving Alternating-Current Optimal Power Flows

Jie Liu[1], Jakub Mareček[2], and Martin Takáč[1]

[1]Lehigh University [2]IBM Research Dublin, Ireland

ISE Technical Report 15T-013

LEHIGH
UNIVERSITY

Hybrid Methods in Solving Alternating-Current Optimal Power Flows

Jie Liu
Lehigh University
Bethlehem, PA, USA

Jakub Marecek
IBM Research
Dublin, Ireland

Martin Takac
Lehigh University
Bethlehem, PA, USA

***Abstract*—Optimisation problems in power systems employing alternating-current models of power flows have driven much recently interest in (convergent hierarchies of) convex relaxations for polynomial optimisation problems. Readily available second-order methods for solving the convex relaxations on real-world large-scale power systems often fail to perform even a single iteration within reasonable run-times. First-order methods have much lower per-iteration computational and memory requirements, but require many more iterations than second-order methods to converge within the same accuracy. We hence study means of switching from first-order methods for solving the convex relaxation to Newton method working on the original non-convex problem, which would allow for convergence under the same conditions as in solvers for the convex relaxation, but with an improved rate of convergence. We illustrate our approach on the alternating current power flows (ACPF) and alternating current optimal power flows (ACOPF).**

***Index Terms*—α-β Theory, Numerical Analysis (Mathematical programming), Optimization, Power system analysis computing**

I. Introduction

Alternating-current optimal power flow problem (ACOPF, [16]) is one of the best known non-convex non-linear optimisation problems, studied extensively since 1960s. Early work focused on applications of Newton method to the non-convex problem, but no guarantees could be given as to its global convergence. More recently, Lavaei and Low [15] have shown that a semidefinite programming (SDP) relaxation produces global optima under some conditions and Ghaddar et al. [8] have shown that a certain hierarchy of SDP relaxations converges asymptotically to the global optimum under some mild conditions. See [16], [17], [21] for further references.

Leading second-order methods for solving the SDP relaxations, such as SeDuMi [24], SDPA [26], or SDPT3 [25], often converge within dozens of iterations on SDP relaxations of even the largest available instances available, but the absolute run-time and memory requirements of a single iteration may be prohibitively large. This has motivated much research into alternative, elaborate relaxations, often exploiting tree-like structures. As an alternative, Marecek and Takac [18] presented a simple Lagrangian relaxation, which is equivalent to that of Lavaei and Low [15], but can be solved within 2 seconds for instances up to 300 buses using first-order methods, which is comparable to the present-best methods without global convergence guarantees, such as [27]. Unfortunately, the iteration complexity of first-order methods remains a challenge, as the instances grow.

To address the challenge, we introduce novel means of combining solvers working on the relaxation and solvers working on the non-convex problem. We employ the first-order method in solving the convexification of [18], until we can guarantee local convergence of Newton method on the non-convex Lagrangian of the problem, possibly considering some regularisation [18]. The guarantee specifically says that when we stop the computation at the point z_0, we know that a Newton method or a similar algorithm with quadratic rate of convergence will generate sequence z_i to the optimum z^*, i.e.

$$\|z_i - z^*\| \leq (1/2)^{2^i - i} \|z_0 - z^*\|. \tag{1}$$

This could be seen as means of on-the-fly choice of the optimisation algorithm, which preserves the global convergence guarantees associated with convexification, whilst improving the convergence rate.

II. Related Work

There is a rich history of work on the convergence of Newton method. We present only a brief overview. Let X be a Banach space, X^* its dual, F a functional on X, and ∇F a Fréchet derivative of F. Further, define the open and closed balls at $x \in X$ as:

$$B(x,r) = \{y \in X;\ \|x-y\| < r\} \text{ and} \tag{2a}$$
$$B[x,r] = \{y \in X;\ \|x-y\| \leqslant r\}, \tag{2b}$$

respectively. Kantorovich's theorem can be stated as follows:

Theorem 1 ([10]). *Let X, Y be Banach spaces, $C \subseteq X$ and $F : C \to Y$ a continuous function, continuously differentiable on $\mathrm{int}(C)$. Take $x_0 \in \mathrm{int}(C)$, L, $b > 0$ and suppose that*

1) $\nabla F(x_0)$ *is non-singular,*
2) $\|\nabla F(x_0)^{-1} [\nabla F(y) - \nabla F(x)]\| \leq L\|x-y\|$ *for any $x, y \in C$,*
3) $\|\nabla F(x_0)^{-1} F(x_0)\| \leq b$,
4) $2bL \leq 1$.

If $B[x_0, t_*] \subset C$, then the sequences $\{x_k\}$ generated by Newton method for solving $F(x) = 0$ with starting point x_0,

$$x_{k+1} = x_k - [\nabla F(x_n)]^{-1} F(x_k), \qquad k = 0, 1, \cdots, \tag{3}$$

is well defined, is contained in $B(x_0, t_*)$, converges to a point $x_* \in B[x_0, t_*]$, which is the unique zero of F in $B[x_0, t_*]$, and

$$\|x_* - x_{k+1}\| \leq \frac{1}{2}\|x_* - x_k\|, \qquad k = 0, 1, \cdots. \tag{4}$$

Moreover, if $2bL < 1$, then

$$\begin{aligned}\|x_* - x_{k+1}\| &\leq \frac{1-\theta^{2^k}}{1+\theta^{2^k}} \frac{L}{2\sqrt{1-2bL}}\|x_* - x_k\|^2 \\ &\leq \frac{L}{2\sqrt{1-2bL}}\|x_* - x_k\|^2, \quad k = 0, 1, \cdots,\end{aligned} \tag{5}$$

where $\theta := t_*/t_{**} < 1$, and x_* is the unique zero of F in $B[x_0, \rho]$ for any ρ such that

$$t_* \leq \rho < t_{**}, \qquad B[x_0, \rho] \subset C, \tag{6}$$

where

$$t_* := \frac{1-\sqrt{1-2bL}}{L}, \qquad t_{**} := \frac{1+\sqrt{1-2bL}}{L}. \tag{7}$$

Under the additional assumption $2bL < 1$, one also obtains quadratic convergence uniqueness of zero x_* of F in $B(x_0, t_{**})$, whenever $B(x_0, t_{**}) \subset C$.

This suggests that whenever $\nabla H(x) + \nabla H(x)^T \succ 0$ for all $x \in B[x_0, t_*]$ or all $x \in C$, we get a monotone operator. Traditionally, it has been assumed that testing a property across the closed ball $B[x_0, t_*]$ or C is difficult. Dvijotham et al. [6], [7] showed, however, that it can be cast as a certain non-convex semidefinite optimisation problem. Let us introduce symmetrisation $\mathrm{Sy}(M) = \frac{M+M^T}{2}$ for $M \in \mathbb{R}^{n\times n}$ and let $\mathcal{C}$ be a compact convex set in $\mathbb{R}^n$ such that $\exists W \in \mathbb{R}^{n\times n}$: $\mathrm{Sy}\left(W \nabla F(V)\right) \succ 0 \quad \forall V : V^c \in \mathcal{C}$. When one considers $x \in \mathcal{C}$ such that $\langle F_W(x), y - x\rangle \geq 0 \quad \forall y \in \mathcal{C}$, one distinguish between the cases of, first, that $\mathcal{C}$ contains a unique solution, and second, that there are no solutions in $\mathcal{C}$.

In the case of alternating-current power flows (ACPF), let us consider the usual instance defined by:

- the graph $G = (V, E)$, where $V, |V| = n$ is partitioned into pv, pq, and $\{\mathcal{S}\}$ slack "buses", and adjacent buses $(i, j) \in E$ are denoted $i \sim j$, and
- the admittance matrix $Y \in \mathbb{C}^{n\times n}$, with $G := \mathrm{Re}(Y)$, $B := \mathrm{Im}(Y)$
- active and reactive injection P_i and Q_i at the bus $i \in \mathrm{pq} \cup \{\mathcal{S}\}$.

[6] define the power-flow operator $F : \mathbb{R}^{2n} \mapsto \mathbb{R}^{2n}$ in terms of complex voltages $V_i = V_i^x + \imath V_i^y, i \in V$ with V^c, with V^c stacked as $V_i^c = V_i^x, V_{n+i}^c = V_i^y$:

$$\begin{aligned}[F\{V^c\}]_i &= G_{ii}\left\{(V_i^x)^2 + (V_i^y)^2\right\} && \text{(8a)}\\ &\quad - \sum_{j\sim i} B_{ij}\left\{V_i^y V_j^x - V_i^x V_j^y\right\} \\ &\quad - \sum_{j\sim i} G_{ij}\left\{V_i^x V_j^x + V_i^y V_j^y\right\} - P_i, i \in V \\ [F\{V^c\}]_{n+i} &= B_{ii}\left\{(V_i^x)^2 + (V_i^y)^2\right\} && \text{(8b)}\\ &\quad + \sum_{j\sim i} B_{ij}\left\{V_i^x V_j^x + V_i^y V_j^y\right\} \\ &\quad + \sum_{j\sim i} G_{ij}\left\{V_i^y V_j^x - V_i^x V_j^y\right\} - Q_i, i \in \mathrm{pq} \\ [F\{V^c\}]_{n+i} &= (V_i^x)^2 + (V_i^y)^2 - v_i^2, i \in \mathrm{pv} && \text{(8c)}\end{aligned}$$

and show one can probe the boundary of set $\mathcal{C}$ by varying the objective of:

$$\begin{aligned}\max_{W} \min_{z\in\mathbb{R}^n, V\in\mathbb{C}^n} \quad & z^T \mathrm{Sy}(W \nabla F\{V\}) z && \text{(9a)}\\ \text{s.t. } & \Re V_i V_j^* \geq \gamma_{ij} \max\{|V_i|^2, |V_j|^2\} && \text{(9b)}\\ & |V_i|^2 = v_i^2, i \in \mathrm{pv} && \text{(9c)}\\ & V_0 = 1 && \text{(9d)}\\ & z^T z = 1 && \text{(9e)}\\ & \|W\|_{\mathrm{nuc}} \leq 1. && \text{(9f)}\end{aligned}$$

We refer to [6] for the complete details. Notice, however, first, that the objective (9a) is non-convex; second, that one may need to solve an infinite number of non-convex non-linear optimization problems (9a–9f) to obtain the complete description of the domain of monotonicity; third, that the approach does not extend easily to polynomial inequalities; and finally, that in ACOPF, one needs to distinguish between domains of monotonicity of local and global optima.

To summarise: Traditionally, the convergence of Newton method could be guaranteed by the non-constructive arguments of the theorem of Kantorovich. Alternatively, one could employ the approach of Dvijotham et al. [6], [7], albeit at a computational cost possibly higher than that of solving ACOPF. Neither approach seems completely satisfactory.

III. Our Approach

We present an alternative approach, based on α-β theory (also known as point estimation theory) of Smale [23] and recently developed insights [14] into the availability and strength of the many possible Lagrangian relaxations of a polynomial optimisation problem. This allows us to decide whether a point is within the domain of monotonicity with respect to a particular Lagrangian, knowing only the Lagrangian and its derivatives at a point, without solving additional optimisation problems. We present two simple illustrations, prior to presenting the main result.

A. First Illustration

First, let us illustrate this approach on the example of a univariate function f, studied by Chen [4]. He shows that under mild assumptions, for each second-order method, there exists a constant t, such that we can switch to the second-order method, whenever we have z_0 such that:

$$(\gamma_f(z_0)+1)\left|\frac{f(z_0)}{\nabla f(z_0)}\right| < t \text{ where} \tag{10a}$$

$$\gamma_f(z) := \sup_{k>1}\left|\frac{f^{(k)}(z)}{k!\nabla f(z)}\right|^{1/k-1} \tag{10b}$$

and $t \approx 0.132301$ for the Newton method. Once again we refer to [4] for this result and the mild assumptions.

Next, we consider a variant of the α-β theory for a system of polynomials, as in [22]. Let us have $f : R^N \to R^m$, which is analytic. Let $x \in R^N$ be such that the derivative

$$\nabla f(x) : R^N \to R^m$$

is surjective and the Moore–Penrose inverse

$$[\nabla f(x)]^\dagger : R^m \to R^N$$

is well defined. Define the Newton operator at x to be

$$N_f(x) := x - [\nabla f(x)]^\dagger f(x).$$

Definition 1 (Approximate Zero of [5])**.** *Let $x \in \mathbb{R}^N$ and consider a sequence $x_0 = x$, $x_{i+1} = N_f(x_i)$ for $i \geq 0$. The point x is an approximate zero of f if this sequence is well defined and there exists a point $x' \in \mathbb{R}^N$ such that $f(x') = 0$ and*

$$||x_i - x'|| \leq (1/2)^{2^i - i}||x_0 - x'||. \tag{11}$$

Then, we call x' the associated zero of x and say that x represents x'.

Proposition 1 ([22], [5])**.** *There exists a universal constant $a_0 \in \mathbb{R}$ and functions α, β of a system of polynomials equations $F = (f_1(x), \ldots, f_m(x)) : \mathbb{R}^N \to \mathbb{R}^m$ and a vector $x \in \mathbb{R}^N$ such that if $\alpha(F, x) \leq \alpha_0$, then x is an approximate zero of F. In addition, if x' denotes its associated zero, then $||x - x'|| \leq 2\beta(F, x)$. where*

$$\alpha(z, F) = \beta(z, F)\sup_{k>1}\left|\frac{DF(z)^\dagger F^{(k)}(z)}{k!}\right|^{1/(k-1)}, \tag{12a}$$

$$\beta(z, F) = \left\|DF(z)^\dagger F(z)\right\|, \tag{12b}$$

and we know that α_0 is $\frac{1}{4}(13 - 3\sqrt{17}) \approx 0.157671$.

Considering that [22] is somewhat difficult to read and a part of a five-paper series, we refer to the survey of Cucker and Smale [5] or the very recent survey of Beltran and Pardo [2] for an overview.

B. Second Illustration

Second, we illustrate the approach on the power-flow operator F (8a–8c) in terms of voltages V, i.e. the power-flow equalities within ACOPF, as in [6]. We can that whether a point is in *a* domain of monotonicity can be tested by the simple comparison of α and α_0, without solving (9a–9f).

Proposition 2. *For every instance of ACPF, there exists a universal constant $a_0 \in \mathbb{R}$ and a function α of the instance of ACPF and a vector $x \in \mathbb{R}^N$ such that if $\alpha(F, x) \leq \alpha_0$, then x is an approximate zero of F.*

Proof. By Proposition 1. □

IV. The Main Result

Finally, we apply our approach to polynomial optimisation. We study a compact basic semi-algebraic set $\mathbf{K}$ defined by:

$$\mathbf{K} = \{\, \mathbf{x} \in \mathbb{R}^n \,: g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m\}, \tag{13}$$

for some polynomials $g_j \in \mathbb{R}[\mathbf{x}]$, $j = 1, \ldots, m$ in $\mathbf{x} \in \mathbb{R}^N$. Notice that we can convert polynomial equalities to two polynomial inequalities, in principle. The corresponding polynomial optimization problem is:

$$P: \quad f^* = \min_{\mathbf{x}\in\mathbb{R}^N} \{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K}\} \tag{14}$$

where $f \in \mathbb{R}[\mathbf{x}]$ is again a polynomial in $\mathbf{x} \in \mathbb{R}^N$. We assume:

Assumption 1. $\mathbf{K}$ *is compact and $0 \leq g_j \leq 1$ on $\mathbf{K}$ for all $j = 1, \ldots, m$. Moreover, the family of polynomials $\{g_j, 1-g_j\}$ generates the algebra $\mathbb{R}[\mathbf{x}]$.*

Notice that if $\mathbf{K}$ is compact, one may always rescale variables $\mathbf{x}_i$ and add redundant constraints $0 \leq \mathbf{x}_i \leq 1$ for all $i = 1, \ldots, m$, such that the family $\{g_j, 1 - g_j\}$ generates the algebra $\mathbb{R}[\mathbf{x}]$ and Assumption 1 holds. For additional background material on polynomial optimisation, we refer to [1].

A. Lagrangians

A Lagrangian function [3, Definition 5.1.1] of the polynomial optimization problem (14) is:

$$L(\mathbf{x}, \lambda) := f(\mathbf{x}) - \sum_{j=1}^{m} \lambda_j g_j(\mathbf{x}), \tag{15}$$

The textbook version [3] of a Lagrangian relaxation is:

$$\rho_0 := \max_{\lambda\in\mathbb{R}^m_+} \min_{\mathbf{x}\in\mathbb{R}^N} L(\mathbf{x}, \lambda) \tag{16}$$

and it is known that $\rho_0 \leq f^*$. One often adds additional regularisation terms [18], which may improve the rate of convergence, but do not remove the fact that one may have $\rho_0 \ll f^*$.

Recently, it has been realised that one can approximate the global optimum f^* as closely as possible, in case one applies the relaxation to a problem $\tilde{P}$ equivalent to P, which has sufficiently many redundant constraints:

Lemma 1 (Corollary 2.1 in [14])**.** *Let Assumption 1 hold for* $\mathbf{K}$ (13)*. There exists* $\tilde{P}_d$*,* $d \in \mathbb{N}$*, such that for every* $\epsilon > 0$*, there exists* $d_\epsilon \in \mathbb{N}$ *such that for every* $d \geq d_\epsilon$*, the Lagrangian relaxation of* $\tilde{P}_d$*, yields a lower bound* $f^* - \epsilon \leq \rho_d \leq f^*$*.*

For the proof, please see [14]. In particular, the Lagrangian adds redundant products of the initial constraints and is of the form:

$$f(\mathbf{x}) - \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2m}} \lambda_{\alpha\beta} \left(\prod_{j=1}^{m} g_j(\mathbf{x})^{\alpha_j}\right) \times \left(\prod_{j=1}^{m} (1-g_j(\mathbf{x}))^{\beta_j}\right),$$

which is non-convex, but polynomial. Neither is surprising, considering any g_j can be a non-convex polynomial, but we will exploit both subsequently.

Furthermore, we can show a stronger statement:

Assumption 2. *The quadratic module*

$$Q(g) := \{\sum_{j=0}^{m} \sigma_j\, g_j \;:\; \sigma_j \in \Sigma[\mathbf{x}], \quad j = 0, 1, \ldots, m\}$$

where $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ *the space of sums of squares (s.o.s.) polynomials, is Archimedean, i.e.* $\exists R > 0 : R^2 - ||\mathbf{x}||^2 \in Q(g)$*.*

Lemma 2. *Let Assumptions 1 and 2 hold for a polynomial optimisation problem* P *with global optimum* f^**. There exists a* convex $\tilde{P}_d$*,* $d \in \mathbb{N}$*, such that for every* $\epsilon > 0$*, there exists* $d_\epsilon \in \mathbb{N}$ *such that for every* $d \geq d_\epsilon$*, the* convex *Lagrangian relaxation of* $\tilde{P}_d$*, yields a lower bound* $f^* - \epsilon \leq \rho_d \leq f^*$*.*

Proof. The proof follows from Theorem 3.6 of Lasserre [13], when one considers a Lagrangian relaxation of the semidefinite programs. □

Notice, however, that for $d \geq 1$, a single iteration of minimising the *convex* Lagrangian of $\tilde{P}_d$, even using a first-order method, can be computationally much more demanding than a single iteration of second-order methods for the basic Lagrangian ρ_0. We would hence like to study the domains of monotonicity with respect to the various Lagrangians.

B. Monotonicity Domains

Let us now introduce the notion of a domain of monotonicity, which allows for varying the Lagrangian under consideration. In what follows, we denote by tilde the potentially lifted variables etc:

Definition 2 (Monotonicity domain with respect to $\tilde{L}$)**.** *For any* $\tilde{x} \in \mathbb{R}^{\tilde{N}}$ *and* $\tilde{L} : \mathbb{R}^{\tilde{N}} \to \mathbb{R}$*, consider a sequence* $\tilde{x}_0 = \tilde{x}$*,* $\tilde{x}_{i+1} = N_{\tilde{L}}(\tilde{x}_i)$ *for* $i \geq 0$*. The point* $\tilde{x}$ *is within the monotonicity with respect to* $\tilde{L}$ *if this sequence is well defined and there exists a point* $\tilde{x}' \in \mathbb{R}^{\tilde{N}}$ *such that* $\tilde{L}(\tilde{x}') = 0$ *and*

$$||\tilde{x}_i - \tilde{x}'|| \leq (1/2)^{2^i - i} ||\tilde{x}_0 - \tilde{x}'||. \tag{17}$$

Then, we call $\tilde{x}'$ *the associated stationary point of* $\tilde{x}$ *and say that* $\tilde{x}$ *represents* $\tilde{x}'$*.*

Lemma 3. *Let us denote the space of all possible descriptions of a polynomial optimisation problems* (14) *in dimension* N *by* $\mathbb{P}^N$*. There exists a universal constant* $a_0 \in \mathbb{R}$ *and a function* $\alpha : \mathbb{P}^N \times \mathbb{R}^N \to \mathbb{R}$ *such that for all* $N \in \mathbb{Z}, p \in \mathbb{P}^N$*, and* $x \in \mathbb{R}^N$*, there exists* $\tilde{P}_d$*,* $d \in \mathbb{N}$*, such that for every* $\epsilon > 0$*, there exists* $d_\epsilon \in \mathbb{N}$ *such that for every* $d \geq d_\epsilon$*, the Lagrangian relaxation* $\tilde{L}_d$ *of* $\tilde{P}_d$*, yields a lower bound* $f^* - \epsilon \leq \rho_d \leq f^*$*. If* $\alpha(\nabla \tilde{L}_d, x) \leq \alpha_0$*, then* x *is in the domain of monotonicity of stationary point* x' *of Lagrangian relaxation* $\tilde{L}_d$ *of* $\tilde{P}_d$*.*

Proof: We apply the reasoning of Chen [4] to first-order conditions of a Lagrangian relaxation of Lemma 1 of the polynomial optimisation problem (14).

Extending the reasoning to convex Lagrangians of Lemma 2 is non-trivial, because the convex Lagrangians are not necessary polynomial (analytic), as defined. However:

Theorem 2. *Let us denote the space of all possible descriptions of a polynomial optimisation problems* (14) *in dimension* N *by* $\mathbb{P}^N$*. There exists a universal constant* $a_0 \in \mathbb{R}$*, such that for all* $N \in \mathbb{Z}, P \in \mathbb{P}^N$*, where Assumptions 1 and 2 hold for* P*, there exists a* $d \in \mathbb{N}$*, such that for every* $\epsilon > 0$*, there exists* $d_\epsilon \in \mathbb{N}$ *such that for every* $d \geq d_\epsilon$*, there is a* convex *Lagrangian relaxation* $\tilde{L}_d$ *in dimension* $\tilde{N}$*, and a function* $\alpha : \mathbb{P}^{\tilde{N}} \times \mathbb{R}^{\tilde{N}} \to \mathbb{R}$ *such that if* $\alpha(\nabla \tilde{L}_d, \tilde{x}) \leq \alpha_0$*, then* x *is the domain of monotonicity of a solution with objective function value* ρ_d *such that* $f^* - \epsilon \leq \rho_d \leq f^*$*.*

Proof. The proof follows from the observation that each *convex* Lagrangian of Lemma 2 is associated with a non-convex, but polynomial Lagrangian of Lemma 1 and for some d, both Lagrangians of $\tilde{P}_d$ will have a function value bounded from below by $f^* - \epsilon$. Formally, for all $N \in \mathbb{Z}, p \in \mathbb{P}^N$, and $x \in \mathbb{R}^N$, there exists $\tilde{P}_d$, $d \in \mathbb{N}$, such that for every $\epsilon > 0$, there exists $d_\epsilon \in \mathbb{N}$ such that for every $d \geq d_\epsilon$, both the Lagrangian relaxation $\tilde{L}_d$ of $\tilde{P}_d$, and the *convex* Lagrangian relaxation of the same problem $\tilde{P}_d$ yield a lower bound $f^* - \epsilon \leq \rho_d \leq f^*$. While solving minimising the convex Lagrangian of the polynomial optimisation problem (14), we can apply the reasoning of Chen [4] to the first-order conditions of the corresponding non-convex Lagrangian. Whenever the test of Chen [4] allows, we we switch to solving the non-convex Lagrangian, while memorising the current value within the non-convex Lagrangian. While solving the non-convex Lagrangian, we check the evolution of the convex Lagrangian and when it does not decrease, we revert to solving the non-convex Lagrangian starting with the memorised value. □

V. The Application to ACOPF

We implement our approach in a method for solving the alternating-current optimal power flow (ACOPF). First, we use a first-order method for a suitable convex Lagrangian [18]. Second, whenever we detect it is safe to switch to Newton method for a non-convex Lagrangian [18], we do so. The test does not require solving any additional optimisation problems, and can be summarised as follows:

Algorithm 1 Schema of the hybrid method

1: Initialise $\mathbf{x} \in \mathbb{R}^N, \lambda \in \mathbb{R}^m$, the augmented non-convex Lagrangian function L, and the augmented convex Lagrangian function $\tilde{L}$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Update $(\mathbf{x}, \lambda)$ using the low-rank coordinate descent of [18] for $\nabla \tilde{L} = 0$
4: **if** $\alpha(\nabla L, \mathbf{x}) \leq \alpha_0$ **then**
5: $S \leftarrow (\mathbf{x}, \lambda)$
6: **for** $l = 0, 1, 2, \ldots$ **do**
7: Update $(\mathbf{x}, \lambda)$ using Newton step on $\nabla L = 0$
8: **if** $\tilde{L}(\mathbf{x}, \lambda) > \tilde{L}(S)$ **then**
9: $(\mathbf{x}, \lambda) \leftarrow S$
10: **break**
11: **end if**
12: **if** $T(\mathbf{x}, \lambda) < \epsilon$ **then**
13: **break**
14: **end if**
15: **end for**
16: **end if**
17: **if** $T(\mathbf{x}, \lambda) < \epsilon$ **then**
18: **break**
19: **end if**
20: **end for**

Corollary 1. *For every instance of ACOPF, there exists a universal constant $a_0 \in \mathbb{R}$ and functions α, β, f of the instance of ACOPF and a vector $x \in \mathbb{R}^N$ such that if $\alpha(f, x) \leq \alpha_0$, then x is in the domain of monotonicity of an optimum of the instance of ACOPF, which is no more than ϵ away from the value of the global optimum with respect to the objective function.*

Proof. By Theorem 2. □

See Algorithm 1 for an overview of the hybridisation we propose. Notice that for the sake of the clarity of presentation, the overview is parametric (e.g., does not specify $T(\mathbf{x}, \lambda)$ used in the termination criterion, does not provide details of the Lagrangian functions for ACOPF, as available from [18]), and assumes that both the convex and non-convex Lagrangian are in the same dimension (i.e., $\mathbf{x} \in \mathbb{R}^N, \lambda \in \mathbb{R}^m$), which need not be the case, generally. Still, it demonstrates the key ideas. The Newton step (Line 7 in Algorithm 1) employs the same regularisation as the convex Lagrangian of [18] and a backtracking line search until a sufficient decrease in L is observed. With the non-convex Lagrangian L, Newton direction may turn out not to be a direction of descent. In dealing with the negative curvature, we multiply the direction by -1 [20], although we could also use a damped variant of Newton method [20]. Multiple Newton steps, each satisfying sufficient decrease, are performed in each iteration of the loop, before a sufficient decrease in the convex Lagrangian is tested. In an extended version of the paper, we present a more detailed algorithm, including the mappings between the domains of the non-convex Lagrangians of Lemma 1 and convex Lagrangians of Lemma 2.

Computational Experiments

Our implementation employs a formulation of the ACOPF considering tap-changing and phase-shifting transformers, parallel lines, and multiple generators at one bus, among others. To save some of the cost of computing Newton direction for large instances, we exploit sparsity of the Hessian matrix in the classical conjugate gradient method [9]. Since the Hessian matrix is usually less than 1% dense for real-world instances, e.g., the highest sparsity recorded for case2383wp is 0.35%, utilizing the sparse structure reduces the computational cost by several orders of magnitude.

We performed numerical experiments on a collection of well-known instances [27], ranging from case30 with 6 generators and 41 branches, to case 2383wp with 327 generators and 2896 branches. In Figure 1, we illustrate the computational performance of our implementation on case30, case39, and case118. In particular, we compare the coordinate descent of [18] (plotted in cyan and labelled as *CD only*), against a pure Newton method (plotted in blue and labelled as *Newton Only*), against hybrid methods switching to Newton method on the non-convex Lagrangian after 10 (blue), 40 (green), 160 (magenta), 320 (black) iterations of the coordinate descent on the convex Lagrangian. First, notice that the choice of the number of iterations of the coordinate descent performed prior to switching over to the Newton method is crucial for the convergence properties of the hybrid method. Second, notice the number of iterations of the coordinate descent to be performed, in order for the hybrid method to converge and in order for it to converge as fast as possible, increases as the size of the instance increases. A test such as ours is hence needed.

To validate the effectiveness of our approach, we performed experiments on case2383wp, which is one of the "Polish" instances. Figure 2 shows that coordinate descent alone (*CD only*) performs much worse than hybrid algorithms, possibly due to the rapid decrease of infeasibility when infeasibility is reduced below some threshold. With Newton steps only (*Newton only*), the infeasibility decreases, but at a rate slower than that of our hybrid method. This is particularly pronounced in the comparison of run-time, rather than the number of iterations; Newton method seems distinctly slow. Our hybrid methods, with *any* choice of the number of CD steps tested, outperform *CD only* and *Newton only*. Notice that in Figures 1-2, some of the lines are much shorter than others. This is because some methods converge to a local optimum, rather than the global optimum, which causes rejections in Line 13 of Algorithm 1. Notice also that the virtual machines used a processor rated at 800 MHz, rendering the wall-clock time less impressive than what it may be on a more powerful machine.

Similar observations can be made also when considering the objective function value, where the hybrid methods converge faster than the others. With a proper choice of the number of CD steps, the hybrid methods improve upon rates of

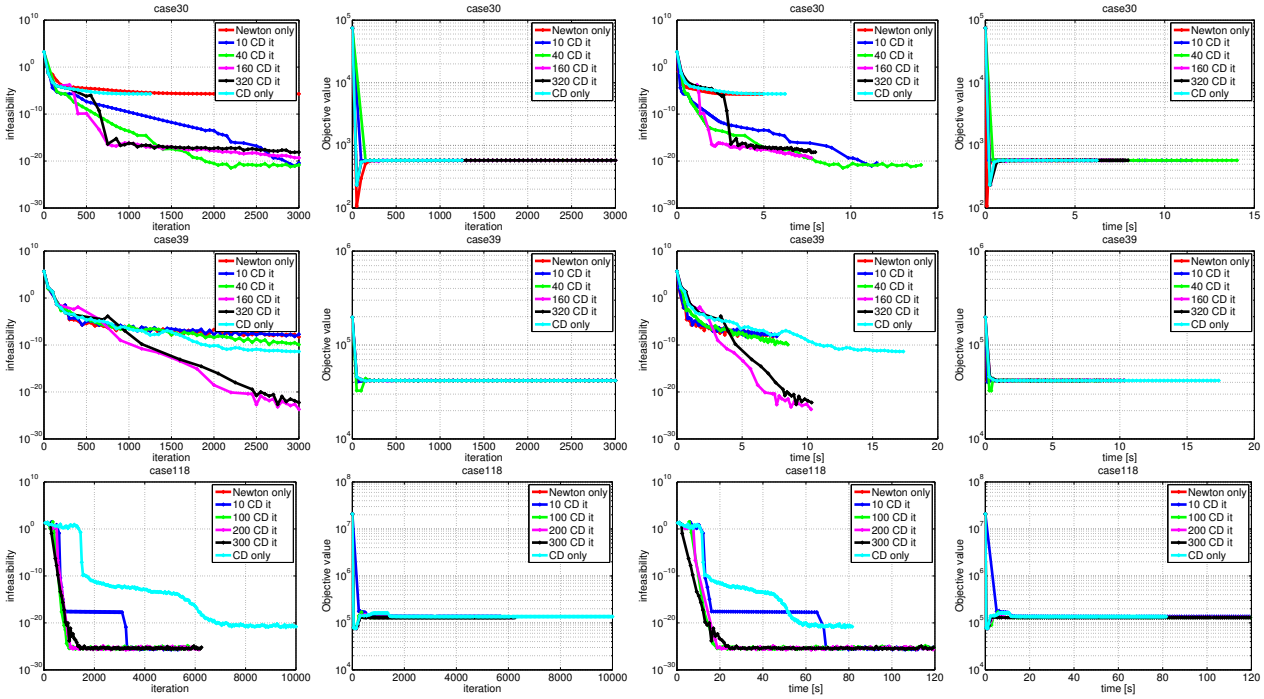

Figure 1: The evolutions of infeasibility and objective function value over the number of iterations (first two columns) and time (last two columns) for various switching strategy (from first order to second order method) on case30, case39 and case118.

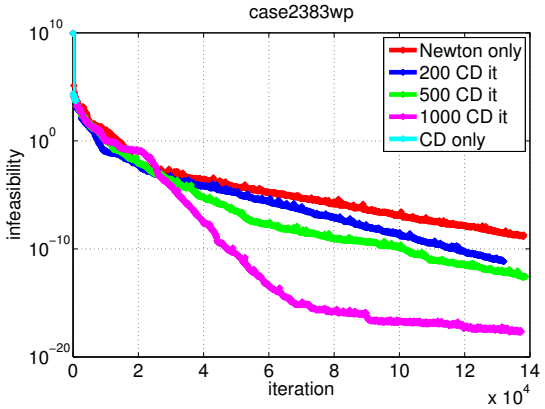

Figure 2: The evolutions of infeasibility over the number of iterations on case2383wp.

convergence possible using SeDuMi [24], SDPA [26], or SDPT3 [25], in a variety of formulations. We aim to present this comparison and more complete results, generally, in an extended version of this paper.

VI. Conclusions

Without the use of (hierarchies of) convex relaxations, Newton-type methods can converge to particularly bad local optima of non-convex problems. Even the fastest first-order methods for computing strong convex relaxations are, however, rather slow on their own. Hybrid methods combining first-order methods for the strong convex relaxations and Newton-type methods for the non-convex problems combine the guarantees of convergence associated with (hierarchies of) convex relaxations and the quadratic rates of convergence of Newton method. This improves upon recent solvers for SDP relaxations in power systems [8], [18] considerably and opens up many novel research directions. Most interestingly, we wonder if such hybrid methods could be combined with the recently proposed cutting-plane methods [12], [11], [19], [8].

References

[1] M. F. Anjos and J.-B. Lasserre, editors. *Handbook on semidefinite, conic and polynomial optimization*, volume 166 of *International series in operations research & management science*. Springer, New York, 2012.

[2] C. Beltrán and L. M. Pardo. Efficient polynomial system-solving by numerical methods. *Journal of Fixed Point Theory and Applications*, 6(1):63–85, 2009.

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

[4] P. Chen. Approximate zeros of quadratically convergent algorithms. *Mathematics of Computation*, 63(207):247–270, 1994.

[5] F. Cucker and S. Smale. Complexity estimates depending on condition and round-off error. *J. ACM*, 46(1):113–184, Jan. 1999.

[6] K. Dvijotham, S. Low, and M. Chertkov. Solving the power flow equations: A monotone operator approach. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, 2015.

[7] K. Dvijotham, S. Low, and M. Chertkov. Solving the power flow equations: A monotone operator approach. abs/1506.08472, 2015.

[8] B. Ghaddar, J. Mareček, and M. Mevissen. Optimal power flow as a polynomial optimization problem. *IEEE Trans. Power Syst.*, page to appear, 2015.

[9] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[10] L. Kantorovich. On Newton's method for functional equations. *Dokl. Akad. Nauk SSSR*, 59(7):1237–1240, 1948.

[11] B. Kocuk, S. S. Dey, and X. A. Sun. Strong socp relaxations for the optimal power flow problem. *Power Systems, IEEE Transactions on*, 2015.

[12] X. Kuang, L. F. Zuluaga, B. Ghaddar, and J. Naoum-Sawaya. Approximating the acopf problem with a hierarchy of socp problems. In *Power Energy Society General Meeting, 2015 IEEE*, pages 1–5, July 2015.

[13] J. Lasserre. Convergent sdp-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):882–843, 2006.

[14] J. B. Lasserre. A lagrangian relaxation view of linear and semidefinite hierarchies. *SIAM Journal on Optimization*, 23(3):1742–1756, 2013.

[15] J. Lavaei and S. Low. Zero duality gap in optimal power flow problem. *IEEE T. Power Syst.*, 27(1):92–107, 2012.

[16] S. Low. Convex relaxation of optimal power flow – part i: Formulations and equivalence. *Control of Network Systems, IEEE Transactions on*, 1(1):15–27, March 2014.

[17] S. Low. Convex relaxation of optimal power flow – part ii: Exactness. *Control of Network Systems, IEEE Transactions on*, 1(2):177–189, June 2014.

[18] J. Marecek and M. Takac. A Low-Rank Coordinate-Descent Algorithm for Semidefinite Programming Relaxations of Optimal Power Flow. *ArXiv*, (arXiv:1506.08568).

[19] D. Molzahn and I. Hiskens. Sparsity-exploiting moment-based relaxations of the optimal power flow problem. *Power Systems, IEEE Transactions on*, 30(6):3168–3180, Nov 2015.

[20] Y. Nesterov, A. Nemirovskii, and Y. Ye. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

[21] P. Panciatici, M. Campi, S. Garatti, S. Low, D. Molzahn, A. Sun, and L. Wehenkel. Advanced optimization methods for power systems. In *Power Systems Computation Conference (PSCC), 2014*, pages 1–18, Aug 2014.

[22] M. Shub and S. Smale. On the complexity of Bezout's theorem i – Geometric aspects. *Journal of the AMS*, 6(2), 1993.

[23] S. Smale. The fundamental theorem of algebra and complexity theory. *Bull. Amer. Math. Soc*, 4(1):1–36, 1981.

[24] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[25] K.-C. Toh, M. J. Todd, and R. H. Tütüncü. Sdpt3 – A matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[26] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, 18(4):491–505, 2003.

[27] 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 T. Power Syst.*, 26(1):12–19, 2011.