

Alternating current optimal power flow with generator selection^{*}

Esteban Salgado¹, Andrea Scozzari², Fabio Tardella³, and Leo Liberti¹

¹ CNRS LIX, École Polytechnique, 91128 Palaiseau, France

`esteban.salgado@polytechnique.edu`, `liberti@lix.polytechnique.fr`

² Facoltà di Economia, Università degli Studi Niccolò Cusano-Telematica, Roma, Italy `andrea.scozzari@unicusano.it`

³ MEMOTEF, Facoltà di Economia, Università “La Sapienza”, Roma, Italy `fabio.tardella@uniroma1.it`

Abstract. We investigate a mixed-integer variant of the alternating current optimal power flow problem. The binary variables activate and deactivate power generators installed at a subset of nodes of the electrical grid. We propose some formulations and a mixed-integer semidefinite programming relaxation, from which we derive two mixed-integer diagonally dominant programming approximation (inner and outer, the latter providing a relaxation). We discuss dimensionality reduction methods to extract solution vectors from solution matrices, and present some computational results showing how both our approximations provide tight bounds.

Keywords: smart grid, semidefinite programming, diagonal dominance, dimensionality reduction.

1 Introduction

The Alternating Current Optimal Power Flow (ACOPF) problem is as follows: given an electric power network consisting of nodes (called *buses*) and links (called *lines*) one seeks an optimal generation and distribution plan of active and reactive power under physical constraints (Ohm’s and Kirchhoff’s laws), and subject to power generation, voltage magnitude and current bounds on each line.

Not every bus can produce power. Those which can are called *generators*. There is often a planning issue related to their activation and deactivation. This is important because of two reasons. First, the minimum amount of power produced by a generator may be a (reasonably large) positive constant, so disabling a generator is not equivalent to keeping it active at minimum levels. Secondly, there can be a cost to keep it activated. Modelling this choice implies the addition of binary variables to the model, which yields a Mixed-Integer Quadratically Constrained Quadratic Programming (MIQCQP) problem.

Most of the recent ACOPF literature [5,9,17,18] ignores the generator activation/deactivation issue: every available generator is always active. Equivalently,

^{*} This work was partially supported by the PGM grant “Projet 2016-1751H”, by a Siebel Energy Institute Seed Grant, and by Università di Roma “La Sapienza” under a visiting grant for the last author.

every binary variable is fixed to 1, which yields a continuous Quadratically Constrained Quadratic Programming (QCQP) formulation. We were drawn to the mixed-integer ACOPF variant by [22], where the authors also consider the selection of Phase-Shifting Transformers (PST), used to control the flow of real power, as well as *shunts*, which are stabilizing devices. We do not consider PSTs nor shunts in this paper: they require technical data (specifically, more information about the admittance matrices) which we do not possess at this stage. From a theoretical point of view, however, the activation/deactivation of generators and the selection of PSTs yield formulations of the same MIQCQP class, that are likely to require similar solution approaches.

In this paper we study the ACOPF with selection of generators (ACOPFG). Based on the ideas in [1, 2, 3], we will derive Mixed-Integer Linear Programming (MILP) formulations using Diagonally Dominant Programming (DDP) for inner and outer approximations for the ACOPF with binary variables. More precisely, we propose the following relaxations:

1. a Mixed-Integer Semidefinite Programming (MISDP) relaxation of the original MIQCQP formulation of the ACOPFG;
2. an *inner* Diagonally Dominant Programming (DDP) [1, 10] approximation of the MISDP relaxation;
3. an *outer* DDP relaxation obtained by replacing the primal DD cone in the inner DDP approximation with its dual.

We shall exhibit some computational results showing that DDP yields tight upper and lower bounds on the optimal objective function value of the original MIQCQP.

One of the important reasons for the usefulness of a tight *lower* objective function bound is that ACOPF/ACOPFG problems are sometimes solved as lower-level subproblems of bilevel problems where the upper-level decisions concern unit electricity prices. A non-guaranteed heuristic ACOPF/ACOPFG solution might be detrimental to finding good solutions of the upper-level problem with cutting plane approaches. Since we cannot hope to find a guaranteed global optimum in reasonable times, a tight lower bound represents a good trade-off.

1.1 Notation

We remark first that alternating currents are commonly modelled by means of scalar and vector quantities over the complex field \mathbb{C} . On the other hand, physicists have always denoted current by i . The ambiguity with the usual notation for $\sqrt{-1}$ is resolved by denoting the latter by j . Accordingly, we shall refrain from using j as an index (as is common in Mathematical Programming). In agreement with standard notation in complex numbers, we denote complex conjugation by means of a bar over the corresponding quantity: \bar{i} , for example, is the complex conjugate of the current, rather than $-\sqrt{-1}$. Complex conjugation is applied componentwise to tensors.

We often neglect to explicitly mention matrix sizes in formulations, in order to lighten the notation. Symbols denoting classes of matrices, such as \mathcal{S} , \mathcal{D} , etc.,

really stand for “the subset of matrices of *appropriate size* with the corresponding property”.

1.2 Contents

The rest of this paper is organized as follows. In Sect. 2 we discuss formulations of the ACOPFG. In Sect. 3 we introduce a MISDP relaxation. In Sect. 4 we present our new inner and outer relaxations based on DDP. In Sect. 5 we briefly discuss dimensionality reduction issues in order to retrieve a good solution from the relaxation output. In Sect. 6 we report computational results.

2 ACOPF formulations

Consider an electric power network with set of buses \mathcal{N} , set of generators $\mathcal{G} \subseteq \mathcal{N}$ and set of lines $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$. The parameters of the problem are:

- $\forall n \in \mathcal{N}$ $S_n^D \triangleq P_n^D + jQ_n^D$ active/reactive power demand at bus n
- $\forall g \in \mathcal{G}$ $S_g^{\min} \triangleq P_g^{\min} + jQ_g^{\min}$ lower bound on power generated at bus g
- $\forall g \in \mathcal{G}$ $S_g^{\max} \triangleq P_g^{\max} + jQ_g^{\max}$ upper bound power generated at bus g
- $\forall n \in \mathcal{N}$ v_n^{\min}, v_n^{\max} bounds on voltage magnitude at bus n
- $\forall (n, m) \in \mathcal{L}$ i_{nm}^{\max} upper bound on current on line (n, m)
- Y bus admittance matrix in $\mathbb{C}^{|\mathcal{N}| \times |\mathcal{N}|}$
- Y^f, Y^s line admittance matrices in $\mathbb{C}^{|\mathcal{L}| \times |\mathcal{N}|}$.

We want to find the optimal values of active and reactive power at each generator $g \in \mathcal{G}$ that is switched on ($S_g^G \triangleq P_g^G + jQ_g^G$). We must also decide the current and the voltage for the nodes and lines of the system. Given that:

$$\begin{aligned} \forall n \in \mathcal{N} \quad i_n &= \sum_{m \in \mathcal{N}} Y_{nm} v_m \\ \forall n \in \mathcal{N} \setminus \mathcal{G} \quad v_n \bar{i}_n &= -S_n^D \\ \forall n \in \mathcal{G} \quad v_n \bar{i}_n &= S_n^G - S_n^D \end{aligned}$$

we can model the problem by only using voltage variables $v_n \in \mathbb{C}$ (for $n \in \mathcal{N}$) [23] and binary variables z_g (for $g \in \mathcal{G}$) representing the activation of a generator to

produce power. The formulation of the ACOPFG is as follows:

$$\left. \begin{aligned} & \min_{\substack{v \in \mathbb{C}^{|\mathcal{N}|} \\ z \in \{0,1\}^{|\mathcal{G}|}}} f(v, z) \\ \forall n \in \mathcal{N} \setminus \mathcal{G} & \quad \sum_{m \in \mathcal{N}} v_n \bar{v}_m \bar{Y}_{nm} = -S_n^{\text{D}} \\ \forall g \in \mathcal{G} & \quad \sum_{m \in \mathcal{N}} v_g \bar{v}_m \bar{Y}_{gm} - S_g^{\max} z_g \leq -S_g^{\text{D}} \\ \forall g \in \mathcal{G} & \quad \sum_{m \in \mathcal{N}} v_g \bar{v}_m \bar{Y}_{gm} - S_g^{\min} z_g \geq -S_g^{\text{D}} \\ \forall n \in \mathcal{N} & \quad v_n \bar{v}_n \leq v_n^{\max} \\ \forall n \in \mathcal{N} & \quad v_n \bar{v}_n \geq v_n^{\min} \\ \forall (n, m) \in \mathcal{L} & \quad \sum_{k, \ell \in \mathcal{N}} \bar{v}_k v_\ell \bar{Y}_{(n,m)k}^{\text{f}} Y_{(n,m)\ell}^{\text{f}} \leq l_{nm}^{\max} \\ \forall (n, m) \in \mathcal{L} & \quad \sum_{k, \ell \in \mathcal{N}} \bar{v}_k v_\ell \bar{Y}_{(n,m)k}^{\text{s}} Y_{(n,m)\ell}^{\text{s}} \leq l_{nm}^{\max}, \end{aligned} \right\} \text{(ACOPFG}_{\mathbb{C}}) \quad (1)$$

where f is a real-valued function that is commonly considered as a polynomial on the total active power generated, i.e.:

$$f(v, z) \triangleq \sum_{g \in \mathcal{G}} \left(c_{g,0} z_g + \sum_{1 \leq \ell \leq L} c_{g,\ell} (P_g^{\text{G}})^{\ell} \right) = \sum_{g \in \mathcal{G}} \left(c_{g,0} z_g + \sum_{\ell \leq L} c_{g,\ell} \left(\text{Re} \left(\sum_{m \in \mathcal{N}} v_g \bar{v}_m \bar{Y}_{gm} \right) \right)^{\ell} \right).$$

We shall consider polynomials which are quadratic in v_g and linear in P_g^{G} (e.g. $L = 1$). This makes (ACOPFG_C) a complex-valued MIQCQP.

By doubling the number of the variables (which implies quadrupling the sizes of the matrices in each constraint), the problem can be exactly reformulated into a real-valued MIQCQP:

$$\left. \begin{aligned} & \min_{\substack{v \in \mathbb{R}^{2|\mathcal{N}|} \\ z \in \{0,1\}^{|\mathcal{G}|}}} v^{\top} C v + c^{\top} z \\ \forall k \in \mathcal{E} & \quad v^{\top} A^k v = a_k \\ \forall \ell \in \mathcal{I} & \quad v^{\top} B^{\ell} v \leq b_{\ell} \\ \forall w \in \mathcal{Z} & \quad v^{\top} Q^w v - q_w^{\max} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} & \quad v^{\top} Q^w v - q_w^{\min} z_{\lceil w/2 \rceil} \geq q_w, \end{aligned} \right\} \text{(ACOPFG}_{\mathbb{R}}) \quad (2)$$

where \mathcal{E} is the set indexing the $2|\mathcal{N} \setminus \mathcal{G}|$ equality constraints, \mathcal{Z} is the set indexing the $2|\mathcal{G}|$ inequalities with binary variables and \mathcal{I} is the set indexing the other $2(2|\mathcal{G}| + 2|\mathcal{N}| + 2|\mathcal{L}|)$ inequality constraints.

We remark that this problem is non-convex. Eq. (2) therefore cannot generally be solved globally using local optimization methods (as a convex QCQP would). Indeed, it is well known that the decision problem associated to Eq. (2) is **NP**-hard [24]

3 MISDP relaxation

Semidefinite Programming (SDP) [4, 16] is widely employed in order to derive relaxations of the continuous version of the ACOPF [5, 9, 18]. We derive a corresponding MISDP relaxation for the ACOPFG. First we rewrite Eq. (2) by

replacing the products of the form $v^\top M v$ as $\langle M, V \rangle = \text{tr}(M^\top V)$ and adding the rank constraint $V = v v^\top$:

$$\left. \begin{array}{l} \min_{V \in \mathcal{S}} \quad \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} \quad \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} \quad \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\max} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\min} z_{\lceil w/2 \rceil} \geq q_w \\ V = v v^\top, \end{array} \right\} (\text{ACOPFG}_{\mathbb{R}}) \quad (3)$$

where \mathcal{S} is the set of all $n \times n$ symmetric matrices.

Given that V does not encode any integrality constraint, Eq. (3) is essentially a MILP with the additional (non-convex) constraint $V = v v^\top$, which states that V must be positive semidefinite (PSD) and have rank 1. The standard MISDP relaxation is obtained by relaxing the rank 1 constraint to $V - v v^\top \succeq 0$, i.e. by requiring that the Schur complement of V and v , defined as

$$\mathcal{S}(V, v) = \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix}$$

is PSD:

$$\left. \begin{array}{l} \min_{\substack{v \in \mathcal{S}, v \in \mathbb{R}^{2|\mathcal{N}|} \\ z \in \{0,1\}^{|\mathcal{G}|}}} \quad \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} \quad \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} \quad \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\max} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\min} z_{\lceil w/2 \rceil} \geq q_w \\ \mathcal{S}(V, v) \succeq 0 \end{array} \right\} (\text{ACOPFG}_{\text{MISDP}}) \quad (4)$$

Most SDP solvers are unable to solve MISDP formulations directly (with some exceptions, e.g. Pajarito [21] and PENLAB [11]). A possible workaround consists in reformulating the binary constraints on z by $\forall g \in \mathcal{G} (z_g^2 = z_g)$. These can in turn be relaxed, within the SDP framework, with the constraints:

- $\text{diag}(Z) = z$
- $\mathcal{S}(Z, z) \succeq 0$,

This yields the following SDP relaxation:

$$\left. \begin{array}{l} \min_{\substack{v \in \mathcal{S}, v \in \mathbb{R}^{2|\mathcal{N}|} \\ z \in \mathcal{S}, z \in \mathbb{R}^{|\mathcal{G}|}}} \quad \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} \quad \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} \quad \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\max} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\min} z_{\lceil w/2 \rceil} \geq q_w \\ \text{diag}(Z) - z = 0 \\ \mathcal{S}(V, v) \succeq 0 \\ \mathcal{S}(Z, z) \succeq 0. \end{array} \right\} (\text{ACOPFG}_{\text{SDP}}) \quad (5)$$

While Eq. (5) can be tackled by an SDP solver, there are two issues with it: (a) the fact that today, current SDP solver technology is far from allowing the systematic solution of ACOF instances of even moderate sizes (into the hundreds or thousands of nodes and beyond); (b) the proposed SDP relaxation of the binary variables usually yields very poor bounds. Instead, we shall investigate below some MILP formulations derived from Eq. (5). While MILP is also **NP**-hard, its state-of-the-art solvers are much faster than MIQCQP solvers.

4 Diagonal dominance

A real square symmetric matrix $Y = (Y_{j\ell})$ is *diagonally dominant* (DD) if

$$\forall j \quad Y_{jj} \geq \sum_{\ell \neq j} |Y_{j\ell}|. \quad (6)$$

The main observation leading to our tight MILP relaxations of the MIQCQP is the well-known fact that *every diagonally dominant matrix is PSD* [13]. This means that by replacing the constraint $\mathcal{S}(V, v) \succeq 0$ with membership in the cone of DD matrices (also called the *DD cone*) we obtain an inner approximation of the SDP cone. We restrict our attention to two formulations of these Diagonally Dominant Programs (DDP) applied to (ACOPF_{MISDP}): a formulation based on the extreme rays of the DD cone [6], and the *outer approximation* obtained by replacing the extreme rays of the DD cone with the extreme rays of the dual DD cone. We devote particular attention to the latter, since it has two desirable properties: (i) it provides a relaxation of the MISDP, and (ii) we found empirically that it provides tight lower bounds.

From Eq. (4) we derive the inner approximation:

$$\left. \begin{array}{l} \min_{V \in \mathcal{S}, v \in \mathbb{R}^{2|\mathcal{N}|}, z \in \{0,1\}^{|\mathcal{G}|}} \quad \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} \quad \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} \quad \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\max} z_w \leq q_w \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w^{\min} z_w \geq q_w \\ V \in \mathcal{D}, \end{array} \right\} \text{(ACOPF}_{\text{inner}}) \quad (7)$$

where \mathcal{D} denotes the DD cone. Given that the decision variable vector v only appears in the conic constraint $\mathcal{S}(V, v) \succeq 0$, we can replace the latter by $V \succeq 0$ without modifying the feasible region. We then inner-approximate it by means of the constraint $V \in \mathcal{D}$, which is equivalent to Eq. (6) and can be written as follows using linear constraints:

$$\left. \begin{array}{l} \forall j \quad V_{jj} \geq \sum_{\ell \neq j} Z_{j\ell} \\ -Z \leq V \leq Z. \end{array} \right\} \quad (8)$$

By [6] and [2, Lemma 3.2], we know that the extreme rays of \mathcal{D} are square matrices defined as follows: $\forall j, \ell$ such that $j < \ell$,

$$\begin{aligned} \bullet E_j &= e_j^\top e_j \\ \bullet E_{j\ell}^+ &= (e_j + e_\ell)^\top (e_j + e_\ell) \\ \bullet E_{j\ell}^- &= (e_j - e_\ell)^\top (e_j - e_\ell) \end{aligned} \quad (9)$$

In other words, the rays either have one nonzero entry in the diagonal which is equal to 1 or have a single nonzero principal minor

$$\begin{pmatrix} 1 & \pm 1 \\ \pm 1 & 1 \end{pmatrix}$$

where, for $j < \ell$, the ones on the diagonal are in positions (j, j) and (ℓ, ℓ) , and the ± 1 components are in positions (j, ℓ) and (ℓ, j) . The main relevant result is that every matrix $M \in \mathcal{D}$ can be written as a non-negative combination of these extreme rays:

$$\exists \delta_j, \delta_{j\ell}^+, \delta_{j\ell}^- \geq 0 \quad M = \sum_j \delta_j E_j + \sum_{j < \ell} (\delta_{j\ell}^+ E_{j\ell}^+ + \delta_{j\ell}^- E_{j\ell}^-). \quad (10)$$

This allows us to write membership of a square symmetric matrix M in the dual DD cone \mathcal{D}^* by means of linear constraints as stated in the following result.

Lemma 4.1 (From paper [23]). *The dual DD cone \mathcal{D}^* can be written as follows:*

$$\mathcal{D}^* = \{M \in \mathcal{S} \mid \forall x \in \mathcal{X} (x^\top M x \geq 0)\}, \quad (11)$$

where $\mathcal{X} = \{(e_j)_j, (e_j \pm e_\ell)_{j < \ell}\}$ is a set of $|\mathcal{N}| + 2 \binom{|\mathcal{N}|}{2} = |\mathcal{N}|^2$ elements.

From Eq. (4) we derive the outer approximation:

$$\left. \begin{aligned} \min_{\substack{V \in \mathcal{S}, v \in \mathbb{R}^{2|\mathcal{N}|} \\ z \in \{0,1\}^{|\mathcal{G}|}}} & \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} & \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} & \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} & \langle Q^w, V \rangle - q_w^{\max} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} & \langle Q^w, V \rangle - q_w^{\min} z_{\lceil w/2 \rceil} \geq q_w \\ & V \in \mathcal{D}^*. \end{aligned} \right\} (\text{ACOPFG}_{\text{outer}}) \quad (12)$$

By our description of the extreme rays of \mathcal{D} , the constraint $V \in \mathcal{D}^*$ can be written using Lemma 4.1. Eq. (12) is an outer approximation of Eq. (4) because $\mathcal{D} \subseteq \mathcal{S}^+ = (\mathcal{S}^+)^* \subseteq \mathcal{D}^*$, where \mathcal{S}^+ is the cone of PSD matrices.

4.1 Iterative inner approximation

There are two potential issues with the inner approximation Eq. (7): (a) given that we are modifying the conic constraints of Eq. (4) by an inner approximation

of the original cone, the resulting problem may be infeasible; (b) even if the problem turns out to be feasible, the solution may not be a good approximation of the MISDP.

The authors of [1, 2] introduce a possible way around these issues by means of iteratively solving a sequence of auxiliary problems, until a cone $\mathcal{C} \subseteq \mathcal{S}^+$ is found on which Eq. (4) is feasible when replacing the conic constraint $V \in \mathcal{D}$ by $V \in \mathcal{C}$. The auxiliary problems are obtained by varying the (square matrix) parameter U in the formulation below:

$$\left. \begin{array}{l} \min_{\substack{V \in \mathcal{S}, z \in \{0,1\}^{|\mathcal{G}|} \\ \delta \in (\mathbb{R}^+)^{4|\mathcal{N}|^2}}} \langle C, V \rangle + c^\top z \\ \forall k \in \mathcal{E} \quad \langle A^k, V \rangle = a_k \\ \forall \ell \in \mathcal{I} \quad \langle B^\ell, V \rangle \leq b_\ell \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w \max_{z \in \{0,1\}^{|\mathcal{G}|}} z_{\lceil w/2 \rceil} \leq q_w \\ \forall w \in \mathcal{Z} \quad \langle Q^w, V \rangle - q_w \min_{z \in \{0,1\}^{|\mathcal{G}|}} z_{\lceil w/2 \rceil} \geq q_w \\ U^\top \left(\sum_{x \in \mathcal{X}} \delta_x x x^\top \right) U = V. \end{array} \right\} (\text{ACOPFG}_{\text{inner}}(U)) \quad (13)$$

We remark that, while $U^\top \left(\sum_{x \in \mathcal{X}} \delta_x x x^\top \right) U$ is an appropriate description of $V \in \mathcal{D}^*$, implementation performances improve if we use an additional auxiliary matrix variable:

$$\begin{aligned} \forall i \quad W_{ii} &= \delta_i + \sum_{j \neq i} (\delta_{ij}^+ + \delta_{i,j}^-) \\ \forall i, j \quad W_{ij} &= \delta_{ij}^+ - \delta_{i,j}^- \\ V &= U^\top W U \end{aligned} \quad (14)$$

The iterative procedure is the following:

1. Solve $(\text{ACOPFG}_{\text{inner}})$, obtaining the solution V^* ;
2. Define $U = \text{chol}(V^*)$;
3. Solve $(\text{ACOPFG}_{\text{inner}}(U))$ to obtain V^* . While the solution is improving, repeat from Step 2.

At the k -th iteration of this procedure, we consider the parametrization matrix U_k and the optimum V_k^* of $(\text{ACOPFG}_{\text{inner}}(U_k))$. Then the following holds [1, 2].

Proposition 4.2. V_k^* is feasible for $(\text{ACOPFG}_{\text{inner}}(U_{k+1}))$ and cannot worsen the current objective value: $\text{val}(\text{ACOPFG}_{\text{inner}}(U_{k+1})) \leq \text{val}(\text{ACOPFG}_{\text{inner}}(U_k))$. Moreover, if V_k^* is PSD and $\text{val}(\text{ACOPFG}_{\text{inner}}(U_k))$ is not optimal, then V_k^* is improving, i.e. $\text{val}(\text{ACOPFG}_{\text{inner}}(U_{k+1})) < \text{val}(\text{ACOPFG}_{\text{inner}}(U_k))$.

In particular, we know that if we start from a feasible problem, the iterated problems will always remain feasible. To find an initial U such that the parametrized problem is feasible, we solve a slightly different formulation, where we add a

slack variable that we minimize in order to attempt to achieve feasibility:

$$\left. \begin{array}{l}
 \min_{\substack{V \in \mathcal{S}, z \in \{0,1\}^{|\mathcal{G}|} \\ \delta \in (\mathbb{R}^+)^4 |\mathcal{N}|^2, \alpha \in \mathbb{R}^4 |\mathcal{N}|^2}} \alpha \\
 \forall k \in \mathcal{E} \quad \langle A^k, V + \alpha \mathbb{I} \rangle = a_k \\
 \forall \ell \in \mathcal{I} \quad \langle B^\ell, V + \alpha \mathbb{I} \rangle \leq b_\ell \\
 \forall w \in \mathcal{Z} \quad \langle Q^w, V + \alpha \mathbb{I} \rangle - q_w^{\max} z_w \leq q_w \\
 \forall w \in \mathcal{Z} \quad \langle Q^w, V + \alpha \mathbb{I} \rangle - q_w^{\min} z_w \geq q_w \\
 U^\top \left(\sum_{x \in \mathcal{X}} \delta_x x x^\top \right) U = V + \alpha \mathbb{I}.
 \end{array} \right\} (\text{ACOPFG}_{\text{ph1}}(U)) \quad (15)$$

Algorithmically, we start with $U = \mathbb{I}$. While $\alpha^* > 0$, we set $U = \text{chol}(V^* + \alpha^* \mathbb{I})$. We solve $(\text{ACOPF}_{\text{ph1}}(U))$ using the procedure previously stated.

The pseudocode for our inner approximation algorithm is given in Alg. 1.

Algorithm 1 Iterative inner approximation

```

1: procedure INNERACOPFMI
2:    $\epsilon \leftarrow 10^{-5}$ ,  $\text{AUX} \leftarrow 1$ ,  $U \leftarrow \mathbb{I}$ 
3:   # Achieve feasible U first
4:   while  $\text{AUX} > 0$  do
5:      $(V^*, z^*, \alpha^*) \leftarrow$  optimal solution of  $(\text{ACOPFG}_{\text{ph1}}(U))$ 
6:      $\text{AUX} \leftarrow \alpha^*$ ,  $U \leftarrow \text{CHOL}(X + \alpha^* \mathbb{I})$ 
7:    $\text{VALF} \leftarrow +\infty$ ,  $\text{VALI} \leftarrow 0$ 
8:   # Now improve the approximation
9:   while  $|\text{VALI} - \text{VALF}| > \epsilon$  do
10:     $\text{VALI} \leftarrow \text{VALF}$ 
11:     $((V^*, z^*), \text{VALF}) \leftarrow$  optimal solution and value of  $(\text{ACOPFG}_{\text{inner}}(U))$ 
12:     $U \leftarrow \text{CHOL}(V^*)$ 
return  $((V^*, z^*), \text{VALF})$ 

```

4.2 Negative rank of outer approximation

One of the issues with Eq. (12) is that, being an outer approximation of Eq. (4), it does not ensure that $V \succeq 0$. In fact, we empirically found that, although the bound was often tight, the solution V^* had a considerable number of negative eigenvalues. We therefore propose to add the following cutting planes to the formulation: for each eigenvector p of V^* corresponding to a strictly negative eigenvalue we add the following inequality to the formulation Eq. (12):

$$p^\top V p \geq 0. \quad (16)$$

These cuts make V^* infeasible since, by definition of negative eigenvector, we have that $p^\top V^* p < 0$. Unfortunately, we found that in practice these cuts, by themselves, do not yield a PSD solution. Nevertheless, this procedure decreases the the number and/or the sum of the strictly negative eigenvalues.

The pseudocode for our outer approximation algorithm is given in Alg. 2. The function SPECTRALDECOMPOSITION() returns the spectral decomposition of its argument.

Algorithm 2 Iterative outer approximation

```

1: procedure OUTERACOPFMI
2:    $P \leftarrow \emptyset$ 
3:    $\lambda_- \leftarrow 1$ 
4:   while  $\lambda_- > 0$  and TIME < MAXTIME do
5:      $((V^*, z^*), \text{VALF}) \leftarrow$  optimal solution of (ACOPFGouter)
6:      $(W, A) \leftarrow$  SPECTRALDECOMPOSITION( $V^*$ )
7:      $\lambda_- \leftarrow$  number of negative eigenvalues
8:      $P_- \leftarrow$  eigenvectors associated to the negative eigenvalues
9:      $P \leftarrow P \cup P_-$ 
   return ( $X^*, z^*, \text{VALF}$ )

```

5 Dimensionality reduction

All our methods solve either inner or outer matrix approximations of the original MIQCQP Eq. (2), derived from Eq. (3) by relaxing the rank constraint. As such, they will not provide a solution vector v^* for the voltage in the original problem Eq. (2), but rather a solution matrix V^* , which will be very unlikely to have rank 1.

In order to heuristically extract a vector v^* from a matrix V^* with higher rank we considered two options for a dimensionality reduction algorithm: (a) Principal Component Analysis (PCA) [15] and (b) Barvinok’s naive algorithm [7, §5]. Both produce some estimate v' of v^* according to different analyses. We then used v' as a starting point for a local descent carried out by a local Nonlinear Programming (NLP) solver.

Both options are simple and effective in different settings. PCA is well known and needs no introduction. Barvinok’s algorithm is as follows:

1. factor V^* into $V^* = F F^\top$;
2. sample a random vector y componentwise from a standard Gaussian distribution $N(0, 1)$;
3. let $v' = Fy$.

The analysis carried out by A. Barvinok shows that v' has high probability of being “not too far” from the feasible set. We remark that we applied Barvinok’s algorithm to a wrong setting (in general), since the analysis only holds whenever V^* is a solution of an SDP, and some of our algorithms do not ensure PSD solutions. This being “just a heuristic”, we proceeded nonetheless, based on the fact that the formulations that produce V^* are themselves derived from SDP formulations.

Some empirical tests showed a promise of superiority of Barvinok’s naive algorithm to PCA (also see [19]), which was therefore our dimensionality reduction method of choice in obtaining computational results (Sect. 6).

6 Numerical results

We tested our approach using Cplex-12.6.3 [14] on YALMIP [20] for the inner and outer approximations. The MIQCQP was solved with Bonmin-4.8.4 [8] on AMPL [12]. Our results were obtained on an Intel i7 dual-core CPU at 2.1GHz with 15GB RAM.

The time limits for decreasing the negative rank (outer approximation) and feasible solution improvement (inner approximation) were set to 300 seconds. The limit of time for solving MIQCQPs was set to 1200 seconds.

Instance	Inner-approximation								Outer-approximation						MIQCQP
	feasible sol		optimal sol						first value	time	final value	itn	time	rank	best value
	itn	time	first value	time	value	itn	time	rank							
WB2	2	1.34	880.97	0.63	877.78	6	3.42	1	876.92	0.52	877.75	57	26.50	2	878.182
WB3	18	10.52	445.56	0.54	417.30	15	8.01	1	398.71	0.64	417.17	52	26.45	2	417.244
WB5	5	2.17	1209.96	0.42	946.69	76	31.82	1	676.76	0.42	945.96	115	72.93	2	946.584
6ww	50	37.55	3156.02	0.76	3156.02	1	0.76	2	2639.93	0.70	3009.20	155	224.86	2	3018.52
case9	18	12.90	2041.87	0.76	1272.64	45	33.89	2	980.00	0.87	994.96	135	301.00	11	1021.26
case14	46	62.96	10742.05	1.50	5778.53	37	55.56	2	4746.00	1.41	4746.00	67	301.72	28	5265.39
case30	-	-	-	-	-	-	-	-	285.77	4.69	285.78	13	317.62	60	344.11
case57	-	-	-	-	-	-	-	-	25460.00	16.91	25460.00	4	682.35	114	25460.00
case89pegase	-	-	-	-	-	-	-	-	5730.15	60.24	5730.15	3	643.17	178	(x)
case118	-	-	-	-	-	-	-	-	96520.00	110.52	96520.00	2	807.43	192	(x)
case300	-	-	-	-	-	-	-	-	392021.18	2944.12	392021.18	1	2944.12	430	(x)

Table 1. ACOFP inner and outer MILP relaxations.

As we can observe the bounds we obtained are tight with respect to the feasible solution we obtained for the MIQCQP formulation. In general, inner approximations seem to provide solutions with a smaller rank than outer ones. Unfortunately, we could not scale these experiments to larger sizes because of the slow convergence of the loop from Step 4 to Step 6 of Alg. 1. It is for this reason that we write “-” on some instances for the inner approximation. When we write “(x)” we mean that the local solver was not able to find a local optimum in the time limit.

References

1. Ahmadi, A., Hall, G.: Sum of squares basis pursuit with linear and second order cone programming. Tech. Rep. 1510.01597v1, arXiv (2015)
2. Ahmadi, A., Hall, G.: Sum of squares basis pursuit with linear and second order cone programming. Contemporary Mathematics (to appear)
3. Ahmadi, A., Majumdar, A.: Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization. Tech. Rep. 1706.02586v1, arXiv (2017)

4. Alizadeh, F.: Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization* 5(1), 13–51 (1995)
5. Bai, X., Wei, H., Fujisawa, K., Wang, Y.: Semidefinite programming for optimal power flow problems. *Electrical Power and Energy Systems* 30, 383–392 (2008)
6. Barker, G., Carlson, D.: Cones of diagonally dominant matrices. *Pacific Journal of Mathematics* 57(1), 15–32 (1975)
7. Barvinok, A.: Measure concentration in optimization. *Mathematical Programming* 79, 33–53 (1997)
8. Bonami, P., Lee, J.: **BONMIN** user’s manual. Tech. rep., IBM Corporation (June 2007)
9. Chen, C., Atamtürk, A., Oren, S.: Bound tightening for the alternating current optimal power flow problem. *IEEE Transactions on Power Systems* 31(5), 3729–3736 (2016)
10. Dias, G., Liberti, L.: Diagonally dominant programming in distance geometry. In: Cerulli, R., Fujishige, S., Mahjoub, R. (eds.) *International Symposium in Combinatorial Optimization*. LNCS, vol. 9849, pp. 225–236. Springer, New York (2016)
11. Fiala, J., Kočvara, M., Stingl, M.: **PENLAB**: A MATLAB solver for nonlinear semidefinite optimization. ArXiv e-prints (Nov 2013)
12. Fourer, R., Gay, D.: *The AMPL Book*. Duxbury Press, Pacific Grove (2002)
13. Gershgorin, S.: Über die abgrenzung der eigenwerte einer matrix. *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques et na* 6, 749–754 (1931)
14. IBM: **ILOG CPLEX 12.6 User’s Manual**. IBM (2014)
15. Jolliffe, I.: *Principal Component Analysis*. Springer, Berlin, 2nd edn. (2010)
16. Klerk, E.D.: *Aspects of Semidefinite Programming*. No. 65 in *Applied Optimization*, Kluwer, Dordrecht (2004)
17. Kuang, X., Ghaddar, B., Naoum-Sawaya, J., Zuluaga, L.: Alternative lp and socp hierarchies for acopf problems. *IEEE Transactions on Power Systems* 32(4), 2828 – 2836 (2017)
18. Lavaei, J., Low, S.: Zero duality gap in optimal power flow problem. *IEEE Transactions on Power Systems* 27(1), 92–107 (2012)
19. Liberti, L., Vu, K.: Barvinok’s naive algorithm in distance geometry. Tech. Rep. working paper, CNRS & Ecole Polytechnique (2017)
20. Löfberg, J.: **YALMIP**: A toolbox for modeling and optimization in MATLAB. In: *Proceedings of the International Symposium of Computer-Aided Control Systems Design*. CACSD, vol. 1. IEEE, Piscataway (2004)
21. Lubin, M., Yamangil, E., Bent, R., Vielma, J.: Extended formulations in mixed-integer convex programming. In: Louveaux, Q., Skutella, M. (eds.) *Integer Programming and Combinatorial Optimization (Proceedings of IPCO16)*. pp. 102–113. No. 9682 in LNCS, Springer, New York (2016)
22. Ruiz, M., Maeght, J., Marié, A., Panciatici, P., Renaud, A.: A progressive method to solve large-scale ac optimal power flow with discrete variables and control of the feasibility. In: *Power Systems Computation Conference (PSCC), 2014*. IEEE, Piscataway (2014)
23. Salgado, E.: *Fast relaxations for Alternating Current Optimal Power Flow*. Master’s thesis, MPRO, Université Paris-Saclay (2017)
24. Vavasis, S.: Quadratic programming is in NP. *Information Processing Letters* 36, 73–77 (1990)