

Optimal Power Flow as a Polynomial Optimization Problem

Bissan Ghaddar, Jakub Marecek, Martin Mevissen *

November 21, 2014

Abstract

Formulating the alternating current optimal power flow (ACOPF) as a polynomial optimization problem makes it possible to solve large instances in practice and to guarantee asymptotic convergence in theory.

We formulate the ACOPF as a degree-two polynomial program and study two approaches to solving it via convexifications. In the first approach, we tighten the first order relaxation of the non-convex quadratic program by adding valid inequalities. In the second approach, we exploit the structure of the polynomial program by using a sparse variant of Lasserre's hierarchy. This allows us to solve instances of up to 39 buses to global optimality and to provide strong bounds for the Polish network within an hour.

1 Introduction

Optimal Power Flow (OPF) in alternating current models (ACOPF) is one of the most important power system optimization problems. Various optimization methods have been widely used to tackle this hard problem [13, 24]. There are numerous extensions of the problem of widely varying tractability, including security-constrained variants taking into account uncertainty [3]. Even the ACOPF alone, however, is a large-scale non-convex non-linear optimization problem, and hence challenging to solve.

While non-linear formulations for OPF capture the system behavior more accurately than linearization, in principle, they pose a challenge for the solvers, which often fail to find the global optimum, or do not guarantee to have found the global optimum. A great variety of relaxations and solution methods to solve the OPF problem has been tested, including non-linear programs, piece-wise linearization, Lagrangian relaxations, genetic algorithms, and interior point methods. For examples, please see surveys [19, 15]. A recent line of research proposed by Bai et al. [1] applied semidefinite programming (SDP) to the OPF problem. Lavaei et al. [12, 20] then showed that the solution of the SDP is the global optimum, under some conditions. Several follow-up computational studies [6, 16] increased the dimension of SDP instances that can be solved, in practice.

Although there has been no single formulation and solution approach suitable for all the various forms of OPF problems, many OPF formulations take the form of a polynomial programming (PP), where the objective, equality constraints and inequality constraints are all given by multivariate polynomials. Equality constraints typically include the power flow network equations and

*IBM Research - Ireland, Damastown Industrial Estate, Mulhuddart, Dublin 15. email: bghaddar@ie.ibm.com, jakub.marecek@ie.ibm.com, martmevi@ie.ibm.com

balance constraints. The inequality constraints often include active/reactive power generation limits, demand constraints, bus voltage limits, and branch flow limits. Using polynomial optimization, one can model the network more accurately, and obtain globally valid lower bounds and globally optimal solutions, under mild conditions. Notably, one can use a wide variety of objective functions, and incorporate further constraints easily, without affecting convergence properties.

For example, besides the minimization of power generation costs, other objectives can be formulated using PP, including minimization of power generation costs with unit commitment costs, minimization of system losses, and maximization of power quality (minimizing voltage deviation). Computationally, one uses a hierarchy of SDP relaxations of Lasserre [10] to convexify the PP problem. This approach was used for the OPF problem to improve the Lavaei-Low bounds (See [17, 7]). Unfortunately, the dimension of these relaxations grow rapidly with the size of the power system, posing a major computational challenge.

In this paper, we present two techniques for tackling the OPF problem. The first technique uses “cutting surfaces” of Ghaddar et al. [4], which are valid inequalities, generated dynamically upon violation at each step of the algorithm. Instead of increasing the degree of the non-negative certificates, as in Lasserre’s hierarchy, the set of polynomial inequalities describing the feasible region of the polynomial program is changed in each iteration, while the degree of the polynomials is fixed. These valid inequalities yield stronger convexifications and hence tighter bounds than the Lavaei-Low [12] SDP relaxation.

The second technique uses the sparse hierarchy of SDP relaxations of Waki et al. [22], which improves the tractability of the Lasserre’s hierarchy by exploiting sparsity of the OPF problem. The relaxations are equivalent to the Lavaei-Low [12] SDP relaxation, where it is exact, and provide tighter relaxations, where it is not (i.e., as the level of the hierarchy increases). Further, we employ matrix completion techniques [8] to break down the largest SDP matrix at the price of introducing additional equality constraints and several smaller matrix inequalities, to make the approach scale to power systems with thousands of buses.

Overall, the main contributions of the paper are:

- stronger convexifications for the OPF problem than those presented in the literature
- larger instances than those published in the literature are solved to global optimality by exploiting structured sparsity of the OPF problem
- proof of convergence of the sparse hierarchy of SDP relaxations for OPF.

Notably, either of the presented techniques improves upon the Lavaei-Low SDP relaxation, whenever the Lavaei-Low SDP relaxation does not provide the global optimum.

2 Optimal Power Flow Problem

We use the same notation as in [12] and [16]. The topology of the power system $P = (N, E)$ is represented as an undirected graph, where each vertex $n \in N$ is called a “bus” and each edge $e \in E$ is called a “branch”. We use $|N|$ to denote the number of buses and $|E|$ to denote the number of branches. Let $G \subseteq N$ be the set of generators and $E \subseteq N \times N$ be the set of all branches modeled as Π -equivalent circuits. The matrix $y \in \mathbb{R}^{|N| \times |N|}$ represents the network admittance matrix, whose sparsity pattern is the same as that of the adjacency matrix of P . \bar{b}_{lm} is the value of the shunt element at branch $(l, m) \in E$ and $g_{lm} + jb_{lm}$ is the series admittance on a branch (l, m) . Let $S_k^d = P_k^d + jQ_k^d$ be the active and reactive load (demand) at each bus $k \in N$ and $P_k^g + jQ_k^g$ represent the apparent power of the generator at bus $k \in G$. Define $V_k = \Re V_k + j\Im V_k$

as the voltage at each bus $k \in N$ and $S_{lm} = P_{lm} + jQ_{lm}$ as the apparent power flow on the line $(l, m) \in E$. The edge set $L \subseteq E$ contains the branches (l, m) such that the apparent power flow limit is less than a certain given tolerance ε .

2.1 Formulation

We focus on the rectangular power-voltage formulation, where

- P_k^{\min} and P_k^{\max} are the limits on active generation capacity at bus k , where $P_k^{\min} = P_k^{\max} = 0$ for all $k \in N/G$.
- Q_k^{\min} and Q_k^{\max} are the limits on reactive generation capacity at bus k , where $Q_k^{\min} = Q_k^{\max} = 0$ for all $k \in N/G$.
- V_k^{\min} and V_k^{\max} are the limits on the absolute value of the voltage at a given bus k .
- S_{lm}^{\max} is the limit on the absolute value of the apparent power of a branch $(l, m) \in L$.

Let e_k be the k^{th} standard basis vector in $\mathbb{R}^{|N|}$, similar to [12], the following matrices are defined

$$\begin{aligned} y_k &= e_k e_k^T y, \\ y_{lm} &= (j \frac{\bar{b}_{lm}}{2} + g_{lm} + j b_{lm}) e_l e_l^T - (g_{lm} + j b_{lm}) e_l e_m^T, \\ Y_k &= \frac{1}{2} \begin{bmatrix} \Re(y_k + y_k^T) & \Im(y_k^T - y_k) \\ \Im(y_k - y_k^T) & \Re(y_k + y_k^T) \end{bmatrix}, \\ \bar{Y}_k &= -\frac{1}{2} \begin{bmatrix} \Im(y_k + y_k^T) & \Re(y_k - y_k^T) \\ \Re(y_k^T - y_k) & \Im(y_k + y_k^T) \end{bmatrix}, \\ M_k &= \begin{bmatrix} e_k e_k^T & 0 \\ 0 & e_k e_k^T \end{bmatrix}, \\ Y_{lm} &= \frac{1}{2} \begin{bmatrix} \Re(y_{lm} + y_{lm}^T) & \Im(y_{lm}^T - y_{lm}) \\ \Im(y_{lm} - y_{lm}^T) & \Re(y_{lm} + y_{lm}^T) \end{bmatrix}, \\ \bar{Y}_{lm} &= -\frac{1}{2} \begin{bmatrix} \Im(y_{lm} + y_{lm}^T) & \Re(y_{lm}^T - y_{lm}) \\ \Re(y_{lm}^T - y_{lm}) & \Im(y_{lm} + y_{lm}^T) \end{bmatrix}. \end{aligned}$$

Let x be a vector of variables defined as $x := [\Re V_k \ \Im V_k]^T$, and let the cost of power generation be $\sum_{k \in G} f_k(P_k^g)$ where $f_k(P_k^g) = c_k^2 (P_k^g)^2 + c_k^1 P_k^g + c_k^0$, with c_k^2, c_k^1, c_k^0 non-negative. The classical OPF problem can be written as a polynomial optimization problem of degree 4,

$$\begin{aligned} \min \sum_{k \in G} f_k(x) & \tag{OP4} \\ \text{s.t. } P_k^{\min} &\leq \text{tr}(Y_k x x^T) + P_k^d \leq P_k^{\max} \\ Q_k^{\min} &\leq \text{tr}(\bar{Y}_k x x^T) + Q_k^d \leq Q_k^{\max} \\ (V_k^{\min})^2 &\leq \text{tr}(M_k x x^T) \leq (V_k^{\max})^2 \\ (\text{tr}(Y_{lm} x x^T))^2 &+ (\text{tr}(\bar{Y}_{lm} x x^T))^2 \leq (S_{lm}^{\max})^2 \end{aligned}$$

The objective function often is the cost of power generation where

$$f_k(x) := \left(c_k^2 (P_k^d + \text{tr}(Y_k x x^T))^2 + c_k^1 (P_k^d + \text{tr}(Y_k x x^T)) + c_k^0 \right)$$

. The constraints, in turn, impose a limitation on the active and reactive power, restrict the voltage on a given bus, and limit the apparent power flow at each end of a given line. By defining variable $W = xx^T$, [12] reformulates the problem as a rank-constrained problem. Subsequently, one can drop the rank constraint to obtain the SDP relaxation [OP-SDP] as in [12].

3 Polynomial Programming Approach

The OPF problem is a particular case of a polynomial optimization problem of the form:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0 \quad i = \{1, \dots, m\} \end{aligned} \quad [\text{PP}]$$

Motivated by the seminal work of Lasserre [10], there has been a lot of recent research activity to devise solution schemes to solve polynomial optimization problems. The main idea of these schemes is based on applying representation theorems from algebraic geometry to characterize the set of polynomials that are nonnegative on a given domain. Given $S \subseteq \mathbb{R}^n$, define $\mathcal{P}_d(S)$ to be the cone of polynomials of degree at most d that are non-negative over S . We use Σ_d to denote the cone of polynomials of degree at most d that are sum-of-squares of polynomials. Using $\mathcal{G} = \{g_i(x) : i = 1, \dots, m\}$ and denoting $S_{\mathcal{G}} = \{x \in \mathbb{R}^n : g(x) \geq 0, \forall g \in \mathcal{G}\}$ the basic closed semi-algebraic set defined by \mathcal{G} , we can rephrase [PP] as

$$\begin{aligned} \max \quad & \varphi \\ = \max \quad & \varphi \end{aligned} \quad \begin{aligned} \text{s.t.} \quad & f(x) - \varphi \geq 0 \quad \forall x \in S_{\mathcal{G}}, \\ \text{s.t.} \quad & f(x) - \varphi \in \mathcal{P}_d(S_{\mathcal{G}}). \end{aligned} \quad [\text{PP-D}]$$

Although [PP-D] is a conic problem, it is not known how to optimize over the cone $\mathcal{P}_d(S_{\mathcal{G}})$ efficiently. Lasserre [10] introduced a hierarchy of SDP relaxations corresponding to liftings of polynomial problems into higher dimensions. In the hierarchy of SDP relaxations, one convexifies the problem, obtains progressively stronger relaxations, but the size of the SDP instances soon becomes computationally challenging. Under assumptions slightly stronger than compactness, the optimal values of these problems converge to the global optimal value of the original problem, [PP].

The approximation of $\mathcal{P}_d(S_{\mathcal{G}})$ used in [10] is the cone $\mathcal{K}_{\mathcal{G}}^r$, where

$$\mathcal{K}_{\mathcal{G}}^r = \Sigma_r + \sum_{i=1}^m g_i(x) \Sigma_{r-\text{deg}(g_i)}, \quad (1)$$

and $r \geq d$. The corresponding optimization problem over S can be written as:

$$\begin{aligned} \max_{\varphi, \sigma_i(x)} \quad & \varphi \\ \text{s.t.} \quad & f(x) - \varphi = \sigma_0(x) + \sum_{i=1}^m \sigma_i(x) g_i(x) \\ & \sigma_0(x) \in \Sigma_r, \sigma_i(x) \in \Sigma_{r-\text{deg}(g_i)}. \end{aligned} \quad [\text{PP-H}_r]^*$$

$[\text{PP-H}_r]^*$ can be reformulated as a semidefinite optimization problem. We denote the dual of $[\text{PP-H}_r]^*$ by $[\text{PP-H}_r]$. The computational cost of the problem clearly depends on both the degree of the polynomials, r , and the dimension of the problem. The number of constraints can be large, especially when many variables and high-degree polynomials are used. Based on the described approach, Molzahn and Hiskens [17] and Josz et al. [7] used $[\text{OP}_4]$ and applied Lasserre's hierarchy to obtain global optimality on instances with up to 5 and 10 buses respectively, where Lavaei-Low is not globally optimal.

3.1 Relationship with Lavaei-Low Formulation

In this work, instead of starting with $[\text{OP}_4]$ and applying the hierarchy $[\text{PP-H}_r]^*$, we reduce the OPF problem to a polynomial program of degree 2:

$$\begin{aligned}
\min \sum_{k \in G} & \left(c_k^2 (P_k^g)^2 + c_k^1 (P_k^d + \text{tr}(Y_k x x^T)) + c_k^0 \right) \\
P_k^{\min} & \leq \text{tr}(Y_k x x^T) + P_k^d \leq P_k^{\max} \\
Q_k^{\min} & \leq \text{tr}(\bar{Y}_k x x^T) + Q_k^d \leq Q_k^{\max} \\
(V_k^{\min})^2 & \leq \text{tr}(M_k x x^T) \leq (V_k^{\max})^2 \\
P_{lm}^2 + Q_{lm}^2 & \leq (S_{lm}^{\max})^2 \\
P_k^g & = \text{tr}(Y_k x x^T) + P_k^d \\
P_{lm} & = \text{tr}(Y_{lm} x x^T) \\
Q_{lm} & = \text{tr}(\bar{Y}_{lm} x x^T).
\end{aligned} \tag{OP}_2$$

Theorem 1. *The first level of the hierarchy for $[\text{OP}_2]$, $[\text{OP}_2\text{-H}_1]^*$, is equivalent to the dual of $[\text{OP-SDP}]$, i.e., Optimization 4 in [12].*

Hence, the first level of the hierarchy for $[\text{OP}_2]$ (i.e., $r = 2$), $[\text{OP}_2\text{-H}_1]^*$ provides the same bound as $[\text{OP-SDP}]$ while the first level of the hierarchy for $[\text{OP}_4]$ (i.e., $r = 4$), provides a bound that is at least as good as $[\text{OP-SDP}]$.

3.2 Inequality Generation Approach

Realising that ACOPF can be modeled as a polynomial program, we aim to tackle the problem using methods recently developed in polynomial optimization. The first method, discussed in this section, is a dynamic approach, which generates valid, but violated inequalities at each step of the algorithm. The idea is based on the work of Ghaddar et al. [5], who proposed the dynamic inequality generation scheme (DIGS) for general PP. In DIGS, the current solution is used to generate polynomial inequalities that are valid on the feasible region of the PP problem. This iterative scheme makes it possible to generate improving approximations without growing the degree of the certificates involved, and hence the size of the SDP problem.

In this section, we use the first level of the relaxation of $[\text{OP}_2]$, i.e. $[\text{OP}_2\text{-H}_1]$, and add valid quadratic inequalities of the form $p(x) \geq 0$. The polynomial $p(x)$ needs to be a valid inequality and at the same time improve on the bound of the relaxation. This can be translated as $p \in \mathcal{P}_d(S) \setminus \mathcal{K}_{\mathcal{G}}^d$, where $d = 2$, the degree of $[\text{OP}_2]$, in this case. The iterative scheme can be summarized as follows:

- Start with $\mathcal{G}_0 = \mathcal{G}$
- Given \mathcal{G}_i let $p_i \in \mathcal{P}_d(S_{\mathcal{G}}) \setminus \mathcal{K}_{\mathcal{G}_i}^d$, define $\mathcal{G}_{i+1} = \mathcal{G}_i \cup \{p_i\}$.

To be able to generate a polynomial $p(x)$, the scheme consists of a master problem and a sub-problem. The master problem is of the same form as $[\text{PP-H}_r]^*$ with the hierarchy level (i.e., r) being fixed to d :

$$\begin{aligned}
\max \varphi & & \text{s.t. } f(x) - \varphi & \in \mathcal{K}_{\mathcal{G}}^d, & & [\text{PP-M}]
\end{aligned}$$

where $\mathcal{K}_{\mathcal{G}}^d$ is as defined in (1) with r fixed to d . The master problem provides lower bounds. The subproblem uses the optimal dual information from the master problem, Y , to generate polynomial inequalities that are valid on the feasible region:

$$\min_p \langle p, Y \rangle \quad \text{s.t. } p \in \mathcal{K}_{\mathcal{G}}^{d+2} \cap \mathbf{R}_d[x]. \quad [\text{PP-S}]$$

These valid inequalities are then incorporated into the master problem, to construct new non-negativity certificates, obtaining better approximations of the OPF. The iterative scheme terminates when the objective function of the subproblem is sufficiently close to 0 [4].

Considering [OP₂-H₁] is equivalent to the Lavaei-Low SDP relaxation, as seen in Section 3.1, one can improve on the Lavaei-Low bound by adding valid inequalities to [OP₂]. In some cases, one can obtain the global optimum. As opposed to the approach proposed in [7, 17], where the hierarchy level, r , is increased ([PP-H_{*r*}]^{*}), in this case, r is fixed to d . That is: Instead of increasing the degree of the non-negative certificates, the degree of the polynomials is fixed and the set of polynomial inequalities describing the feasible region of the polynomial program is increased. Valid inequalities are used to construct new certificates that provide better approximations and hence provide stronger convexification at each iteration. Consequently, the relaxation is improving at each stage of the algorithm and hence better bounds are obtained at each iteration, while sizes of the positive semidefinite matrices and the numbers of constraints can be significantly lower as compared to [PP-H_{*r*}]^{*}.

3.3 Exploiting OPF Structure

The current scalability of state-of-the-art SDP solvers limits the tractability of the Lasserre hierarchy even for medium-scale polynomial programs.

One approach to improve the tractability of the Lasserre hierarchy is to exploit *correlative* sparsity of a polynomial optimization problem [PP] of dimension n due to Waki et al. [22], which can be represented by the symbolic $n \times n$ *correlative sparsity pattern matrix* R , defined by

$$R_{ij} = \begin{cases} \star & \text{for } i = j \\ \star & \text{for } x_i, x_j \text{ in the same monomial of } f \\ \star & \text{for } x_i, x_j \text{ in the same constraint } g_k \\ 0 & \text{otherwise,} \end{cases}$$

and its associated adjacency graph G , the *correlative sparsity pattern graph*. Let $\{I_k\}_{k=1}^p$ be the set of maximal cliques of a chordal extension of G following the construction in [22], i.e. $I_k \subset \{1, \dots, n\}$. Given that a chordal extension of arbitrary graphs is not unique, it is important to choose a chordal extension which can be computed efficiently while keeping the number of additional edges as small as possible, since the size of matrix inequalities in the sparse hierarchy is determined by the cardinality of the maximal cliques I_1, \dots, I_p . Note, that the ordering \mathcal{O} applied to R determines the chordal extension of R , and hence the number and cardinality of the maximal cliques $\{I_k\}_{k=1}^p$. The problem of minimising the size of the sparse hierarchy of SDP relaxations for [PP] is therefore equivalent to finding the ordering which results in a symbolic Cholesky factorisation with the minimal number of fill-ins, or finding the chordal extension of G with the minimal number of edges added. While the problem of finding the minimal chordal extension is NP-hard, a number of heuristics for orderings have been proposed that aim to keep the number of fill-ins in the symbolic Cholesky factorisation such as the symmetric approximate

minimum degree ordering. The sparse approximation of $\mathcal{P}_d(S)$ is $\mathcal{K}_{\mathcal{G}}^r(I)$, given by

$$\mathcal{K}_{\mathcal{G}}^r(I) = \sum_{k=1}^p \left(\Sigma_r(I_k) + \sum_{j \in J_k} g_j \Sigma_{r-\deg(g_j)}(I_k) \right),$$

where $\Sigma_d(I_k)$ is the set of all sum-of-squares polynomials of degree up to d supported on I_k and (J_1, \dots, J_p) is a partitioning of the set of polynomials $\{g_j\}_j$ defining S such that for every j in J_k , the corresponding g_j is supported on I_k . The support $I \subset \{1, \dots, n\}$ of a polynomial contains the indices i of terms x_i which occur in one of the monomials of the polynomial. The sparse hierarchy of SDP relaxations is then given by

$$\begin{aligned} & \max_{\varphi, \sigma_k(x), \sigma_{r,k}(x)} \varphi && \text{[PP-SH}_r\text{]}^* \\ \text{s.t. } & f(x) - \varphi = \sum_{k=1}^p \left(\sigma_k(x) + \sum_{j \in J_k} g_j(x) \sigma_{j,k}(x) \right) \\ & \sigma_k \in \Sigma_r(I_k), \sigma_{j,k} \in \Sigma_{r-\deg(g_j)}(I_k). \end{aligned}$$

We denote the dual of $\text{[PP-SH}_r\text{]}^*$ by $\text{[PP-SH}_r\text{]}$. In the case that R is sparse, i.e., $|I_k| \ll n$, then the resulting matrix variables are of size $\binom{|I_k|+r}{r}$, instead of $\binom{n+r}{r}$. While $\text{[PP-SH}_r\text{]}^*$ provides a weaker relaxation to [PP] than $\text{[PP-H}_r\text{]}^*$ for a fixed relaxation order r in general, the asymptotic convergence result for the dense hierarchy extends to the sparse case:

Assumption 1. *Let S denote the feasible set of a problem of form [PP] . Let $\{I_k\}_k$ denote the p maximal cliques of a chordal extension of the sparsity pattern graph of the [PP] .*

- (i) *Then, there is a $M > 0$ such that $\|x\|_{\infty} < M$ for all $x \in S$.*
- (ii) *Ordering conditions for index sets as in Assumption 3.2 (i) and (ii) in [11].*
- (iii) *Running-intersection-property, c.f. [11], holds for $\{I_k\}_k$.*

Remark 1. *Note, that if S is compact, it is easy to add up to p redundant quadratic inequality to the definition of S , s.t. Assumption 1 (i) is satisfied. (ii) can be satisfied by construction and re-ordering of the sets $\{I_k\}_k$. The running-intersection-property is satisfied for the maximal cliques of a chordal graph, as pointed out in [9]. Thus, Assumption 1 is satisfied for both, $\text{[OP}_2\text{]}$ and $\text{[OP}_4\text{]}$.*

Now, we can formulate the convergence result.

Proposition 1 (Asymptotic Convergence). *If Assumption 1 holds for the feasible sets of $\text{[OP}_2\text{]}$ and $\text{[OP}_4\text{]}$ respectively, then for the sparse hierarchy $\text{[OP}_2\text{-SH}_r\text{]}^*$ for $\text{[OP}_2\text{]}$ and $\text{[OP}_4\text{-SH}_r\text{]}^*$ for $\text{[OP}_4\text{]}$ and their respective duals the following holds:*

- (a) $\inf \text{[OP}_2\text{-SH}_r\text{]} \nearrow \min(\text{[OP}_2\text{]})$ as $r \rightarrow \infty$,
 $\inf \text{[OP}_4\text{-SH}_r\text{]} \nearrow \min(\text{[OP}_4\text{]})$ as $r \rightarrow \infty$.
- (b) $\sup \text{[OP}_2\text{-SH}_r\text{]}^* \nearrow \min(\text{[OP}_2\text{]})$ as $r \rightarrow \infty$,
 $\sup \text{[OP}_4\text{-SH}_r\text{]}^* \nearrow \min(\text{[OP}_4\text{]})$ as $r \rightarrow \infty$.
- (c) (i) *If the interior of the feasible set of $\text{[OP}_4\text{]}$ is nonempty, there is no duality gap between $\text{[OP}_4\text{-SH}_r\text{]}$ and $\text{[OP}_4\text{-SH}_r\text{]}^*$.*
(ii) *There is no duality gap between $\text{[OP}_2\text{-SH}_1\text{]}$ and $\text{[OP}_2\text{-SH}_1\text{]}^*$.*

(d) If $[OP_2]$ has a unique global minimizer, x^* , then as r tends to infinity the components of the optimal solution of $[OP_2-SH_r]$ corresponding to the linear terms converge to x^* (an analogous result holds for $[OP_4]$).

Proof. (a), (c) and (d) are a Corollary of Theorem 3.6 of Lasserre [11], (b) (i) follows from Theorem 5 of [9], (b) (ii) from note in [22] on primal and dual sparse SDP relaxation of order 1 for quadratic optimization problems. \square

Moreover, for $[OP_2]$ the following proposition holds.

Proposition 2. *The sparse SDP relaxation $[OP_2-SH_1]^*$ of order one for $[OP_2]$ is equivalent to the first order relaxation of the dense Lasserre hierarchy $[OP_2-H_1]^*$ for $[OP_2]$ and the Lavaei-Low dual sdp relaxation.*

Proof. Follows from the fact that sparse and dense SDP relaxation of order 1 are equivalent for non-convex quadratic optimization problems, as proven in Section 4.5 of Waki et al. [22] and the Theorem 1. \square

Remark 2. *For a fixed order r , the sparse hierarchy $[OP_2-SH_1]$ has $O(\kappa^{2r})$ variables, where κ is the maximum number of variables appearing in the objective or an inequality constraint of $[PP]$. The largest matrix inequality is of size $O(\kappa^r)$. This is in contrast to $O(n^{2r})$ variables and matrix variables of size $O(n^r)$ in the dense hierarchy $[OP_2-H_1]$. In case the $[PP]$ is very sparse, i.e., $\kappa \ll n$, the size of the sparse hierarchy is vastly smaller than the dense one.*

4 Numerical Results

In order to illustrate the performance of the approaches, we improve the relaxation $[OP_2-H_1]^*$ iteratively using the inequality generation scheme (we refer to as DIGS), in some cases to proven optimality. Next, we exploit the sparsity of the polynomial program to solve higher order relaxations, proving global optimality for further instances.¹ Figure 1 summarizes the formulations and relaxations discussed in this paper.

4.1 Small-scale Instances

First, we use three small test cases, devised by Bukhsh et al. [2] and Lesieutre et al. [14] such that the Lavaei-Low SDP relaxation is not optimal. In Tables 1–3, bold entries indicate that our approaches prove global optimality.

Example 1. *The first example is an instance with two buses [2], where the maximum voltage on the second bus varies from 0.976 to 1.028 (see Table 1). From Figure 2, it can be seen that after adding one inequality (i.e., one iteration of DIGS), the Lavaei-Low bound improves significantly. The optimal value is obtained in 8 iterations. For none of the 9 instances, MATPOWER converged and the solution MATPOWER found was far from the optimal. For example, for $V_2^{\max}=1.022$, the value of 713.27 is reported.*

¹The two techniques are implemented in MATLAB running on a PC with a 3.5Ghz processor, running Red Hat Linux. The DIGS approach is implemented using APPS [4]. SparseColO was used for exploiting sparsity in the master and the subproblem. Exploiting sparsity of polynomial program $[OP_4]$ is done using SparsePoP [23]. To solve the resulting SDP relaxation for both approaches, SeDuMi [21] is used as the SDP solver. All test instances are taken from [2] and [18]. For the polish instances the formulation in [16] is used to take into account multiple generators and transformers with off-nominal voltage ratios and phase shifts.

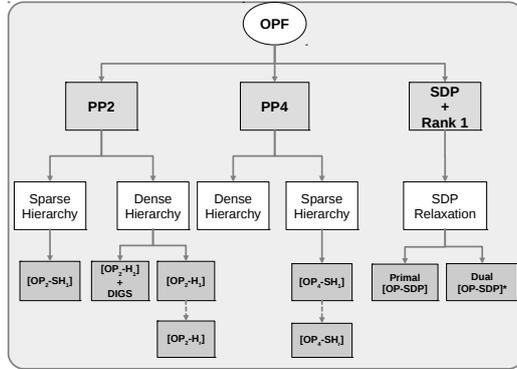


Figure 1: The different formulations and their relaxations.

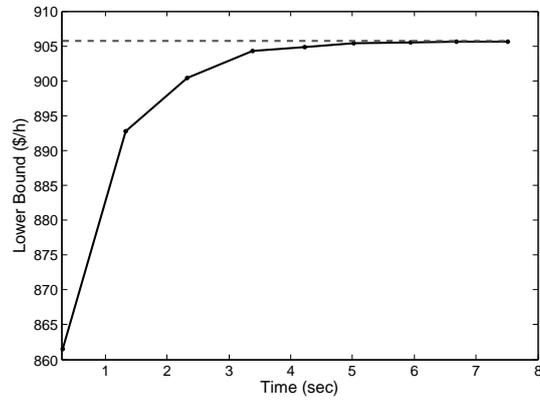


Figure 2: WB2 Bounds for $V_2^{\max}=1.022$.

Table 1: WB2 computational results.

V_2^{\max}	DIGS			[OP ₂ -SH ₁]		[OP ₄ -SH ₁]		[OP ₄ -SH ₂]	
	Iter	s	s Time	Bound	Time	Bound	Time	Bound	Time
0.976	905.76	1	0.9	905.76	0.2	905.76	0.4		
0.983	905.73	6	5.1	903.12	0.2	905.73	1.8		
0.989	905.73	6	4.3	900.84	0.1	905.72	1.7	905.73	1.8
0.996	905.73	6	4.6	898.17	0.2	905.73	1.4	905.73	1.6
1.002	905.73	6	4.8	895.86	0.1	905.72	1.8	905.73	1.5
1.009	905.73	8	6.4	893.16	0.2	905.71	1.9	905.73	0.6
1.015	905.73	6	4.7	890.82	0.1	905.71	0.8	905.73	0.6
1.022	905.73	8	6.5	888.08	0.1	905.71	2.6	905.73	1.7
1.028	905.73	8	5.1	885.71	0.1	904.59	0.8	905.73	0.8

Table 2: LMBM3 computational results.

S_{23}^{\max}	DIGS			[OP ₂ -SH ₁]		[OP ₄ -SH ₁]	
	Iter	s	s Time	Bound	Time	Bound	Time
28.35	10294.88	7	13.3	6307.97	0.2	10294.88	1.0
31.16	8179.99	6	11.2	6206.78	0.2	8179.99	0.7
33.96	7414.94	5	19.2	6119.71	0.2	7414.94	0.8
36.77	6895.19	5	19.5	6045.33	0.3	6895.19	0.7
39.57	6516.17	5	19.8	5979.38	0.2	6516.17	0.7
42.38	6233.31	5	18.1	5919.12	0.2	6233.31	0.7
45.18	6027.07	5	19.3	5866.68	0.1	6027.07	0.8
47.99	5882.67	3	12.1	5819.02	0.2	5882.67	0.7
50.79	5792.02	2	9.2	5779.34	0.3	5792.02	0.7
53.60	5745.04	1	0.7	5745.04	0.2	5745.04	0.8

Example 2. The second example is LMBM3 with 3 buses [14]. The results from Table 2 indicate that both approaches are successful in providing the optimal solution of these problems when Lavaei-Low relaxation is not exact. In this example, MATPOWER reported the optimal solution for all 10 instances.

Example 3. The last example consists of 5 buses [2]. The results presented in Table 3 are consistent with the previous two examples. Applying DIGS and exploiting sparsity solved all test cases to optimality. MATPOWER provided the optimal solution for all 8 instances.

4.2 Large-scale Instances

Next, we consider medium- and large-scale instances distributed with MATPOWER [18]. Table 4 presents MATPOWER objective function value in addition to computational results for [OP₂-H₁]* with SparseColO and [OP₄-SH₁] using SparsePoP. [OP₂-H₁]* captures the Lavaei-Low dual relaxation and obtains the same bounds and has similar computational performance. The computational time of [OP₂-H₁]* can be significantly improved using SparseColO which utilizes domain-space sparsity of a semidefinite matrix variable and range-space sparsity of a linear matrix inequality constraint. It is used as a preprocessor, which reduces the dimension of matrix variables in an SDP relaxation before applying SeDuMi. For instances larger than 39 buses only

Table 3: WB5 computational results.

Q_5^{\min}	DIGS			[OP ₂ -SH ₁]		[OP ₄ -SH ₁]	
	Iter	s	Time	Bound	Time	Bound	Time
-20.51	1146.48	3	28.4	954.82	0.3	1146.48	25.5
-10.22	1209.11	4	32.6	963.83	0.3	1209.11	17.1
0.07	1267.79	5	49.0	972.80	0.2	1267.44	26.3
10.36	1323.86	5	49.4	981.89	0.4	1323.86	20.9
20.65	1377.97	4	39.1	990.95	0.2	1377.97	15.4
30.94	1430.54	4	40.1	1005.13	0.3	1430.54	20.9
41.23	1481.81	5	49.6	1033.07	0.3	1481.81	14.6
51.52	1531.97	5	49.7	1070.39	0.5	1531.97	18.2

Table 4: Computational results for IEEE and Polish network instances.

Instance	MATPOWER Objective	[OP ₂ -H ₁]* + SparseColO			[OP ₄ -SH ₁]		
		Bound	Dim	Time	Bound	Dim	Time
case9mod	4267.07	2753.23	588×168	0.6	3087.89	1792×14847	17.5
case14mod	7806.10	7792.72	888×94	0.9	7991.07	7508×66740	904.2
case30mod	623.01	576.89	4706×684	3.8	578.56	36258×49164	13740.0
case39	41864.18	41862.08	7282×758	2.2	41864.18	26076×215772	4359.1
case57	41737.79	41737.79	13366×356	3.2	*	*	
case118	129660.69	129654.62	56620×816	6.1	*	*	
case300	719725.08	719711.63	362076×1938	13.6	*	*	
case2383wp	1.869×10^6	1.814×10^6	22778705×47975	3731.5	*	*	
case2736sp	1.308×10^6	1.307×10^6	30019740×57408	3502.2	*	*	

[OP₂-H₁]* can be solved, as [OP₄-SH₁] becomes computationally expensive for SparsePoP. Using DIGS, optimality of case9mod is proven in 3 hours and 7 iterations. For case14mod, DIGS performed 2 iterations within 5 hours and improved upon the Lavaei-Low bound. For instances up to 2736 buses one can solve [OP₂-H₁]* within an hour, but the generation of cutting surfaces becomes too consuming.

5 Conclusion

In this work, we propose to formulate the optimal power flow as a polynomial programming problem and present two techniques for deriving relaxations stronger than that of Lavaei and Low [12]. For several instances on up to 39 buses, where the Lavaei-Low relaxation is not exact, we provide globally optimal solutions for the first time. Furthermore, we show that the strong relaxations are tractable for medium- and large-scale instances.

The approaches are rather general. They make it possible to use arbitrary multivariate polynomials in the objective function and constraints, without the need to change the solver. Additionally, binary variables can be included, e.g., to model discrete decisions in transmission switching.

References

- [1] X. Bai, H. Wei, K. Fujisawa, and Y. Wang. Semidefinite programming for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, 30(6):383–392, 2008.
- [2] W. A. Bukhsh, A. Grothey, K. I. McKinnon, and P. Trodden. Local solutions of optimal power flow. *Power Systems, IEEE Transactions on*, 28(4):4780–4788, 2013.
- [3] F. Capitanescu, J. L. Martinez Ramos, P. Panciatici, D. Kirschen, A. Marano Marcolini, L. Platbrood, and L. Wehenkel. State-of-the-art, challenges, and future trends in security constrained optimal power flow. *Electric Power Systems Research*, 81(8):1731–1741, 2011.
- [4] B. Ghaddar. *New Conic Optimization Techniques for Solving Binary Polynomial Programming Problems*. PhD thesis, University of Waterloo, 2011.
- [5] B. Ghaddar, J. Vera, and M. Anjos. An iterative scheme for valid polynomial inequality generation in binary polynomial programming. In *Integer Programming and Combinatorial Optimization*, volume 6655 of *Lecture Notes in Computer Science*, pages 207–222. 2011.
- [6] R. A. Jabr. Exploiting sparsity in SDP relaxations of the opf problem. *Power Systems, IEEE Transactions on*, 27(2):1138–1139, 2012.
- [7] C. Jozs, J. Maeght, P. Panciatici, and J. C. Gilbert. Application of the moment-SOS approach to global optimization of the opf problem. arXiv:1311.6370v1, 2013.
- [8] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita. Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. *Mathematical Programming*, 129(1):33–68.
- [9] M. Kojima and M. Muramatsu. A note on sparse SOS and SDP relaxations for polynomial optimization problems over symmetric cones. *Computational Optimization and Applications*, 42(1):31–41, 2009.
- [10] J. Lasserre. Global optimization problems with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11:796–817, 2001.
- [11] J. Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):882–843, 2006.
- [12] J. Lavaei and S. Low. Zero duality gap in optimal power flow problem. *Power Systems, IEEE Transactions on*, 27(1):92–107, 2012.
- [13] J. Lavaei, D. Tse, and B. Zhang. Geometry of power flows and optimization in distribution networks. *Power Systems, IEEE Transactions on*, 29(2):572–583, 2014.
- [14] B. Lesieutre, D. Molzahn, A. Borden, and C. DeMarco. Examining the limits of the application of semidefinite programming to power flow problems. In *49th Annual Allerton Conference*, pages 1492–1499, 2011.
- [15] S. H. Low. Convex relaxation of optimal power flow part i: Formulations and equivalence. *IEEE Transactions on Control of Network Systems*, 1(1):15–27, 2014.

- [16] D. Molzahn, J. Holzer, B. Lesieutre, and C. DeMarco. Implementation of a large-scale optimal power flow solver based on semidefinite programming. *IEEE Transactions on Power Systems*, 28(4):3987–3998, 2013.
- [17] D. K. Molzahn and I. A. Hiskens. Moment-based relaxation of the optimal power flow problem. arXiv:1312.1992v2, 2013.
- [18] U. of Washington. Power systems test case archive.
- [19] K. Pandya and S. Joshi. A survey of optimal power flow methods. *Journal of Theoretical & Applied Information Technology*, 4(5):450–458, 2008.
- [20] S. Sojoudi and J. Lavaei. Semidefinite relaxation for nonlinear optimization over graphs with application to power systems. *Preprint*, 2013.
- [21] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12, 1999.
- [22] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.
- [23] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. SparsePOP : a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software*, 35(2):15, 2008.
- [24] B. Zhang and D. Tse. Geometry of injection regions of power networks. *Power Systems, IEEE Transactions on*, 28(2):788–797, 2013.

Proof of Theorem 1. Notice that the variables in [OP₂] are x , P_k^g , and P_{lm} and Q_{lm} . However, not all the monomials appear in the polynomial formulation and hence using the first level of the hierarchy, \mathcal{K}_G^r , where $r = 2$ one can approximate [OP₂]:

$$\begin{aligned}
& \max \quad \varphi \\
& \text{s.t.} \quad \sum_{k \in G} \left(c_k^2 (P_k^g)^2 + c_k^1 (P_k^d + \text{tr}(Y_k x x^T)) + c_k^0 \right) - \varphi \\
& \quad = A(x) + \sum_{k \in G} B_k(P_k^g) + \sum_{(l,m) \in L} C_{lm}(P_{lm}, Q_{lm}) \\
& \quad + \sum_{k \in N} \bar{\lambda}_k (P_k^{\max} - P_k^d - \text{tr}(Y_k x x^T)) \\
& \quad + \sum_{k \in N} \underline{\lambda}_k (-P_k^{\min} + P_k^d + \text{tr}(Y_k x x^T)) \\
& \quad + \sum_{k \in N} \bar{\gamma}_k (Q_k^{\max} - Q_k^d - \text{tr}(\bar{Y}_k x x^T)) \\
& \quad + \sum_{k \in N} \underline{\gamma}_k (-Q_k^{\min} + Q_k^d + \text{tr}(\bar{Y}_k x x^T)) \\
& \quad + \sum_{k \in N} \bar{\mu}_k ((V_k^{\max})^2 - \text{tr}(M_k x x^T)) \\
& \quad + \sum_{k \in N} \underline{\mu}_k ((-V_k^{\min})^2 + \text{tr}(M_k x x^T)) \\
& \quad + \sum_{(l,m) \in L} a_{lm} ((S_{lm}^{\max})^2 - P_{lm}^2 - Q_{lm}^2) \\
& \quad + \sum_{k \in G} b_k (P_k^g - \text{tr}(Y_k x x^T) - P_k^d) \\
& \quad + \sum_{(l,m) \in L} c_{lm} (P_{lm} - \text{tr}(Y_{lm} x x^T)) \\
& \quad + \sum_{(l,m) \in L} d_{lm} (Q_{lm} - \text{tr}(\bar{Y}_{lm} x x^T))
\end{aligned}$$

where $A(x)$, $B_k(P_k^g)$, $C_{lm}(P_{lm}, Q_{lm})$ are polynomials that are sum of squares as a function of x , P_k^g , and P_{lm} and Q_{lm} respectively. That is $A(x) = xAx^T$, $B_k(P_k^g) = \begin{bmatrix} 1 \\ P_k^g \end{bmatrix} B_k \begin{bmatrix} 1 \\ P_k^g \end{bmatrix}^T$, and $C_{lm}(P_{lm}, Q_{lm}) = \begin{bmatrix} 1 \\ P_{lm} \\ Q_{lm} \end{bmatrix} C_{lm} \begin{bmatrix} 1 \\ P_{lm} \\ Q_{lm} \end{bmatrix}^T$, where A , B_k , and C_{lm} are positive semidefinite matrices of dimension $2|N| \times 2|N|$, 2×2 and 3×3 respectively. The variables $\bar{\lambda}_k, \underline{\lambda}_k, \bar{\gamma}_k, \underline{\gamma}_k, \bar{\mu}_k, \underline{\mu}_k$, and a_{lm} , are non-negative variables and b_k, c_{lm} , and d_{lm} are free variables. By equating the coefficients

of the monomials of the above problem, we rewrite it as

$$\begin{aligned}
& \max \quad \varphi \\
& \text{s.t.} \quad \sum_{k \in G} c_k^1 P_k^d + \sum_{k \in G} c_k^0 - \varphi \\
& \quad = \sum_{k \in G} B_k^{00} + \sum_{(l,m) \in L} C_{lm}^{00} + \sum_{k \in N} \bar{\lambda}_k (P_k^{\max} - P_k^d) \\
& \quad + \sum_{k \in N} \underline{\lambda}_k (-P_k^{\min} + P_k^d) + \sum_{k \in N} \bar{\gamma}_k (Q_k^{\max} - Q_k^d) \\
& \quad + \sum_{k \in N} \underline{\gamma}_k (-Q_k^{\min} + Q_k^d) + \sum_{k \in N} \bar{\mu}_k (V_k^{\max})^2 \\
& \quad - \sum_{k \in N} \underline{\mu}_k (V_k^{\min})^2 + \sum_{(l,m) \in L} a_{lm} (S_{lm}^{\max})^2 - \sum_{k \in G} b_k P_k^d \\
& \quad \sum_{k \in N} c_1^k Y_k = A - \sum_{k \in N} (\bar{\lambda}_k Y_k + \underline{\lambda}_k Y_k - \bar{\gamma}_k \bar{Y}_k \\
& \quad + \underline{\gamma}_k \bar{Y}_k - \bar{\mu}_k (V_k^{\max})^2 + \underline{\mu}_k (V_k^{\min})^2 - b_k Y_k) \\
& \quad - \sum_{(l,m) \in L} (c_{lm} Y_{lm} + d_{lm} \bar{Y}_{lm}) \\
& \quad 0 = 2B_k^{12} + b_k \quad c_k^2 = B_k^{22} \\
& \quad 0 = c_{lm} + 2C_{lm}^{12} \quad 0 = d_{lm} + 2C_{lm}^{13} \\
& \quad 0 = 2C_{lm}^{23} \quad 0 = -a_{lm} + C_{lm}^{22} \\
& \quad 0 = -a_{lm} + C_{lm}^{33} \\
& \quad A, B_k, C_{lm} \succeq 0.
\end{aligned}$$

By substituting some of the variables:

[OP₂-H₁]*

$$\begin{aligned}
& \max \quad \sum_{k \in G} c_k^1 P_k^d + \sum_{k \in G} c_k^0 - \sum_{k \in G} B_k^{00} - \sum_{(l,m) \in L} C_{lm}^{00} \\
& \quad - \sum_{k \in N} \bar{\lambda}_k (P_k^{\max} - P_k^d) - \sum_{k \in N} \underline{\lambda}_k (-P_k^{\min} + P_k^d) \\
& \quad - \sum_{k \in N} \bar{\gamma}_k (Q_k^{\max} - Q_k^d) - \sum_{k \in N} \underline{\gamma}_k (-Q_k^{\min} + Q_k^d) \\
& \quad - \sum_{k \in N} \bar{\mu}_k (V_k^{\max})^2 + \sum_{k \in N} \underline{\mu}_k (V_k^{\min})^2 \\
& \quad - \sum_{(l,m) \in L} C_{lm}^{22} (S_{lm}^{\max})^2 - \sum_{k \in G} 2B_k^{12} P_k^d \\
& \text{s.t.} \quad A = \sum_{k \in N} (c_1^k Y_k + \bar{\lambda}_k Y_k - \underline{\lambda}_k Y_k + \bar{\gamma}_k \bar{Y}_k \\
& \quad - \underline{\gamma}_k \bar{Y}_k + \bar{\mu}_k (V_k^{\max})^2 - \underline{\mu}_k (V_k^{\min})^2 - 2B_k^{12} Y_k) \\
& \quad - \sum_{(l,m) \in L} (2C_{lm}^{12} Y_{lm} + 2C_{lm}^{13} \bar{Y}_{lm}) \\
& \quad c_k^2 = B_k^{22} \quad 0 = 2C_{lm}^{23} \\
& \quad C_{lm}^{22} - C_{lm}^{33} = 0 \\
& \quad A, B_k, C_{lm} \succeq 0.
\end{aligned}$$

which is equivalent to optimization problem 4 described in [12], i.e., the dual of [OP-SDP]. \square