

A Global Solution Algorithm for AC Optimal Power Flow through Linear Constrained Quadratic Programming

Masoud Barati^a

^a*Electrical and Computer Engineering, University of Pittsburgh, Swanson School of Engineering, Pittsburgh, 15206, PA, USA*

Abstract

We formulate the Alternating Current Optimal Power Flow Problem (ACOPF) as a Linear Constrained Quadratic Program (LCQP) with a substantial number of negative eigenvalues (r) subjected to linear constraints, a problem acknowledged as NP-hard. We introduce two algorithms, Feasible Successive Linear Programming (FSLP) and Feasible Branch-and-Bound algorithm (FBB), aimed at a global optimal solution methodology. These algorithms employ a diverse range of simple but efficient optimization strategies such as bounded successive linear programming, convex relaxation, initialization, and branch-and-bound. Their objective is to discover a globally optimal solution to the inherent ACOPF, adhering to a predefined ϵ -tolerance level. The complexity bound of the FSLP and FBB algorithms is $\mathcal{O}\left(N \prod_{i=1}^r \left\lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \right\rceil\right)$, here, N signifies the complexity involved in resolving the subproblems at each node within the framework of the FBB algorithm. The variables t_l and t_u , elements of the set of real numbers raised to the power of r , represent the lower and upper boundaries of t respectively. Additionally, $-|t|^2$ is identified as the only negative quadratic component present in the objective function of the elevated ACOPF problem. Second, we employ a combination of penalized semidefinite modeling, convex relaxation, and line search techniques to design a globally feasible branch-and-bound algorithm for the LCQP form of ACOPF, capable of locating an optimal solution to ACOPF within ϵ -tolerance. Our initial numerical results illustrate that the FSLP and FBB algorithms can remarkably identify a global optimal solution for large-scale ACOPF instances, even when r is large. Furthermore, the suggested algorithm demonstrates enhanced performance over

other cutting-edge methods in the majority of PG-lib test scenarios.

Keywords: AC optimal power flow, brutal force algorithm, Branch and bound algorithm, global solution, lifted problem, linear constrained quadratic programming, nonconvex optimization, second order cone programming, successive linear program, semidefinite programming.

This work was supported by the National Science Foundation under Grant ECCS no. 1711921.

1. Introduction

The idea of AC Optimal Power Flow (ACOPF) problems was first introduced by Carpentier in the 1960s. The focus was to determine the optimal operational states for multiple generators within a transmission network, accommodating spatially and temporally varying demand [1, 2]. Rooted in the research surrounding power systems and their operation, OPF has matured into a well-founded discipline, utilized worldwide for day-to-day power grid management and regulation.

The ACOPF is an optimization problem widely encountered in different segments of power systems, strictly adhering to the AC power flow equations. This problem is usually structured to minimize the cost of generation while concurrently maintaining adherence to physical principles and engineering limitations. Conventional ACOPF problems are classified as Quadratically Constrained Quadratic Programs (QCQPs), which are complex mathematical models. According to the nonlinear characteristics of power flow equations, ACOPF problems display nonconvexity, rendering optimal power flow (OPF) problems intrinsically non-deterministic polynomial-time hard (NP-hard), as stated in the citation [3]. Given their nonlinear and nonconvex characteristics, ACOPF problems have been a focal point of extensive research across multiple disciplines. Existing solutions for ACOPF problems are generally divided into two primary categories: those relying on mathematical algorithms and those employing ACOPF-specific algorithms.

From a mathematical perspective, ACOPF problems are characterized as quadratic programs (QPs) with a handful of negative eigenvalues, subjected to quadratic constraints. The extensive application scope of non-convex QPs, coupled with their computational complexity, has drawn the attention of researchers from various fields, resulting in a plethora of algorithms presented in

the literature. Initial research efforts on non-convex QPs were predominantly concentrated on acquiring first-order or (weak) second-order Karush-Kuhn-Tucker (KKT) critical points with favorable objective values. For example, in reference [4], an uncomplicated alternating updating scheme was proposed, specifically designed for a QPs subclass, the bi-linear program. This method progressively converges to a KKT point. The authors expanded on this by detailing a local search process founded on linear optimization techniques to pinpoint a local optimum. In [5], a successive linear optimization approach was devised to ascertain a KKT point of QPs. In a similar vein, [6] offered an interior point method to determine a KKT point of the related QPs.

Over the past two decades, semidefinite optimization programs (SDPs) have rapidly evolved and have proven to be highly effective in addressing non-convex QPs by providing tight relaxations and approximate solutions. Reference [7] explores relaxations for nonconvex QCQPs rooted in semidefinite programming (SDP) and the reformulation-linearization technique (RLT). This study demonstrates the combined application of SDP and RLT constraints yields bounds significantly superior to those produced when either technique is used independently.

The concept of successive semidefinite relaxations (SDR) of a binary polytope and the confirmation of the process's finite convergence were explored in [8]. Reference [9] combined positive polynomial representations as sums of squares with moments theory to develop successive SDRs. This hierarchy was demonstrated to exhibit finite convergent behavior under certain conditions. The topic of successive SDRs pertaining to generic QPs was covered in [10, 11].

However, as highlighted in [12, 13], the number of variables and constraints dramatically increases with the rising levels of hierarchy in these procedures, which renders these methods computationally non-scalable. To remedy this, an alternative approach of formulating a branch-and-bound (B&B) for QPs based on SDRs was proposed in [14, 15]. The adoption of the entirely co-positive program, as documented in [16], served to refine the B&B methodology employed for LCQPs. Despite these B&B-based global solvers being capable of solving many small-scale LCQPs, they can surpass time/memory constraints and generate only a suboptimal solution for problem sizes ranging between 50 and 100, as observed in [16].

More recently, [17] proposed a global algorithm, ADMBB, applicable for quadratic programs subjected to linear and convex quadratic constraints and possessing a few negative eigenvalues. This algorithm can pinpoint a globally

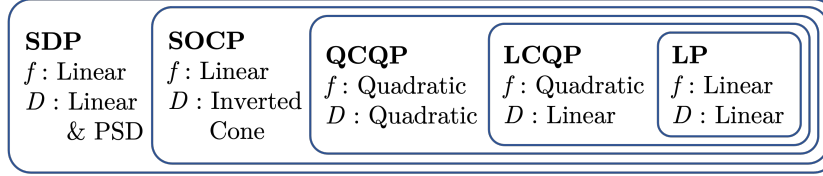


Figure 1: The atlas of ACOPF model in the literature including the specification of each model.

optimal solution for the fundamental QP within a pre-determined epsilon-tolerance. Unfortunately, this methodology is not computationally practical for large-scale applications, and the presented algorithm does not consistently provide bounded solution results.

Due to the inherent nonconvexity of ACOPF problems, contemporary ACOPF-based algorithms are increasingly focusing on convexified methodologies. The methods mentioned can be broadly categorized into two distinct groups: relaxations and approximations, as stated in [18]. The atlas of ACOPF models is depicted in Fig. 1. As outlined in (1), the objective function f can manifest as both linear and quadratic forms, while the feasible domain $\mathcal{D}$ can accommodate a range of constraint types, including linear, quadratic, inverted cone, and positive semidefinite linear constraints.

$$\min_{x \in \mathcal{D}} f(x) \quad (1)$$

Among the various algorithms, semi-definite programming (SDP) [19, 21] and second-order cone programming (SOCP) [24, 25] have been widely adopted and are well-established. However, it is crucial to mention that the conventional SOCP relaxation is not able to accommodate loop-based network constraints, specifically those where the total summation of voltage angle differences inside a loop equals zero. A contemporary investigation, referenced in [28], introduced a triad of enhancements to the second-order cone programming (SOCP) refinement applied to ACOPF. It is noteworthy, as discussed in [26, 28], that the solution space permitted by the semidefinite programming (SDP) refinement is encompassed by that of the SOCP refinement.

Leveraging these techniques, a number of improved algorithms, such as convex iteration, have been proposed, showcasing superior efficiency [20]. In [21], SDP was first used to address OPF problems, where OPF problems were reformulated into a SDP structure, and an interior point method (IPM) algo-

rithm within SDP was developed. [19] proposed a solution to the nonconvex ACOPF through the formulation of a convex semidefinite program based on the Lagrangian dual problem. The condition to ensure zero duality gap, derived from this work, was also satisfied for IEEE power test cases with up to 300 buses.

The research conducted in [20] leveraged sparsity methodologies to improve the SOCP relaxation of ACOPF. We accomplished this by ensuring that the sub-matrices, associated with the largest cliques and smallest chord-free cycles in the cycle foundation, maintained a positive semidefinite state. Moreover, virtual lines were added to minimal chordless cycles, thereby transforming each into 3-node cycles. Additionally, a convex iteration was incorporated into their proposed solver, resulting in a convex relaxation that was tightly gapped.

The [22] proposed a general solution framework for deriving near-optimal solutions for the SDP relaxation of QCQPs. They also introduced a rank penalty based on the nuclear norm of the Hermitian matrix to the SDP relaxation problem's objective, enforcing a low-rank solution. Reference [29] proposed a penalized perturbed SDP relation with the aim to augment the real part of certain entries of the Hermitian matrix, thereby guaranteeing a rank-1 solution under the optimality condition. The larger the value of the regularization coefficient, the more substantial the penalty imposed on the rank of the Hermitian matrix.

The papers [30, 31] further investigated the presence of a rank-1 SDP solution in relation to obtaining a global solution for SCOPF with line contingencies. The authors developed a graph-theoretic convex program aimed at pinpointing network lines that pose problems, and integrated the loss incurred over these lines into the objective function, thus serving as a regularization term. Should the relaxation not be exact, penalizing the SDP problem serves as an alternative and feasible strategy.

In reference [28], a novel method utilizing convex quadratically constrained quadratic programs has been introduced. This method incorporates a penalty strategy that is consistent with semidefinite programming, second-order cone programming, and parabolic optimization techniques, ensuring the derivation of viable solutions for optimal power flow problems, given certain conditions. This method mainly tackles two prevalent challenges: firstly, the penalization constants are typically predefined using heuristic approaches; and secondly, the addition of regularization terms to the objective function in the ACOPF problem can alter its original value, as these terms persist even at the optimal

solution.

Drawing inspiration from [23] and the penalized SDP relaxation algorithm, this paper introduces two innovative algorithms: Feasible Successive Linear Programming (FSLP) and Feasible Branch-and-Bound algorithm (FBB). The principal contributions of this paper are summarized as follows:

1. Elegant penalty terms, representing the second-order determinants of the symmetric semi-definite matrix, were integrated into the original objective function with the aim of achieving a rank of 1 for the matrix.
2. The penalized SDP relaxation formulated in this study is standard quadratic programming with linear constraints (LCQP), making it simpler to solve compared to other formats.
3. Taking into account the number of negative eigenvalues (r), two proposed algorithms were introduced: a bounded FSLP algorithm and a novel branch-and-bound (FBB) algorithm. These algorithms are designed to expedite the process of finding the optimal solution, surpassing the performance of current solution methodologies.
4. When global optimal solutions are obtained, the penalty terms in the objective function are equal to zero. This indicates that these terms do not influence the objective values when optimal rank-1 solutions are acquired. This is a distinctive feature compared to the aforementioned methods, which used regularizers as penalty terms without exhibiting the vanishing property in the rank-1 solution result.

The organization of this paper is outlined as follows: Section II presents the fundamental setup of the ACOPF formulation. Section III examines the penalized relaxation of SDP within the ACOPF problem. The different formats of ACOPF and the proposed bounded feasible successive linear programming algorithm are detailed in Section IV. Section V delves into the proposed novel branch-and-bound algorithm, accompanied by supporting theorems and the corresponding proofs. The computational findings are detailed in Section VI, while the conclusions are consolidated in Section VII.

2. AC Optimal Power Flow

We delineate the construction of the ACOPF problem being examined within the context of this research work.

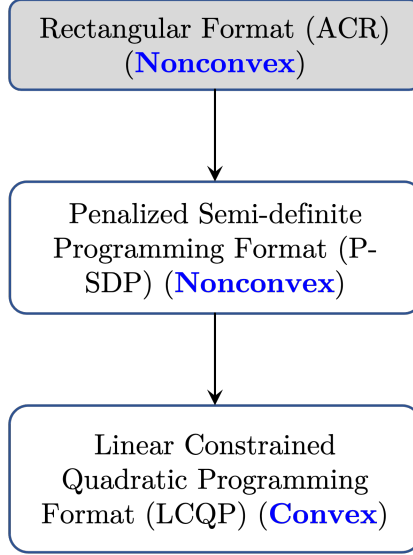


Figure 2: The trend of ACOPF model transformation in the presented solution methodology.

2.1. Mathematical ACOPF Formulation

In the present study, we consider one of the most prevalent configurations of ACOPF problems in rectangular format, primarily centered on minimizing the generation expense while simultaneously adhering to physical laws and engineering restrictions. This section revisits the fundamental principles of AC power grid model in the form of optimization programming and integrates these principles to shed light on the pivotal ACOPF problem. It also establishes the notation that is consistently used throughout the paper.

Electric power grids incorporate different elements, including generating units, demands, nodes or buses, and transmission lines. In a broad analysis, the collections $\mathcal{N}$ of nodes and $\mathcal{L}$ of edges constitute a network graph denoted by $(\mathcal{N}, \mathcal{L})$, where the nodes correlate with buses and the edges with the connecting lines. It is essential to recognize $\mathcal{L}$ as a set of bidirectional edges. However, for each connecting line $(i, j) \in \mathcal{L}$, there is a designated directionality, with a “source” (from) side (i, j) and a “destination” (to) side (j, i) .

thus enabling the tracking of line losses as power flows transition *from-end* to *to-end*. Usually, a slack bus $r \in \mathcal{N}$ is denoted for easy comparison of solutions and elimination of symmetric solutions. AC power flow equa-

tions define the power flow within the network. These equations outline the relationship between complex quantities of current I , voltage V , power S , and admittance y , which are based on the physical principles of Kirchhoff's current law, Ohm's law, and the computation of the AC power grid. The integration of the aforementioned three principles results in the well-known network power flow equations,

$$S_i = S_{G_i} - S_{D_i} = \sum_{(i,j) \in \mathcal{L}(i)} S_{ij} \quad \forall i \in \mathcal{N} \quad (2)$$

$$\begin{aligned} S_{ij} &= V_i I_{ij}^* = V_i y_{ij}^* (V_i^* - V_j^*) \\ &= y_{ij}^* V_i V_i^* - y_{ij}^* V_i V_j^* \quad \forall (i,j), \forall (j,i) \in \mathcal{L} \end{aligned} \quad (3)$$

Note that for bus $i \in \mathcal{N}$, the sum over $(i,j) \in \mathcal{L}$ integrates the power flow directed in the “from” orientation, while the sum over $(j,i) \in \mathcal{L}$ combines the flow directed in the “to” direction. Equations (2)-(3) constitute the basis for optimization of the power grid in operation and planning applications. These two equations are Nonconvex and nonlinear. However, each specific application enhances the functionality of these two equations with some additional supporting constraints. We employ a representative ACOPF problem formulation. This involves the conservation of active and reactive power flow at each node $i \in \mathcal{N}$, expressed as a function of the real and reactive power flow of each connecting line $(i,j) \in \mathcal{L}$ to every node $i \in \mathcal{N}$. The set of all lines connected to node i , is shown by $(i,j) \in \mathcal{L}(i)$. It also accounts for the maximum and minimum bounds of the real and reactive power flow of each connecting line (i,j) . Further considerations include upper and lower voltage magnitude limits at each node $i \in \mathcal{N}$, and the reference bus $r \in \mathcal{N}$ phase

angle is set as zero.

$$\begin{aligned}
& \min \sum_{i \in \mathcal{G}} f_i(P_{Gi}) \\
& s.t. \ P_{ij} = (g_{i0} + g_{ij})(V_{di}^2 + V_{qi}^2) - V_{di}(g_{ij}V_{dj} - b_{ij}V_{qj}) \\
& \quad + V_{qi}(b_{ij}V_{dj} + g_{ij}V_{qj}), \quad \forall (i, j) \in \mathcal{L} \\
& \quad Q_{ij} = -(b_{i0} + b_{ij})(V_{di}^2 + V_{qi}^2) + V_{di}(b_{ij}V_{dj} + g_{ij}V_{qj}) \\
& \quad - V_{qi}(g_{ij}V_{dj} - b_{ij}V_{qj}), \quad \forall (i, j) \in \mathcal{L} \\
& \quad P_{Gi} - P_{Di} = \sum_{(i,j) \in \mathcal{L}(i)} P_{ij}, \quad \forall i \in \mathcal{N} \\
& \quad Q_{Gi} - Q_{Di} = \sum_{(i,j) \in \mathcal{L}(i)} Q_{ij}, \quad \forall i \in \mathcal{N} \\
& \quad -P_{ij}^{max} \leq P_{ij} \leq P_{ij}^{max}, \quad \forall (i, j) \in \mathcal{L} \\
& \quad P_i^{min} \leq P_{Gi} \leq P_i^{max}, \quad \forall i \in \mathcal{N} \\
& \quad Q_i^{min} \leq Q_{Gi} \leq Q_i^{max}, \quad \forall i \in \mathcal{N} \\
& \quad (V_i^{min})^2 \leq V_{di}^2 + V_{qi}^2 \leq (V_i^{max})^2, \quad \forall i \in \mathcal{N} \\
& \quad V_{qk} = 0. \quad k \in \text{ref}
\end{aligned} \tag{4}$$

The $\mathcal{G}$ represents the generator buses in (4), while $\mathcal{N} = \{1, 2, \dots, |\mathcal{N}|\}$ represents all buses within the system. The real and reactive power generation units are denoted by P_{Gi} and Q_{Gi} . The load is considered as a constant load with real and reactive power consumption P_{Di} , and Q_{Di} . The real and imaginary parts of the complex voltages V_i are characterized as V_{di} and V_{qi} . Furthermore, g_{ij} and b_{ij} represent the real and imaginary parts of the series admittance element of the line $(i, j) \in \mathcal{L}$ as functions of the series resistance r_{ij} and reactance x_{ij} , where series conductance, $g_{ij} = \frac{r_{ij}}{r_{ij}^2 + x_{ij}^2}$ and series susceptance, $b_{ij} = -\frac{x_{ij}}{r_{ij}^2 + x_{ij}^2}$. The $g_{i0} = \frac{r_{i0}}{r_{i0}^2 + x_{i0}^2}$ and $b_{i0} = -\frac{x_{i0}}{r_{i0}^2 + x_{i0}^2}$ are the shunt elements connected to the bus $i \in \mathcal{N}$. The quadratic monomials of the voltage terms featured in P_{ij} and Q_{ij} equations within (4) render the ACOPF problem nonconvex and categorize them as NP-hard. The (4) implies the ACOPF as a Quadratic Constrained Quadratic Program (QCQP) format. The presence of quadratic voltage terms in the active and reactive power flow equations, as seen in (4), results in the ACOPF problems being nonconvex and thus classifies them as NP-hard. In response to the complexity and nonlinearity of this ACOPF problem, the penalized semidefinite program-

$$\mathbf{W} = \begin{array}{c|ccccccccc} & \dots & d_i & \dots & d_j & \dots & q_i & \dots & q_j & \dots \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_i & \dots & W_{ii} = V_{d_i}^2 & \dots & W_{ij} = V_{d_i} V_{d_j} & \dots & W_{i(i+|\mathcal{N}|)} = V_{d_i} V_{q_i} & \dots & W_{i(j+|\mathcal{N}|)} = V_{d_i} V_{q_j} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_j & \dots & W_{ji} = V_{d_j} V_{d_i} & \dots & W_{jj} = V_{d_j}^2 & \dots & W_{j(i+|\mathcal{N}|)} = V_{d_j} V_{q_i} & \dots & W_{j(j+|\mathcal{N}|)} = V_{d_j} V_{q_j} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_i & \dots & W_{(i+|\mathcal{N}|)i} = V_{q_i} V_{d_i} & \dots & W_{(i+|\mathcal{N}|)j} = V_{q_i} V_{d_j} & \dots & W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} = V_{q_i}^2 & \dots & W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)} = V_{q_i} V_{q_j} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_j & \dots & W_{(j+|\mathcal{N}|)i} = V_{q_j} V_{d_i} & \dots & W_{(j+|\mathcal{N}|)j} = V_{q_j} V_{d_j} & \dots & W_{(j+|\mathcal{N}|)(i+|\mathcal{N}|)} = V_{q_j} V_{q_i} & \dots & W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} = V_{q_j}^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

Figure 3: The shape of matrix $\mathbf{W}$ for two buses i and j includes all relevant elements in $\mathbf{W}$.

ming algorithm (SDP) outlined in [19] has been used, the details of which will be elaborated in the following section.

3. Semidefinite Programming (SDP) Relaxation

This section initially involves the relaxation of the general ACOPF problem (4) utilizing semidefinite programming (SDP). To start with, we first define the notation. The vector $\mathbf{V}$ is the voltage vector of an $|\mathcal{N}|$ -bus power system.

$$\mathbf{V} = [V_{d_1}, V_{d_2}, \dots, V_{d_{|\mathcal{N}|}} \mid V_{q_1}, V_{q_2}, \dots, V_{q_{|\mathcal{N}|}}]^\top \quad (5)$$

Consider the $2|\mathcal{N}| \times 2|\mathcal{N}|$ Hermitian matrix defined as $\mathbf{W}$. Suppose $\mathbf{W} = \mathbf{V}\mathbf{V}^\top$ is defined as a $2|\mathcal{N}| \times 2|\mathcal{N}|$ symmetric matrix. In this case, equation (4) can be restructured utilizing the matrix $\mathbf{W}$ for two representative buses, i and j , which correlate with respective constant voltage squares and bivariate voltage products. When considering a convex cost function, the semidefinite programming relaxation version can be derived from (4), which is attained by dismissing the nonconvex rank-1 constraint in (6). The constraints characterizing $\mathbf{W}$ include:

$$\begin{aligned} \mathbf{W} &\succeq 0 \\ \text{rank}(\mathbf{W}) &= 1 \end{aligned} \quad (6)$$

It should be emphasized that within the newly delineated $\mathbf{W}$ -space, the voltage constraints from (4) need to be presented in a different format.

$$\begin{aligned}
& \min \sum_{i \in \mathcal{G}} f_i(P_{G_i}) \\
& s.t. \ P_{ij} = (g_{i0} + g_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\
& \quad - g_{ij}W_{ij} + b_{ij}W_{i(j+|\mathcal{N}|)} - b_{ij}W_{(i+|\mathcal{N}|)j} \\
& \quad - g_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L} \\
& \quad Q_{ij} = -(b_{i0} + b_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\
& \quad + b_{ij}W_{ij} + g_{ij}W_{i(j+|\mathcal{N}|)} - g_{ij}W_{(i+|\mathcal{N}|)j} \\
& \quad + b_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L}
\end{aligned} \tag{7}$$

$$\begin{aligned}
P_{G_i} - P_{D_i} &= \sum_{(i,j) \in \mathcal{L}(i)} P_{ij}, \quad \forall i \in \mathcal{N} \\
Q_{G_i} - Q_{D_i} &= \sum_{(i,j) \in \mathcal{L}(i)} Q_{ij}, \quad \forall i \in \mathcal{N}
\end{aligned} \tag{8}$$

$$\begin{aligned}
& -P_{ij}^{max} \leq P_{ij} \leq P_{ij}^{max}, \quad \forall (i, j) \in \mathcal{L} \\
& P_i^{min} \leq P_{G_i} \leq P_i^{max}, \quad \forall i \in \mathcal{N} \\
& Q_i^{min} \leq Q_{G_i} \leq Q_i^{max}, \quad \forall i \in \mathcal{N} \\
& (V_i^{min})^2 \leq W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} \leq (V_i^{max})^2 \\
& \quad \forall i \in \mathcal{N}
\end{aligned} \tag{9}$$

$$\begin{aligned}
& W_{k(k+n)} = W_{(k+n)(k+n)} = 0, \quad \exists k \in \text{ref}, \\
& \text{rank}(\mathbf{W}) = 1, \\
& \mathbf{W} \succeq 0.
\end{aligned} \tag{10}$$

Presuming the generator cost functions are linear, i.e., $f_i(P_{G_i}) = c_i P_{G_i}$, the objective function in (7) becomes a linear function of the W 's. In the redefined W -space, it is observed that the objective function assumes a linear form and is subjected to linear constraints. This SDP relaxation format yields a lower-bound solution that deviates significantly from an exact optimal solution. For an approximation that closely aligns with a near-global optimum, it is advisable to integrate a convexified variant of the Rank-1 constraint into the optimization formulation as expressed in equation (7). In the forthcoming sections, we demonstrate that the Rank-1 constraint transforms into quadratic monomials in W -space within the objective function, which carries some negative eigenvalues— a characteristic indicative of a

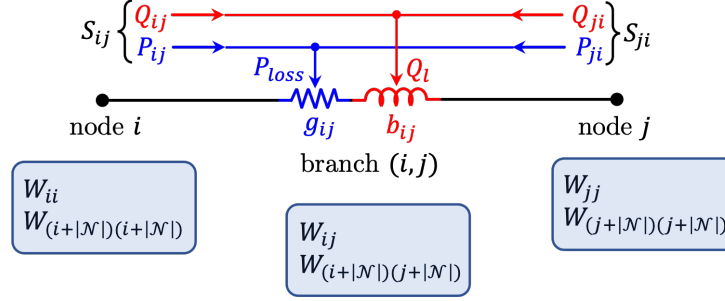


Figure 4: The corresponding elements of the Hermitian matrix related to nodes i and j of branch (i, j) in $\mathbf{W}$ -space; the interplay between active and reactive power flow at the sending and receiving ends, taking into account ohmic losses and line reactive power consumption.

typical quadratic programming scenario. Consequently, we aim to resolve a quadratic program that presents a limited number of negative eigenvalues subject to linear constraints (LCQP). Within this framework, the symbol tr denotes the trace operation on a matrix, defined as the aggregate of its diagonal entries. On the other hand, $\succeq$ indicates that the matrix $\mathbf{W}$ is positive semidefinite. W_{ii} represents the i -th diagonal term in the $\mathbf{W}$ matrix, while the other forms of $W_{(\cdot)(\cdot)}$ in the Hermitian matrix are illustrated in the extensive matrix $\mathbf{W}$.

Within this setting, solutions adhering to the Rank-1 constraint are recognized as the globally optimal solutions to the ACOPF challenges.

Theorem 3.1. *The Rank-1 solutions for the matrix the second-order determinants of W can reclaim W due to the following statement. The Rank-1 for the matrix W implies that all 2×2 minor determinants of W also possess a rank of 1.*

Proof. Let's consider $\omega = \begin{pmatrix} W_{ii} & W_{ij} \\ W_{ji} & W_{jj} \end{pmatrix}$, which is a symmetric 2×2 matrix.

Its leading principal minors are defined as $D_1 = W_{ii}$ and $D_2 = W_{ii}W_{jj} - W_{ij}^2$. To identify all the principal minors, we look at $\Delta_1 = W_{ii}$ and $\Delta_1 = W_{jj}$ (both of first order) as well as $\Delta_2 = W_{ii}W_{jj} - W_{ij}^2$ (of second order).

We aim to understand the implications of leading principal minors being positive for the 2×2 matrices. Let's denote $\omega = \begin{pmatrix} W_{ii} & W_{ij} \\ W_{ij} & W_{jj} \end{pmatrix}$ as a symmetric 2×2 matrix. We need to demonstrate that if $D_1 = W_{ii} > 0$ and $D_2 = W_{ii}W_{jj} - W_{ij}^2 > 0$, then ω is positively definite.

Given that $D_1 = W_{ii} > 0$ and $D_2 = W_{ii}W_{jj} - W_{ij}^2 > 0$, we can also conclude that $W_{jj} > 0$, since $W_{ii}W_{jj} > W_{ij}^2 \geq 0$. The characteristic equation of ω is

$$\lambda^2 - (W_{ii} + W_{jj})\lambda + (W_{ii}W_{jj} - W_{ij}^2) = 0$$

and it gives us two solutions (as A is symmetric), given by

$$\lambda_{1,2} = \frac{W_{ii} + W_{jj}}{2} \pm \frac{\sqrt{(W_{ii} + W_{jj})^2 - 4(W_{ii}W_{jj} - W_{ij}^2)}}{2}$$

As $(W_{ii} + W_{jj}) > \sqrt{(W_{ii} + W_{jj})^2 - 4(W_{ii}W_{jj} - W_{ij}^2)}$, both solutions are positive, indicating that ω is positive definite.

At this juncture, we are prepared to verify this claim in a comprehensive way. Let's demonstrate this statement through mathematical induction. The base case where $n = 1$ is quite straightforward. For the inductive step, let ω represent 2×2 principal submatrix of W and define $d_m(t) = \det(\omega + tI)$. According to Jacobi's formula, $d'_m(t) = \text{tr}(\text{adj}(\omega + tI))$. As the diagonal entries of $\text{adj}(\omega + tI)$ consist of some proper principal minors of $W + tI$, and considering that under the induction assumption, all proper principal submatrices of W are positive semidefinite, $d'_m(t)$ must be positive when $t > 0$. This implies that d_m strictly ascends to $[0, \infty)$. Therefore, for all $t > 0$, $d_m(t) > d_m(0) = \det(\omega) \geq 0$.

This leads us to conclude that all principal minors of $W + tI$ are positive when $t > 0$. Applying the original Sylvester's criterion for positive definite matrices, we find that $W + tI$ is positive definite. When taking $t \rightarrow 0^+$, it follows that W is positive semidefinite.

Alternatively, the assertion can be easily proven without the need for mathematical induction. If we take ω to be any 2×2 principal submatrix of W (this includes instances where $n = 2$ and $\omega = W$) and $t > 0$, then $\det(\omega + tI) = t^2 + \sum_{k=1}^2 s_2(\omega)t^{2-k}$, where $s_2(\omega)$ represents the total summation of all the 2×2 principal minors of ω . Each principal minor of ω also qualifies as a principal minor of W , $s_2(\omega)$ is non-negative. Thus, $\det(\omega + tI) \geq t^2 > 0$, implying that all principal minors of $W + tI$ are positive. Consequently, according to Sylvester's criterion for positive definite matrices, $W + tI$ is positive definite, which in turn makes $W = \lim_{t \rightarrow 0^+} (W + tI)$ positive semidefinite. $\square$

Theorem 3.2. *Furthermore, ohmic losses in the transmission line ensure that the cumulative sum of the two relevant minor determinants associated with the line $(i, j) \in \mathcal{L}$ is non-negative.*

Proof. Ohmic losses in the transmission line connecting nodes i and j are quantified by the subsequent expression:

$$\begin{aligned} P_{ij} = & (g_{i0} + g_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\ & - g_{ij}W_{ij} + b_{ij}W_{i(j+|\mathcal{N}|)} - b_{ij}W_{(i+|\mathcal{N}|)j} \\ & - g_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L} \end{aligned} \quad (11)$$

$$\begin{aligned} P_{ji} = & (g_{j0} + g_{ji})(W_{jj} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)}) \\ & - g_{ji}W_{ji} + b_{ji}W_{j(i+|\mathcal{N}|)} - b_{ji}W_{(j+|\mathcal{N}|)i} \\ & - g_{ji}W_{(j+|\mathcal{N}|)(i+|\mathcal{N}|)}, \quad \forall (j, i) \in \mathcal{L} \end{aligned} \quad (12)$$

$$\begin{aligned} P_{ij} + P_{ji} = & g_{i0}W_{ii} + g_{j0}W_{jj} + g_{ij}(W_{ii} + W_{jj} - 2W_{ij}) \\ & + g_{i0}W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} + g_{j0}W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} \\ & + g_{ij}(W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} \\ & - 2W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}), \quad \forall (i, j) \in \mathcal{L} \end{aligned} \quad (13)$$

$$\begin{aligned} Q_{ij} = & -(b_{i0} + b_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\ & + b_{ij}W_{ij} + g_{ij}W_{i(j+|\mathcal{N}|)} - g_{ij}W_{(i+|\mathcal{N}|)j} \\ & + b_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L} \end{aligned} \quad (14)$$

$$\begin{aligned} Q_{ji} = & -(b_{j0} + b_{ji})(W_{jj} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)}) \\ & + b_{ji}W_{ji} + g_{ji}W_{j(i+|\mathcal{N}|)} - g_{ji}W_{(j+|\mathcal{N}|)i} \\ & + b_{ji}W_{(j+|\mathcal{N}|)(i+|\mathcal{N}|)}, \quad \forall (j, i) \in \mathcal{L} \end{aligned} \quad (15)$$

$$\begin{aligned} Q_{ij} + Q_{ji} = & -b_{i0}W_{ii} - b_{j0}W_{jj} - b_{ij}(W_{ii} + W_{jj} - 2W_{ij}) \\ & - b_{i0}W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} - b_{j0}W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} \\ & - b_{ij}(W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} \\ & - 2W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}), \quad \forall (i, j) \in \mathcal{L} \end{aligned} \quad (16)$$

The terms W_{ii} and W_{jj} maintain non-negative values. Additionally, according to Theorem 3.1, we have $W_{ii} + W_{jj} - 2W_{ij} \geq 0$ and $W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} - 2W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)} \geq 0$. Consequently, we can infer that,

$$P_{ij} + P_{ji} \geq 0, \quad (17)$$

Furthermore, since b_{i0} , b_{j0} , and b_{ij} all have positive values,

$$Q_{ij} + Q_{ji} \geq 0. \quad (18)$$

□

Therefore, incorporating two fresh inequalities (17) and (18) into the SDP relaxation method can steer the solution towards achieving a precise result.

Finally, leveraging Theorem 3.1 and 3.2, we introduce a penalized relaxation approach of SDP subject to linear constraints as a function of W s. In this methodology, the minor determinants are incorporated as new penalty terms in the objective function of (7), along with the consideration of constraints (17) and (18). This strategy ensures the tightest relaxation for SDP. Remarkably, this novel formulation can significantly steer the rank of the W matrix towards 1.

3.1. Penalized SDP Relaxation

The SDP relaxation with an imposed penalty for the equation denoted as (7) is expressed as follows,

$$\begin{aligned} \min \quad & \sum_{k \in \mathcal{G}} f_k(P_{Gk}) + \sum_{(i,j) \in \mathcal{L}} \lambda_{ij}(W_{ii}W_{jj} - W_{ij}^2) \\ & + \sum_{(i,j) \in \mathcal{L}} \lambda_{ij}(W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} + W_{(j+|\mathcal{N}|)(j+|\mathcal{N}|)} \\ & \quad - 2W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}) \end{aligned} \quad (19)$$

$$\begin{aligned} s.t. \quad & P_{ij} = (g_{i0} + g_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\ & \quad - g_{ij}W_{ij} + b_{ij}W_{i(j+|\mathcal{N}|)} - b_{ij}W_{(i+|\mathcal{N}|)j} \\ & \quad - g_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L} \\ & Q_{ij} = -(b_{i0} + b_{ij})(W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)}) \\ & \quad + b_{ij}W_{ij} + g_{ij}W_{i(j+|\mathcal{N}|)} - g_{ij}W_{(i+|\mathcal{N}|)j} \\ & \quad + b_{ij}W_{(i+|\mathcal{N}|)(j+|\mathcal{N}|)}, \quad \forall (i, j) \in \mathcal{L} \\ & P_{ij} + P_{ji} \geq 0, \quad \forall (i, j), (j, i) \in \mathcal{L} \\ & Q_{ij} + Q_{ji} \geq 0, \quad \forall (i, j), (j, i) \in \mathcal{L} \\ & -P_{ij}^{max} \leq P_{ij} \leq P_{ij}^{max}, \quad \forall (i, j) \in \mathcal{L} \end{aligned} \quad (20)$$

$$\begin{aligned}
P_{G_i} &= P_{D_i} + \sum_{(i,j) \in \mathcal{L}(i)} P_{ij}, & \forall i \in \mathcal{N} \\
Q_{G_i} &= Q_{D_i} + \sum_{(i,j) \in \mathcal{L}(i)} Q_{ij}, & \forall i \in \mathcal{N}
\end{aligned} \tag{21}$$

$$\begin{aligned}
P_i^{min} &\leq P_{G_i} \leq P_i^{max}, & \forall i \in \mathcal{N} \\
Q_i^{min} &\leq Q_{G_i} \leq Q_i^{max}, & \forall i \in \mathcal{N} \\
(V_i^{min})^2 &\leq W_{ii} + W_{(i+|\mathcal{N}|)(i+|\mathcal{N}|)} \leq (V_i^{max})^2 \\
&& \forall i \in \mathcal{N}
\end{aligned} \tag{22}$$

$$W_{k(k+n)} = W_{(k+n)(k+n)} = 0. \quad \exists k \in \text{ref}$$

In equation (19), the hyperparameter $\lambda_{ij} = \rho((g_{i0} + g_{ij})^2 + (b_{i0} + b_{ij})^2)$ signifies the relative importance of each transmission line in the objective function. Here, ρ is a large value (for instance, 1000). The problem defined by equations (19) - (22) presents a nonconvex quadratic function with negative eigenvalues under linear constraints, constituting a canonical Linear Constrained Quadratic Programming problem (LCQP). The following section will detail the methodology to solve this LCQP problem using Feasible Successive Linear Programming (FSLP) and a new branch and bound (BB) algorithm.

3.2. ACOPF in the form of LCQP

Firstly, we reformulate problem (19) - (22) into a standard quadratic form. This simplification will facilitate a more straightforward expression.

$$\begin{aligned}
\min \quad & f(W) = W^\top A W + b^\top W \\
s.t. \quad & W \in \mathcal{F}, \mathcal{F} = \{W \in \mathbb{R}^n : LW \leq l\}
\end{aligned} \tag{23}$$

In this case, W denotes a vector in $\mathbb{R}^n$, A signifies an $n \times n$ matrix in $\mathbb{R}$, L represents an $m \times n$ matrix in $\mathbb{R}$, b stands for a vector in $\mathbb{R}^n$, and l is a vector in $\mathbb{R}^m$. Crucially, the W vector in equation (23) is a reconfigured version of the W matrix found in equations (19) - (22), taking advantage of the symmetric attribute of the W matrix. Figure 5 illustrates the new dimension of this vector. The matrix W has active elements shown in blue that correspond to the components of the Hermitian matrix, equating to $(2|\mathcal{N}| - 1) + \frac{(2|\mathcal{N}|-1)(2|\mathcal{N}|-1)-(2|\mathcal{N}|-1)}{2}$. Therefore, the size of the W vector is $n \times 1$, where $n = (2\mathcal{N} - 1)\mathcal{N}$. Considering the presence of negative eigenvalues in matrix A , the quadratic term in the objective function can

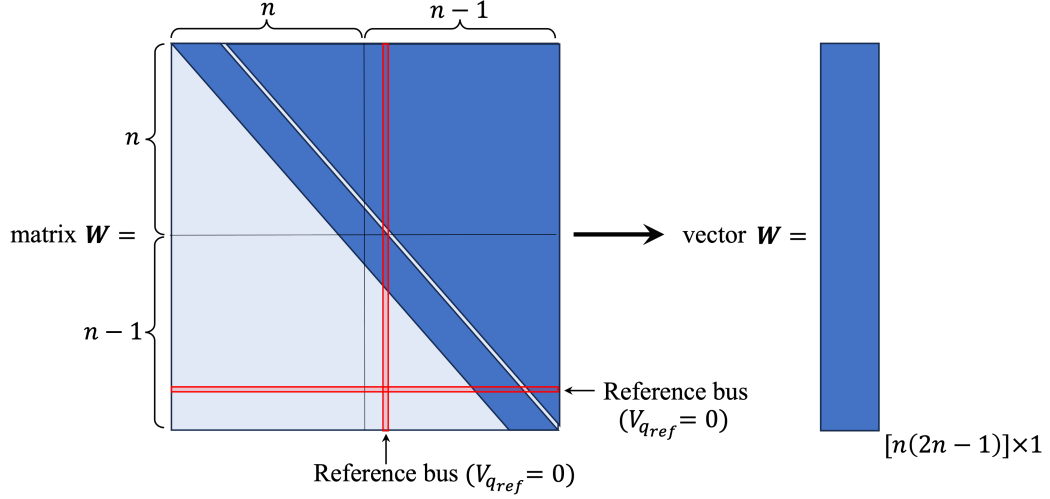


Figure 5: the shape of $\mathbf{W}$ matrix.

be reformulated through the application of the Difference Convex (D.C.) program, as demonstrated in [32],

$$\begin{aligned} W^\top A W &= W^\top A_+ W - W^\top A_- W \\ &= W^\top A_+ W - \|C W\|^2, \end{aligned} \quad (24)$$

Matrices that are symmetric and exhibit non-negative definiteness A_+ and A_- originate from the singular value decomposition (SVD) of A . Moreover, C is an $r \times n$ matrix, with r representing the count of negative eigenvalues in A . Details regarding the matrix C can be found in the *FBB Algorithm* section. Hence, the problem (23) can be reframed as the following elevated optimization problem,

$$\min_{(W,t) \in \mathcal{F}} b^\top W + W^\top A_+ W - \|t\|^2, \quad (25)$$

$$\mathcal{F} = \{(W, t) \mid L W \leq l, \ t_l \leq t \leq t_u, \ C W = t\}.$$

The notation $t_l, t_u \in \mathbb{R}^r$ signifies the respective lower and upper limits, which are obtained by solving the following linear optimization problems,

$$t_l^i = \min_{W \in \mathcal{F}} c_i^\top W, \quad t_u^i = \max_{W \in \mathcal{F}} c_i^\top W, \quad i = \{1, \dots, r\}, \quad (26)$$

Here, the term $c_i^\top$ denotes the transpose of the i -th row of the matrix C . Similar to the approach used in [33, 34], a linear surrogate objective function

can be utilized to represent the negative quadratic component over a specific set t bounded by the range $[t_l, t_u]$. Consequently, we derive the ensuing linear representation:

$$\begin{aligned} \min \quad & W^\top A_+ W + b^\top W - t^\top C W, \\ \text{s.t.} \quad & W \in \mathcal{F}. \end{aligned} \quad (27)$$

Expanding on the resolution of the issue delineated in (27), we have developed an algorithm for successive linear programming that is viable, denoted as FSLP, to address the problem referenced in (25). This algorithm employs an iterative approach, sequentially updating W and t at each iteration to find the optimal solution.

4. Bi-convex format of the ACOPF and Feasible Successive Linear Programming Algorithm

Subsequently, we examine a distinct Lagrangian approach for the problem denoted by (25), described in the following manner,

$$\mathcal{L}(W, t, \lambda) = W^\top A_+ W + \lambda^\top (C W - t) + b^\top W - \|t\|^2, \quad (28)$$

In this scenario, the Lagrangian multipliers are utilized for the constraint $C W = t$. A thorough examination of the pertinent Karush–Kuhn–Tucker (KKT) conditions related to the previously mentioned Lagrangian function reveals t . It can be deduced that $\frac{\partial \mathcal{L}(W, t, \lambda)}{\partial t} = -2t - \lambda = 0$, thereby leading to the inference that $\lambda = -2t$. It implies that λ is not an additional variable.

It is important to note that the solution achieving optimality for the problem delineated in (23) is required to fulfill the criteria set by the KKT conditions associated with the respective Lagrangian function, as pointed out in [35]. Consequently, the Lagrangian function can be reformulated in the following manner:

$$\mathcal{L}^*(W, t) = W^\top A_+ W + b^\top W - 2t^\top C W + \|t\|^2 \quad (29)$$

For convenience, we call the above Lagrangian function as the optimal Lagrangian function. Because

$$\begin{aligned} & W^\top A_+ W + b^\top W - 2t^\top C W + \|t\|^2 \\ &= W^\top A_+ W + b^\top W - \|C W\|^2 + \|t - C W\|^2, \end{aligned}$$

therefore, the problem (23) can be reformulated as a bi-convex optimization problem.

$$\begin{aligned} \min \quad & W^\top A_+ W + b^\top W - 2t^\top CW + \|t\|^2 \\ \text{s.t.} \quad & W \in \mathcal{F} \end{aligned} \quad (30)$$

Although the method draws on traditional successive linear programming techniques, there is no guarantee in this particular problem that a bounded solution will be found. This means that there is a potential for t_l^i to become $-\infty$ and t_u^i to become $+\infty$. As such, our proposed FSLP offers an enhanced bounded successive linear programming strategy that avoids unbounded solutions. Using an elementary case of Young's inequality [36], the inequality with exponent 2 for two variables t and CW can lead to (31).

$$t^\top CW \leq \frac{\|t\|^2}{2} + \frac{\|CW\|^2}{2} \quad (31)$$

The inequality (31) provides an upper bound limit on the bi-convex term and the following inequality provides a lower bound limit on the bi-convex term in the objective function. This inequality is based on the first-order Taylor series on bi-variant term $t^\top(CW)$.

$$t^\top(CW) \geq t^{k\top} CW^k + (t - t^k)^\top CW^k + t^{k\top} C(W - W^k) \quad (32)$$

Therefore, an enhanced bi-convex optimization problem can be written in (33).

$$\begin{aligned} \min \quad & W^\top A_+ W + b^\top W - 2t^\top CW + \|t\|^2 \\ \text{s.t.} \quad & t^\top CW \leq \frac{\|t\|^2}{2} + \frac{\|CW\|^2}{2} \\ & t^\top CW \geq t^{k\top} CW^k + (t - t^k)^\top CW^k \\ & \quad + t^{k\top} C(W - W^k) \\ & W \in \mathcal{F}. \end{aligned} \quad (33)$$

Notably, the strategy of transforming non-convex QP issues into bi-convex optimization is commonly found in literature [37, 38, 39]. The unique aspect of the model (33) is the bounded condition imposed on the variable t . The bi-convex optimization proposed in [37, 38, 39] has no restrictions on t , leading

to unbounded solution results. We can observe that the solution to the problem (33) can enhance the value of the objective function more efficiently compared to its unrestricted counterpart. It is crucial to note that for a given t , the problems (33) and (27) are equivalent.

Given a choice of t within the interval $[t_l, t_u]$, it becomes clear that the problem outlined in equation (33) is feasible, bounded and properly formulated. Then, we can infer several important theorems.

Theorem 4.1. *If $\bar{W}$ is an optimal solution to the problem denoted by (33), under the condition that $t = C\bar{W}$, then $\bar{W}$ qualifies as a KKT point for (33).*

Let $\mathcal{S}(t)$ denote the set of optimal solutions for (33), and assume that $W(t)$ is an element of $\mathcal{S}(t)$. The following lemma outlines several essential characteristics of the mapping $CW(t)$ in relation to t .

Lemma 4.2. *Given that $W(t) \in \mathcal{S}(t)$, we observe the following:*

1. *For any $W \in \mathcal{F}$ satisfying the inequality*

$$|CW - t| - |CW(t) - t| \leq 0,$$

it follows that

$$f(W) - f(W(t)) \geq 0,$$

2. *If W^* represents the global optimal solution of (33), then*

$$|CW(t) - t| - |CW^* - t| \leq 0, \tag{34}$$

3. *The function $CW(t)$ is monotonic with respect to t , as evidenced by*

$$(t_1 - t_2)^\top [CW(t_1) - CW(t_2)] \geq 0, \tag{35}$$

where

$$t_1 \neq t_2 \in [t_l, t_u], \quad W(t_1) \in \mathcal{S}(t_1), \quad W(t_2) \in \mathcal{S}(t_2).$$

Proof. The initial assertion is a direct consequence of the recognition that $W(t)$ constitutes the optimal solution for problem (30), coupled with the realization that $|CW - t| \leq |CW(t) - t|$. Now, let us turn our attention to the second proposition. Given that W^* represents the global optimum of the problem and considering $W(t) \in \mathcal{F}$, we can confidently state that

$$f(W(t)) - f(W^*) \geq 0. \tag{36}$$

Given that W^* is a feasible solution for problem (30), and given $W(t) \in \mathcal{S}(t)$, it follows that

$$f(W(t)) + \|CW(t) - t\|^2 - f(W^*) - \|CW^* - t\|^2 \leq 0. \quad (37)$$

From equations (36) and (37), it is evident that

$$\|CW(t) - t\| - \|CW^* - t\| \leq 0. \quad (38)$$

This verifies second statement of the theorem.

Next, for third statement, assume $t_1 \neq t_2 \in [t_l, t_u]$, $W(t_1) \in \mathcal{S}(t_1)$, and $W(t_2) \in \mathcal{S}(t_2)$. Using the optimal conditions of problem (30), we can deduce that

$$\begin{aligned} f(W(t_1)) + \|CW(t_1) - t_1\|^2 &\leq f(W(t_2)) + \|CW(t_2) - t_1\|^2, \\ f(W(t_2)) + \|CW(t_2) - t_2\|^2 &\leq f(W(t_1)) + \|CW(t_1) - t_2\|^2. \end{aligned}$$

Adding the above two inequalities, we derive

$$\begin{aligned} t_2^\top CW(t_2) + t_1^\top CW(t_1) &\geq t_2^\top CW(t_1) + t_1^\top CW(t_2) \\ \iff (t_1 - t_2)^\top C(W(t_1) - W(t_2)) &\geq 0, \end{aligned}$$

which yields relation (35). This concludes the proof.

Moreover, the lower bound constraint (32) provides a comprehensive way to prove the statement. The linearized bi-variant function at t_1 can be expressed as follows.

$$\begin{aligned} t^\top CW(t) \big|_{t=t_1} &\geq t_1^\top CW(t_1) + (t - t_1)^\top CW(t_1) \\ &\quad + t_1^\top C(W(t) - W(t_1)) \end{aligned}$$

Hence, at $t = t_2$, we have,

$$\begin{aligned} t_2^\top CW(t_2) &\geq t_1^\top CW(t_1) + (t_2 - t_1)^\top CW(t_1) \\ &\quad + t_1^\top C(W(t_2) - W(t_1)) \end{aligned}$$

It implies that,

$$\begin{aligned} (t_2 - t_1)^\top CW(t_2) &\geq (t_2 - t_1)^\top CW(t_1) \\ (t_1 - t_2)^\top CW(t_2) &\leq (t_1 - t_2)^\top CW(t_1) \end{aligned}$$

Finally,

$$(t_1 - t_2)^\top C(W(t_1) - W(t_2)) \geq 0$$

□

The method of Feasible Successive Linear Programming (FSLP) utilized to tackle LCQP problems is encapsulated in the algorithm delineated below.

Algorithm 1: Feasible Successive Linear Programming
FSLP(W^0, ϵ)

Input: Commencing with W^0 as the starting value and employing $\epsilon > 0$ as the threshold for termination

Output: The ultimate solution (W^{k+1}, t^{k+1}) is returned when the algorithm halts

Step 0 Initialization: Set the iteration counter k to 0. Calculate the initial value of t^k using the formula $t^k = CW^k$, where C is a given matrix, and W^k is the current estimate of the parameter values;

Step 1 Iterative Optimization: Solve the given optimization problem (denoted by equation (33)) with t set to the current t^k to obtain the optimal solution $W(t^k)$. Update the parameter estimates by setting $W^{k+1} = W(t^k)$. Calculate the new value of t for the next iteration using the formula $t^{k+1} = CW^{k+1}$;

Step 2 Convergence Check: Compute the Euclidean distance between t^{k+1} and t^k . If this distance exceeds the square root of the stopping criterion $\sqrt{\epsilon}$, increase the number of iterations k by 1 and return to the step of “Iterative Optimization”. If this condition does not hold, assume that the solution has converged;

We need supplementary supporting properties to ensure that the solution of (33) behaves appropriately throughout the FSLP iterative sequence. The following Lemma offers proof of the solution’s monotonic behavior at each step in the iteration.

Lemma 4.3. *Consider the sequences W^k and t^k , produced through the FSLP algorithm, with W^* representing the global optimal solution of (23). The following propositions are valid:*

(23). *The following assertions hold true:*

1. *The sequence t^k exhibits a monotonic trend, as can be seen from*

$$(t^k - t^{k-1})^\top (t^{k+1} - t^k) \geq 0;$$

2. *For every k , the inequality $f(W^{k-1}) - f(W^k) - \|t^{k-1} - t^k\|^2 \geq 0$ is valid;*
3. *For each k , the inequality $f(W^*) \leq f(W^k) \leq f(W^*) + \|t^* - t^{k-1}\|^2$ holds true.*

Proof. The first statement follows the third statement of Lemma 4.2 and the FSLP algorithm. The lower bound constraint (32) provides a comprehensive way to prove this statement. The linearized bi-variant function at t^{k-1} can be expressed as follows.

$$\begin{aligned} t^\top CW(t) \big|_{t=t^{k-1}} &\geq t^{k-1\top} CW(t^{k-1}) + (t - t^{k-1})^\top CW(t^{k-1}) \\ &\quad + t^{k-1\top} C(W(t) - W(t^{k-1})) \end{aligned}$$

Hence, at $t = t^k$, we have,

$$\begin{aligned} t^\top CW(t^k) &\geq t^{k-1\top} CW(t^{k-1}) + (t^k - t^{k-1})^\top CW(t^{k-1}) \\ &\quad + t^{k-1\top} C(W(t^k) - W(t^{k-1})) \end{aligned}$$

It implies that,

$$\begin{aligned} (t^k - t^{k-1})^\top CW(t^k) &\geq (t^k - t^{k-1})^\top CW(t^{k-1}) \\ (t^k - t^{k-1})^\top (CW(t^k) - CW(t^{k-1})) &\geq 0 \end{aligned}$$

Finally,

$$(t^k - t^{k-1})^\top (t^{k+1} - t^k) \geq 0$$

Now, let's examine the second statement. Since W^k is the optimal solution of (30) for $t = t^{k-1}$ and $W^{k-1} \in \mathcal{F}$, it follows that:

$$f(W^k) + |CW^k - t^{k-1}|^2 - f(W^{k-1}) - |CW^{k-1} - t^{k-1}|^2 \leq 0,$$

Factoring in that $CW^{k-1} = t^{k-1}$, we derive:

$$f(W^{k-1}) - f(W^k) - |t^k - t^{k-1}|^2 \geq 0, \quad \forall k \geq 1.$$

Pertaining to the third statement, given that W^* represents the global optimum for problem (23) and W^k is an element of $\mathcal{F}$, we deduce the subsequent:

We can infer the following from (39):

$$f(W^k) - f(W^*) \geq 0, \quad \forall k. \quad (39)$$

Also, given that W^k is the optimal solution of problem (33) when $t = t^k$ and $W^* \in \mathcal{F}$, we can deduce:

$$f(W^k) + |CW^k - t^{k-1}|^2 - f(W^*) - \|CW^* - t^{k-1}\|^2 \leq 0. \quad (40)$$

It then follows from the above two equations (39) and (40) that,

$$f(W^*) \leq f(W^k) \leq f(W^*) + \|CW^* - t^{k-1}\|^2.$$

This concludes the proof. $\square$

Subsequently, we will describe the characteristics of the accumulation point of a sequence that is produced by the FSLP algorithm. To achieve this, we present the concept of an ϵ -local minimizer as defined below.

Definition 4.1. We denote W^* as an ϵ -local minimizer of problem (23) within a given neighborhood, given that W^* belongs to $\mathcal{F}$,

$$\mathcal{N}(W^*, \delta) = W : \|W - W^*\| \leq \sqrt{\epsilon},$$

should the following condition hold for every $W \in \mathcal{N}(W^*, \epsilon) \cap \mathcal{F}$:

$$f(W^*) - f(W) \leq \mathcal{O}(\epsilon) \quad (41)$$

is satisfied, where $\lim_{\epsilon \rightarrow 0} \frac{\mathcal{O}(\epsilon)}{\epsilon} = \delta$ for certain value of $\delta > 0$.

Let's consider the following example that illustrates the concept of ϵ -approximation in the vicinity of a global optimal solution.

$$\begin{aligned} \min_{W_{22}, W_{11}} \quad & 100W_{11}^2 + 130W_{22}^2 - 210W_{11}W_{22} - 60W_{11} + 30 \\ \text{s.t.} \quad & W_{22} + 2W_{11} \leq b_i, \quad \forall i \in \{1, 2, \dots, 31\} \\ & -0.5 \leq W_{ii} \leq 9.5, \quad \forall i \in \{1, 2\} \\ & W \in \mathcal{F}. \end{aligned} \quad (42)$$

Where, b_i is a linearspace on $[-1.5, 1.5]$, such as $b_i = -1.5 + \frac{i \times (1.5 - (-1.5))}{30}$, for $\forall i = \{1, 2, \dots, 30\}$. Fig. 6 compares the situation of two global and local solutions. In LCQP optimization problems, finding the exact optimal solution is often challenging, especially when the problem has many negative eigenvalues. Therefore, in practice, we often aim to find an approximate solution that is "good enough" – a solution within a certain small distance, often denoted ϵ , from the true global optimal solution. An ϵ -approximation of global and local solutions refers to approximate solutions that lie within a distance ϵ from the true global and local optimal solutions.

Fig. 6 describes the points that represent solutions and circles with a certain radius to denote the ϵ neighborhoods (the areas within a distance ϵ

from the points) around these solutions. The green point, which lies outside the ϵ -radius, represents an approximation that does not meet the required level of precision. This point is situated in an expanded region with a radius of ϵ' (where $\epsilon' > \epsilon$), indicating that this approximation is further away from the optimal solution than we would ideally want.

However, the red point within the ϵ -radius represents a more precise approximation. Since it is positioned within the area bounded by the ϵ -radius around the true optimal solution, this indicates that the red point is close enough to the optimal solution to be considered a satisfactory ϵ -approximation. It is important to note that the smaller the value of ϵ , the closer the approximation is to the optimal solution. In other words, a smaller ϵ corresponds to a higher level of precision. Therefore, in this case, the red point represents a better approximation of the optimal solution than the green point.

The aforementioned definition can be regarded as an extensive interpretation of the term ϵ -solution, as delineated in [40]. As a result, we proceed to formulate the ensuing theorem.

Theorem 4.4. *Consider the case where $W(t)$ represents the optimal solution to equation (30) for a specific $t \in [t_l, t_u]$. When the condition $t = CW(t)$ is met, it follows that $W(t)$ will act as an ϵ -local solution to the function $f(W)$ within a suitable neighborhood.*

Proof. Given that $W(t)$ is the optimal solution to problem (30) and assuming the condition $t = CW(t)$ is true, we can deduce:

$$f(W) + \|CW(t) - CW\|^2 \geq f(W(t)), \quad \forall W \in \mathcal{F}. \quad (43)$$

By leveraging Definition 4.1, we can derive the following conclusion when a specific ϵ is set:

$$\|CW - CW(t)\|^2 \leq \|C\|^2 \|W - W(t)\|^2 \leq \epsilon \|C\|^2,$$

When combined with (43), it results in:

$$f(W) \geq f(W(t)) - \epsilon.$$

where $\lim_{\epsilon \rightarrow 0} \frac{\mathcal{O}(\epsilon)}{\epsilon} = \|C\|^2$. This substantiates that $W(t)$ functions as a ϵ -local solution to the problem (19)-(22), with the neighborhood $\mathcal{N}(W(t), \epsilon)$ taken into account. This brings the theorem to the end of the proof. $\square$

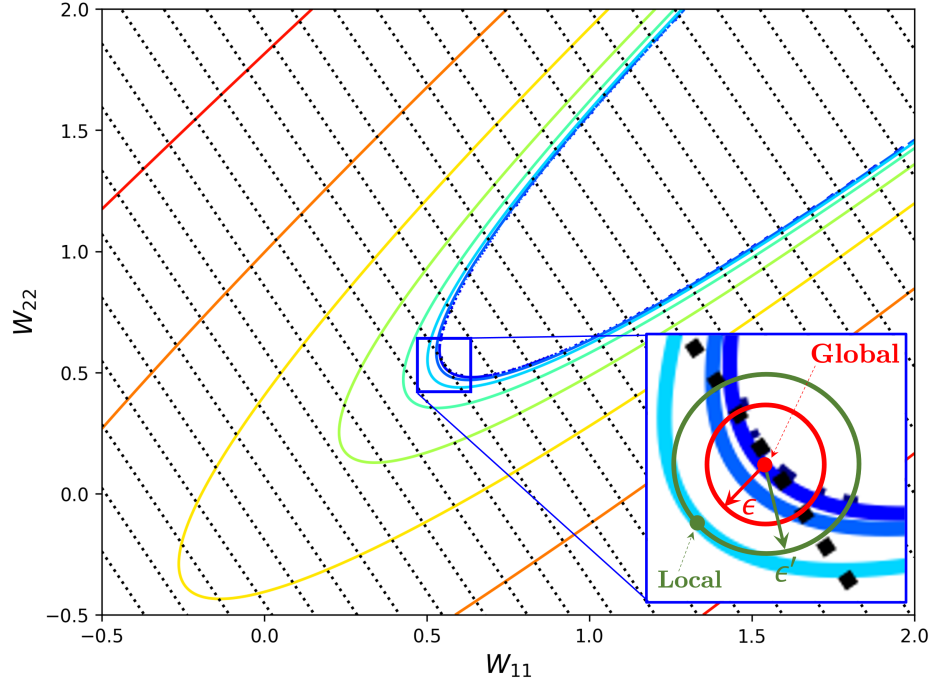


Figure 6: ϵ -approximation of global and local solutions. The green point is positioned outside the ϵ radius. The point is positioned in an expanded region with radius $\epsilon' > \epsilon$. The red point, representing the global solution, is located within the area bounded by the radius ϵ .

By employing Theorem 4.4 and Lemma 4.3, we are equipped to deduce the convergence trajectories of the FSLP algorithm, aiming towards an ϵ -local minimizer for problem (23), delineated in the subsequent section:

Theorem 4.5. *Suppose we possess a sequence (W^k, t^k) formulated by the FSLP algorithm with $\epsilon = 0$, and $(\bar{W}, \bar{t})$ serves as a convergence point for this sequence. In such a case, $\bar{W}$ stands as not merely a KKT point but also an ϵ -local minimizer for the given problem.*

Proof. The second claim in Lemma 4.3 suggests that the sequence $f(W^k)$ is non-ascending. Given that $\mathcal{F}$ is a compact set and W^k is a subset of $\mathcal{F}$, this means that the sequence $f(W^k)$ is bounded and therefore converges, leading to the situation where $f(W^{k-1}) - f(W^k)$ approaches zero. The resulting conclusion is as follows:

$$\lim_{k \rightarrow \infty} \|t^{k-1} - t^k\| = 0. \quad (44)$$

Assume that $(\bar{W}, \bar{t})$ is an accumulation point of the sequence (W^k, t^k) . In such a case, there is a subsequence (W^{k_j}, t^{k_j}) satisfying the following conditions:

$$W^{k_j} \rightarrow \bar{W}, \quad t^{k_j} \rightarrow \bar{t}, \quad \text{where } j \rightarrow \infty$$

From equation (44), we can conclude that as j approaches infinity, t^{k_j-1} approaches $\bar{t}$. Given that $t^k = CW^k$ holds for all k , it follows that $\bar{t} = C\bar{W}$. The compactness of $\mathcal{F}$ implies that $\bar{W}$ is in $\mathcal{F}$. Given that W^{k_j} is an optimal solution to problem (30) with $t = t^{k_j-1}$, we can deduce:

$$\begin{aligned} W^\top A_+ W + b^\top W - 2(t^{k_j-1})^\top CW + \|t^{k_j-1}\|^2 \\ \geq (W^{k_j})^\top A_+ W^{k_j} + b^\top W^{k_j} - 2(t^{k_j-1})^\top CW^{k_j} + \|t^{k_j-1}\|^2, \\ \forall W \in \mathcal{F}. \end{aligned}$$

By applying the limit to both sides of the inequality as j tends towards infinity ($j \rightarrow \infty$), we can derive:

$$\begin{aligned} W^\top A_+ W + b^\top W - 2\bar{t}^\top CW + \|\bar{t}\|^2 \\ \geq \bar{W}^\top A_+ \bar{W} + b^\top \bar{W} - 2\bar{t}^\top C\bar{W} + \|\bar{t}\|^2, \quad \forall W \in \mathcal{F}. \end{aligned}$$

This implies that $\bar{W}$ serves as the optimal solution for equation (30) where $t = \bar{t} = C\bar{W}$, and concurrently acts as an ϵ -local solution as well as a Karush-Kuhn-Tucker (KKT) point for the problem in question. Hence, the proof reaches its conclusion. $\square$

Nonetheless, in scenarios where $r > 1$, the traditional Successive Linear Programming (SLP) algorithm may face challenges in pinpointing the optimal solutions for problem (23). This difficulty stems from the fact that the conventional SLP algorithm depends on scrutinizing the convex hull formed by the accumulation points generated by the algorithm itself. Furthermore, the bounded FSLP algorithm can pinpoint a highly satisfactory feasible solution, which can be employed as an upper limit in the novel Branch-and-Bound (B&B) method.

5. A novel branch-and-bound Algorithm

In this part, we propose a new comprehensive algorithm designed to find a globally optimal solution for problem (??), while maintaining compliance

with a predetermined ϵ -tolerance. This objective is accomplished by amalgamating the FSLP algorithm, the branch-and-bound (B&B) structure, and convex relaxation techniques. Additionally, we guarantee the convergence of our proposed algorithm and examine its complexity.

5.1. Tighter Lifted Convex Relaxation

Convex relaxation of the lifted problem is a methodology employed in optimization theory. This technique involves simplifying a complex or nonconvex problem—referred to as the "lifted" problem, which signifies a problem that has been altered or enhanced to facilitate easier resolution—by approximating it to a more manageable convex problem. This simplification aids in the problem's resolution, typically utilizing approaches that ensure the identification of the global optimum, thereby preventing the solution process from getting entrapped in local optima.

We initiate by looking at a restricted adaptation of the lifted problem, where the variable t is bound by a subbox $[l, u]$, with both l and u belonging to $\mathbb{R}^r$.

$$\begin{aligned} \min \quad & f(W, t) = W^\top A_+ W + b^\top W - \|t\|^2 \\ \text{s.t.} \quad & CW - t = 0, \\ & W \in \mathcal{F}. \quad t \in [l, u] \end{aligned} \tag{45}$$

where $l, u \in \mathbb{R}^r$ with $[l, u] \subseteq [t_l, t_u]$.

Denote the i^{th} row vector of the matrix C as $c_i^\top$, with λ_i representing the absolute value of the negative eigenvalues of A for all $i = \{1, \dots, r\}$, and p_i as the respective orthogonal unit eigenvectors for all $i = \{1, \dots, r\}$. According to the singular value decomposition of A , we get $t_i = c_i^\top W = \sqrt{\lambda_i} p_i^\top W$ for all $i = \{1, \dots, r\}$. Given that $\sum_{i=1}^n p_i p_i^\top = I$, we can deduce:

$$\begin{aligned} \sum_{i=1}^r \frac{s_i}{\lambda_i} &= \sum_{i=1}^r W^\top (p_i p_i^\top) W \leq W^\top \left(\sum_{i=1}^r p_i p_i^\top \right) W \\ &= W^\top W \leq \bar{u}^\top W, \quad \forall W \in \mathcal{F} \end{aligned}$$

where s_i equals the square of t_i , and $\bar{u}$ is the upper boundary of the variable x over the set $\mathcal{F}$. Meanwhile, when t falls within the range $[l, u]$, we have:

$$s_i = t_i^2 \leq (l_i + u_i)t_i - l_i u_i, \quad i = \{1, \dots, r\}.$$

Thus, we can infer the subsequent convex relaxation for the equation (45):

$$\begin{aligned}
\min \quad & f(W, t) = W^\top A_+ W + b^\top W - \sum_{i=1}^r s_i, \\
s.t. \quad & CW - t = 0, \\
& t_i^2 \leq s_i, \\
& s_i \leq (l_i + u_i)t_i - l_i u_i, \\
& \sum_{i=1}^r \frac{s_i}{\lambda_i} \leq \bar{u}^\top W, i = 1, \dots, r, \\
& W \in \mathcal{F}, t \in [l, u].
\end{aligned} \tag{46}$$

Following this, we examine the objective values at the optimal solutions for problem (45) and its relaxation (46). It is found that:

Theorem 5.1. *Consider $f_{[l,u]}^*$ and $v_{[l,u]}^*$ as the optimal values of (45) and its relaxed counterpart (46), respectively. Let's assume $(\bar{W}, \bar{s}, \bar{t})$ to be the optimal solution to (46). In that case, we get:*

$$0 \leq f_{[l,u]}^* - v_{[l,u]}^* \leq f(\bar{W}, \bar{t}) - v_{[l,u]}^* \leq \frac{\|u - l\|^2}{4}.$$

Proof. Clearly, $(\bar{W}, \bar{t})$ is a feasible solution to (45). Since (46) is a convex relaxation of (45), it follows from the choice of $f_{[l,u]}^*$ that

$$\begin{aligned}
0 &\leq f_{[l,u]}^* - v_{[l,u]}^* \leq f(\bar{W}, \bar{t}) - v_{[l,u]}^* \\
&= \sum_{i=1}^r (\bar{s}_i - \bar{t}_i^2) \\
&\leq \sum_{i=1}^r [-\bar{t}_i^2 + (l_i + u_i)\bar{t}_i - l_i u_i] \\
&\leq \frac{1}{4} \sum_{i=1}^r (u_i - l_i)^2 = \frac{1}{4} \|u - l\|^2,
\end{aligned} \tag{47}$$

with the latter inequality being a direct consequence of the constraints placed upon s_i in equation (46). This brings us to the conclusion of our proof. $\square$

In light of the demonstration provided for Theorem 5.1, we can propose the following:

Proposition 5.1. *Given that $(\bar{W}, \bar{s}, \bar{t})$ represents the optimal solution for equation (46) and $\epsilon > 0$. If the condition $\sum_{i=1}^r [\bar{s}_i - \bar{t}_i^2] \leq \epsilon$ is satisfied, then the pair $(\bar{W}, \bar{t})$ qualifies as an ϵ -optimal solution to equation (45).*

Proof. We observe that $(\bar{W}, \bar{t})$ is a feasible solution to problem (45). From equation (47), we have

$$0 \leq f(\bar{W}, \bar{t}) - f_{[l,u]}^* \leq f(\bar{W}, \bar{t}) - v_{[l,u]}^* \leq \sum_{i=1}^r (\bar{s}_i - \bar{t}_i^2) \leq \epsilon.$$

Therefore, we can deduce

$$f(\bar{W}, \bar{t}) \leq f_{[l,u]}^* + \epsilon,$$

signifying that $(\bar{W}, \bar{t})$ qualifies as an ϵ -optimal solution to problem (45). This brings our proof to a close. $\square$

Observe that, given a feasible point W^* , we have the ability to meticulously select a neighborhood $\mathcal{N}(W^*, \epsilon)$ that satisfies the subsequent constraint,

$$t_i \in \left[l_i = (CW^*)_i - \sqrt{\epsilon/r}, u_i = (CW^*)_i + \sqrt{\epsilon/r} \right], \quad (48)$$

In the end, the bound difference can be characterized as a function of the number of eigenvalues and the tolerance level established for the approximation in the quest for the global solution.

$$u_i - l_i - 2\sqrt{\epsilon/r} \leq 0, \quad i = \{1, \dots, r\} \quad (49)$$

From Theorem 5.1, we can obtain the following Theorem.

Theorem 5.2. *Consider $W^* \in \mathcal{F}$ and a neighborhood $\mathcal{N}(W^*, \delta)$ that fulfills all the constraints delineated in (50). Let $(\bar{W}, \bar{s}, \bar{t})$ denote the optimal solution to the relaxed problem (46). In this case, $(\bar{W}, \bar{t})$ stands as an ϵ -local solution to problem (45), relative to the neighborhood $\mathcal{N}(W^*, \epsilon)$.*

Proof. Let us denote the optimal values of (45) and its relaxed counterpart (46) as $f_{[l,u]}^*$ and $v_{[l,u]}^*$, respectively. Given Theorem 5.1, we can infer that

$$f(\bar{W}, \bar{t}) - f_{[l,u]}^* \leq f(\bar{W}, \bar{t}) - v_{[l,u]}^* \leq \frac{\|u - l\|^2}{4}.$$

Together with (50), we derive that

$$f(\bar{W}, \bar{t}) - f_{[l,u]}^* \leq \frac{\|u - l\|^2}{4} \leq \min_{W \in \mathcal{N}(W^*, \epsilon), t \in [l,u]} f(W, t) + \epsilon.$$

Hence, $(\bar{W}, \bar{t})$ qualifies as an ϵ -local solution to (45) in relation to the neighborhood $\mathcal{N}(W^*, \epsilon)$. This brings the proof to its conclusion. $\square$

Theorem 5.2 sets the foundation for segmenting each interval $[t_l^i, t_u^i]$ into discrete subintervals, of which the count can be determined by $\lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \rceil$. These intervals are represented as $[l_i, u_i]$. The definition of t_l^i and t_u^i are guided by their respective formulae. This process results in the decomposition of the feasible domain, $\mathcal{F}$, into a multitude of smaller, manageable subdomains. Each one of these subdivisions is characterized by its unique box constraints, which impose restrictions such that t_i lies within $[l_i, u_i]$ and the constraint width, $u_i - l_i$, does not exceed $2\sqrt{\epsilon/r}$.

After implementing this subdivision, we can address problem (46) within each of these distinct areas, thereby acquiring an ϵ -local approximation for every individual region. Subsequently, from all these locally approximated solutions, we select the one that results in the smallest objective function value to serve as our final solution.

Consequently, the final solution obtained inherently qualifies as a global ϵ -approximate solution to the initial problem (23). Therefore, this segmentation strategy effectively furnishes a viable ϵ -approximation for (23).

For ease of reference, we term this procedure as the “brutal force algorithm”. The “Brute force” refers to a general problem-solving approach in computer science and operations research. A brute force algorithm is a simple, straightforward approach to solve a problem based on the problem’s definition and feasible solutions. It generates all possible solutions for a problem and checks each solution individually to see if it satisfies the problem’s constraints. If the solution satisfies the constraints, it is considered a valid solution; if not, it is discarded, and the algorithm moves on to the next potential solution. The forthcoming theorem validates the aforementioned assertions.

Theorem 5.3. *The “brutal force algorithm” is capable of discovering a global ϵ -approximate solution to (23) within a time complexity of $\mathcal{O}\left(\prod_{i=1}^r \lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \rceil N\right)$, where N represents the complexity involved in the resolution of (46).*

While providing an ϵ -approximate solution, the brutal force algorithm is generally inefficient due to its conservative approach to working with small neighborhoods. Essentially, the algorithm partitions the problem into many tiny subproblems and tries to find local solutions within each one, which could lead to a high computational burden, especially for larger problems. It can be deemed an overkill approach due to the excessive fine-grain partitioning involved.

However, it provides a solid theoretical foundation for building more efficient algorithms. In the following subsection, we will explore how to improve upon this by integrating the Faster FSLP algorithm with other methods. The goal is to develop an enhanced global algorithm for the problem expressed in (23). This approach is expected to be more effective and less computationally demanding and will allow us to handle more complex problems in a reasonable amount of time.

5.2. *The FSLP-embedded Branch-and-Bound algorithm*

In this section, we delve into the architecture of a newly formulated global algorithm that adeptly integrates the Faster Successive Linear Programming (FSLP) approach principles with the well-established branch-and-bound (B&B) techniques. The application of convex quadratic relaxation strategies enhances this integration. For simplicity, we label this innovative hybrid algorithm as FBB.

FBB is endowed with two key characteristics that empower it with a distinctive computational edge:

1. *Global Optimality Verification and Upper Bound Refinement:* The FBB algorithm can verify the global optimality of the solutions procured through FSLP and improve the existing upper bound by invoking a new FSLP process under specific conditions. This unique feature allows greater flexibility and precision in searching for the global optimum.
2. *Acceleration of B&B Convergence Using the FSLP Upper Bound:* FBB leverages the upper bound derived from the FSLP method to accelerate the convergence rate of the B&B approach. This feature significantly reduces the computational resources required, making FBB a much more efficient tool to tackle the problem (23).

Using the strength of these characteristics, FBB provides a robust and effective approach to solving complex optimization problems such as (23).

In the subsequent sections, we will outline the specifics of these features and their application within the FBB framework.

Let's begin by exploring the novel branching technique, which is fundamentally predicated on dividing the feasible rectangle (expressed as Δ) of t in equation (46). The domain Δ is split into multiple subrectangles during each iteration. A conventional approach is to partition the longest side of the subrectangle at its center point. However, this method might not effectively enhance the lower bound as it does not exploit the particular structure of equation (46).

To address this limitation, we propose an adaptive branch-and-cut rule. Take into consideration a sub-rectangle Δ represented as $[l, u]$, and let the optimal solution to equation (46) over Δ be (W^*, s^*, t^*) . If condition $\sum_{i=1}^r s_i^* - (t_i^*)^2 \leq \epsilon$ holds, then by Proposition, W^* is considered an ϵ -optimal solution to (45) over Δ , and further partitioning of this subrectangle will be unnecessary.

On the other hand, if the aforementioned condition is not met, it signifies the existence of at least one i within the set $\{1, \dots, r\}$ such that $s_i^* - (t_i^*)^2 \geq \frac{\epsilon}{r}$. In this case, we select i^* which minimizes the expression $s_i^* - (t_i^*)^2$. Subsequently, the sub-rectangle Δ is divided into two new sub-rectangles, Δ_1 and Δ_2 , along the edge $[l_{i^*}, u_{i^*}]$ at a specific point w . This point w , lying in the interval (l_{i^*}, u_{i^*}) , is referred to as the branching value. Following this, the constraints for the intervals $[l_{i^*}, w]$ and $[w, u_{i^*}]$ are considered individually.

$$s_{i^*} \leq (l_{i^*} + w)t_{i^*} - l_{i^*}w, \quad s_{i^*} \leq (w + u_{i^*})t_{i^*} - wu_{i^*} \quad (50)$$

Referring to Fig. 7, it is clear that the two constraints from equation (50) effectively eliminate (W^*, s^*, t^*) when w is equivalent to $t_{i^*}^*$. Let's define two linear equations, l_1 and l_2 , given by $l_1 : s_{i^*} = (l_{i^*} + w)t_{i^*} - l_{i^*}w$ and $l_2 : s_{i^*} = (w + u_{i^*})t_{i^*} - wu_{i^*}$ respectively. To identify the optimal linear cuts, we formulate an optimization problem (referenced as (??)). The goal of this problem is to minimize the area confined by the parabolic curve $s_{i^*}^* = (t_{i^*}^*)^2$ and the two straight lines l_1 and l_2 , within the interval $[l_{i^*}, u_{i^*}]$.

Algorithm 2: The FBB Algorithm

Input: Initialized with inputs A, b, L, l and a stop criterion $\epsilon > 0$

Output: A global ϵ -solution denoted by W^*

Step 0 (Preparation): Identify r , the eigenvalues of A . If $r > 5$, set $\rho = 2^r$ and for each $j \in \{1, \dots, \rho\}$, let μ_j be a binary vector of size r with elements in $\{-1, 1\}$. If $r > 5$, limit ρ to 2. Consider μ_1 and μ_2 as r -dimensional vectors, where every element of μ_1 is 1, and every element of μ_2 is -1 . For each $j \in \{1, \dots, \rho\}$, solve the problem $\min_{(W, t) \in \mathcal{F}_t} \mu_j^T t$ to derive the optimal solution, denoted as t_j ;

Step 1 (Local Solutions): Identify nearly local ϵ -optimal solutions, referred to as W_j^k , for the problem (23) by executing the FSLP (t^0, ϵ) function, where $t^0 = t_j$ for each $j \in \{1, \dots, \rho\}$. Determine the argument W^* that yields the minimum function value of the set $\{f(W_j^k) \mid j = 1, \dots, \rho\}$. The value of the function in this argument, $f(W^*)$, is assigned to v^* ;

Step 2 (Optimal Solution and Initialization): Solve the problem (46) within the boundaries $[l = t_l, u = t_u]$ to yield an optimal solution (W^0, s^0, t^0) , along with a lower limit v^0 . If $f(W^0) < v^*$, update the upper limit v^* with $f(W^0)$ and replace W^* with W^0 . Set $k := 0$, define l^k and u^k as t_l and t_u respectively. Set the region Δ_k as the interval $[l^k, u^k]$. Initialize the set Ω as $\{[\Delta_k, v^k, (W^k, s^k, t^k)]\}$;

Step 3: while $\Omega \neq \emptyset$ **do**

Phase (I) Selection Node: From the set Ω , select the node

$[\Delta_k, v^k, (W^k, s^k, t^k)]$ with the smallest value of v^k . Remove the node from Ω ;

Phase (II) Termination: If $v^k \geq v^* - \epsilon$ or $f(W^k) - v^k \leq \epsilon$, then both W^* and W^k qualify as ϵ optimal solutions to (23), indicating the algorithm ends;

Phase (III) Partitioning: Identify the index i^* that maximizes $s_i^k - (t_i^k)^2$ for every $i \in \{1, \dots, r\}$. Set w_{i^*} as the midpoint of the interval $[l_{i^*}^k, u_{i^*}^k]$, which equals $\frac{l_{i^*}^k + u_{i^*}^k}{2}$. If the pair $(s_{i^*}^k, t_{i^*}^k)$ is an element of the set $\Gamma_k(w_{i^*})$, assign the branching point β_{i^*} to w_{i^*} . Otherwise, let β_{i^*} be $t_{i^*}^k$. Divide the rectangle Δ_k into two smaller rectangles, $\Delta_k^{k_1}$ and $\Delta_k^{k_2}$, along the edge $[l_{i^*}^k, u_{i^*}^k]$ at the branching point β_{i^*} ;

Phase (IV) Problem Solving and Set Expansion: Solve the problem (46) over each of the two new subrectangles $\Delta_k^{k_1}$ and $\Delta_k^{k_2}$. This yields a lower bound, denoted as v^{k_j} , as well as an optimal solution $(W^{k_j}, s^{k_j}, t^{k_j})$ for each $j = \{1, 2\}$. Append to the set Ω the tuples $[\Delta_k^{k_1}, v^{k_1}, (W^{k_1}, s^{k_1}, t^{k_1})]$ and $[\Delta_k^{k_2}, v^{k_2}, (W^{k_2}, s^{k_2}, t^{k_2})]$;

Algorithm 3: Algorithm Description

while *true* **do**

new subrectangles, $\Delta_k^{k_1}$ and $\Delta_k^{k_2}$. Doing this yields a lower bound, denoted as v^{k_j} , as well as an optimal solution that is represented as $(W^{k_j}, s^{k_j}, t^{k_j})$ for each $j = \{1, 2\}$. After obtaining these values, expand the set Ω by including the results related to the new sub-rectangles. Specifically, append to the set Ω the tuples $[\Delta_k^{k_1}, v^{k_1}, (W^{k_1}, s^{k_1}, t^{k_1})]$ and $[\Delta_k^{k_2}, v^{k_2}, (W^{k_2}, s^{k_2}, t^{k_2})]$, which contain the information about the newly created subrectangles, their corresponding lower bounds, and optimal solutions.

Phase (V) (Reinitiating FSLP and Updating the

Solution): First, define $\hat{W}$ as the solution that minimizes the function f over the optimal solutions of the new subrectangles, that is, $\hat{W} = \arg \min\{f(W^{k_1}), f(W^{k_2})\}$.

if $f(\hat{W}) < v^*$ **then**

This implies that the function value at $\hat{W}$ is less than our current best upper bound v^* . Under this condition, execute the function $\text{FSLP}(t^0, \epsilon)$ with an initial value $t^0 = C\hat{W}$ to find an ϵ -local solution $\bar{W}^k$ for problem (23).

Following this, update the current optimal solution W^* and the upper bound v^* . The updated W^* becomes the argument that minimizes the function f over $\hat{W}$ and $\bar{W}^k$, represented as $W^* = \arg \min\{f(\hat{W}), f(\bar{W}^k)\}$. The upper bound, denoted as v^* , is updated to reflect the value of the function evaluated at the new optimal solution, expressed as $v^* = f(W^*)$.

Phase (VI) (Node Pruning): Here, we remove any redundant or non-competitive nodes from the set Ω . Specifically, remove all nodes denoted by $[\Delta_k^j, v^j, (W^j, s^j, t^j)]$ for which $v^j \geq v^* - \epsilon$. Nodes that meet this criterion are less likely to lead to an improved solution. This is because their lower bound, denoted by v^j , is not significantly lower than the current best solution, v^* , given the specified tolerance, ϵ . Following the pruning process, increment the iteration counter k by one, which means moving on to the next iteration of the algorithm.

Now, we are prepared to delve into the specifics of the innovative global

algorithm designed to deal with problem (23). The answer to this straightforward unconstrained optimization problem is given as follows:

$$S(w) = \frac{1}{2}(u_{i^*} - l_{i^*})w^2 - \frac{1}{2}(u_{i^*}^2 - l_{i^*}^2)w + \frac{1}{6}(u_{i^*}^3 - l_{i^*}^3).$$

Figure 7 illustrates that choosing the midpoint, $w^* = \frac{l_{i^*} + u_{i^*}}{2}$, as the point of partition along the edge $[l_{i^*}, u_{i^*}]$ of the sub-rectangle Δ , minimizes the area $S(w)$. This strategy is preferred for dividing Δ . However, if implementing the secant constraints at $w = w^*$, as shown in equation (50), does not eliminate the solution set (W^*, s^*, t^*) for equation (46) within Δ , the lower bound improvements for either sub-rectangle Δ_1 or Δ_2 could be minimal.

To bypass this issue, an alternative approach is to select $w = t_{i^*}^*$. Specifically, if the constraints defined in equation (50) with $w = w^*$ successfully exclude the solution set (W^*, s^*, t^*) , then Δ should indeed be partitioned at the midpoint w^* . Otherwise, to effectively refine the bounding process, partitioning Δ at $w^* = t_{i^*}^*$, based on the optimal value of the decision variable $t_{i^*}^*$, becomes the preferred strategy.

Now, we are prepared to delve into the specifics of the innovative global algorithm designed to deal with problem (23). In Algorithm 2, Part III, $\Gamma_k(w_{i^*})$ can be represented by two secant cuts:

$$\Gamma_k(w_{i^*}) = \left\{ (s_{i^*}, t_{i^*}) \mid \begin{array}{l} s_{i^*} > (l_{i^*}^k + w_{i^*})t_{i^*} - l_{i^*}^k w_{i^*} \\ s_{i^*} > (w_{i^*} + u_{i^*}^k)t_{i^*} - w_{i^*} u_{i^*}^k \end{array} \right.$$

The following observations pertain to the stages of the FBB algorithm:

1. To begin, we employ the FSLP algorithm, utilizing various feasible points from the higher-level problem as starting points, with the aim of discovering a satisfactory ϵ -local solution for equation (23).
2. During the first phase of the third step, the FSLP algorithm is reinitialized to locate a more favorable ϵ -local solution, on the condition that the objective value at the feasible point derived from the solution of the relaxation problem is lower than the current upper limit.
3. Subsequently, in the third phase of the third step, we proceed to prune the optimal solution for the relaxation problem after each iteration in order to improve the lower bound.

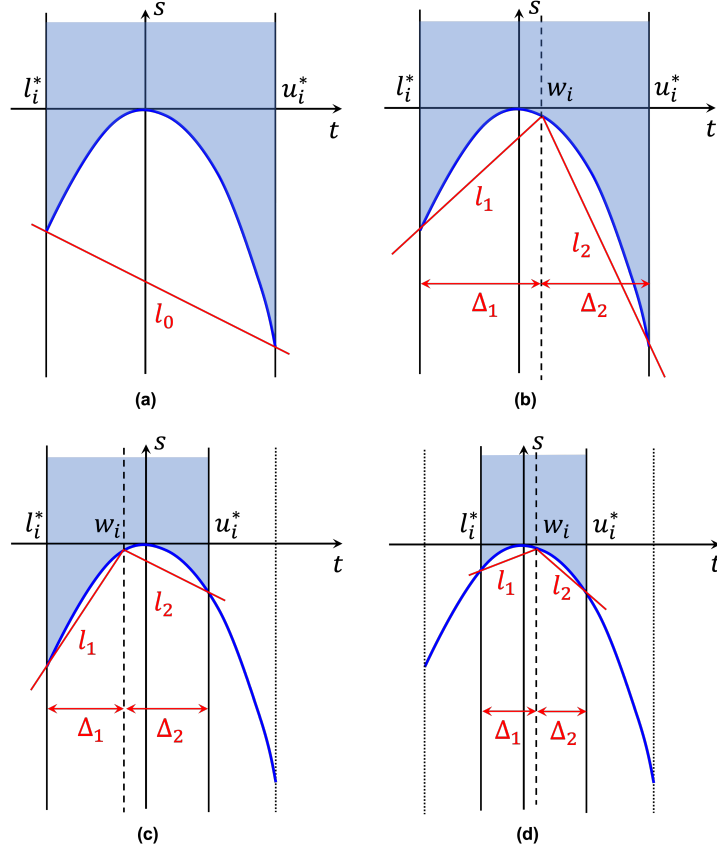


Figure 7: Partition on the variable t_{i^*} .

We will now demonstrate the convergence of the FBB Algorithm. Let's consider a sub-rectangle, denoted as $\bar{\Delta} = [\bar{l}, \bar{u}]$, that complies with the following condition.

$$\bar{u}_i - \bar{l}_i - 2\sqrt{\epsilon/r} \leq 0 \quad \forall i \in \{1, \dots, r\}. \quad (51)$$

Assume that $(\bar{W}, \bar{s}, \bar{t})$ is the optimal solution for equation (46) over the range of $\bar{\Delta}$. As stated in equation (47), we find that,

$$\bar{s}_i - \bar{t}_i^2 \leq \frac{1}{4}(\bar{u}_i - \bar{l}_i)^2, i = \{1, \dots, r\}, \quad (52)$$

This suggests that,

$$\sum_{i=1}^r [\bar{s}_i - (\bar{t}_i)^2] \leq \epsilon. \quad (53)$$

Based on Proposition 5.1, we can affirm that $\bar{W}$ is an ϵ -optimal solution for equation (45) within the domain of $\bar{\Delta}$. In alignment with the third and fourth procedures of the FBB algorithm, this specific sub-rectangle $\bar{\Delta}$ will be conclusively removed from the set Ω . This signifies that a sub-rectangle Δ that complies with equation (51) will no longer be partitioned in the course of the algorithm.

In the event that the optimal solution $(\bar{W}, \bar{s}, \bar{t})$ of a current sub-problem fails to meet the requirements of equation (53) and necessitates further investigation, there must exist an index i^* within the set $\{1, \dots, r\}$ such that $\bar{s}_{i^*} - \bar{t}_{i^*}^2 > \frac{\epsilon}{r}$. As per the equation, this implies that $\bar{u}_{i^*} - \bar{l}_{i^*} > 2\sqrt{\frac{\epsilon}{r}}$, signifying that the length of the i^* -th edge of the corresponding sub-rectangle must be greater than $2\sqrt{\frac{\epsilon}{r}}$.

As specified in Step 3, Phase III, it will proceed to division at the point $\bar{t}_{i^*}$ or at the midpoint of this edge. In addition, the length of any edge of the subrectangle associated with a subproblem that needs to be solved will always be greater than or equal to $2\sqrt{\epsilon/r}$.

Consequently, every edge of the starting rectangle is divided into no more than $\left\lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \right\rceil$ segments. This means that to obtain an ϵ -accurate solution for equation (23), the quantity of simplified subproblems that must be addressed with the FBB algorithm is bounded by,

$$\prod_{i=1}^r \left\lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \right\rceil.$$

Based on the prior discussion, it can be concluded that,

Theorem 5.4. *The FBB Algorithm is capable of discovering a global ϵ -approximate solution to the given problem within a maximum time frame of $\mathcal{O}(\prod_{i=1}^r \left\lceil \frac{\sqrt{r}(t_u^i - t_l^i)}{2\sqrt{\epsilon}} \right\rceil N)$, where N signifies the complexity associated with solving problem (46).*

6. Computational Results

The effectiveness of the FSLP and FBB algorithms for providing precise solutions for instances in the `PG-lib` library [41] has been illustrated through numerous tests and a comparison made with the nonlinear solver `Baron 23.6.15` [42] and the nonconvex mode of the `Gurobi 10` [46] solver. These tests were carried out on a server equipped with dual Intel Xeon CPUs,

each with 12 cores and 2 threads operating at 2.5 GHz, accompanied by $4 \times 16\text{GB}$ RAM and a Linux-based operating system. A maximum computation time of 3 hours was permitted for all methods. Baron utilized the multithreaded variant of **CPLEX20.10** [43], supporting up to 64 threads. On the other hand, the FSLP and FBB methods incorporated the **Mosek** [44] semidefinite solver to solve Penalized semidefinite programs. At each branch and bound node, distinct solvers were used for solving the LCQP of the FBB method and the **CPLEX 20.10** solver was employed for solving the QP of the FSLP method. For computing feasible local solutions, **Ipopt** [45] was utilized as a local solver.

The tests were conducted utilizing data from medium-sized power networks ranging from 3 to 300 buses, following the ACOPF formulation defined by constraints (19)-(21). Table 1 offers comprehensive characteristics for each test case, including the name of the cases, the number of buses, generators, and lines in the examined power network. It also displays the optimal solution discerned by the FSLP and FBB methods within the allotted 3 hours of computing time.

We employ a unique performance profile that measures CPU times to evaluate the efficiency of different solvers. The fundamental concept is as follows: for every test case n and each solver s , we designate t_{ns} as the time taken to solve the test case n using solver s . We then establish the performance ratio as κ_{ns} . Assuming that N_{tc} represents the total count of test cases analyzed, a comprehensive evaluation of solver s performance for a specified τ is determined as follows.

$$P(\kappa_{ns} \leq \tau) = \frac{\text{the count of test cases}}{N_{tc}} \quad (54)$$

$$\kappa_{ns} = \frac{t_{ns}}{\min_s t_{ns}} \quad (55)$$

Fig. 8 presents the performance profile that details the CPU times for the **IPOPT**, **Baron 23.6.15**, and **Gurobi 10** methods applied to power system test cases. From our observations, it is clear that the proposed LCQP-ACOPF algorithm with bounded feasible SLP and **Gurobi 10** surpass the comparative **Baron 23.6.15** solver and the proposed Penalized-SDP ACOPF integrated with an unbounded SLP algorithm from [47] in aspects of CPU time utilization and the number of test cases resolved. Specifically, Baron can only optimally solve 6 of the 20 test cases considered, with **case14-ieee** being the largest among them. The two methods, LCQP-ACOPF and Gurobi,

Name	$ \mathcal{N} $	$ \mathcal{G} $	$ \mathcal{L} $	Opt
caseWB2	2	1	1	12.262
caseWB3	3	1	2	587.340
pglib_case3_lmbd	3	3	3	5655.465
caseWB5	5	2	6	9.076
pglib_case5_pjm	5	5	5	17623.048
case6ww	6	3	11	4074.121
pglib_case14_ieee	14	5	20	2665.655
pglib_case24_ieee_rts	24	33	38	52006.217
pglib_case30_as	30	6	41	472.612
pglib_case30_ieee	30	6	41	5485.280
pglib_case39_epri	39	10	46	168303.267
pglib_case57_ieee	57	7	80	33629.932
pglib_case73_ieee_rts	73	99	120	183995.780
pglib_case89_pegase	89	12	210	109232.650
pglib_case118_ieee	118	54	186	96881.525
pglib_case162_ieee_dtc	162	12	284	89234.092
pglib_case179_goc	179	29	263	77442.263
pglib_case200_activ	200	38	245	26231.327
pglib_case240_pserc	240	143	448	49.12
pglib_case300_ieee	300	69	411	521.98

Table 1: The attributes of the power test cases under consideration in the PG-lib library.

prove to be more effective, as they manage to solve 12 test cases with the unbounded-SLP, 13 test cases using Gurobi, and 14 using LCQP-ACOPF within a 3-hour CPU time limit. Furthermore, the performance profile indicates that the LCQP-ACOPF method operates faster than Gurobi for these power test cases.

For test cases involving 57 and 200 buses, we extended the iteration time to 50 and analyzed the total summation of all the principal 2×2 minor determinants in the objective function as a vanishing regularizer. The trend pertaining to the sum of all principal 2×2 minor determinants is illustrated in Fig. 9. A notable exponential decline in the total sum of all principal minor determinants is visible within the first 25 iterations. This suggests that the FBB algorithm’s solution could potentially attain feasibility for the lifted-ACOPF problem following a series of cut generations across the (Δ_1, Δ_2)

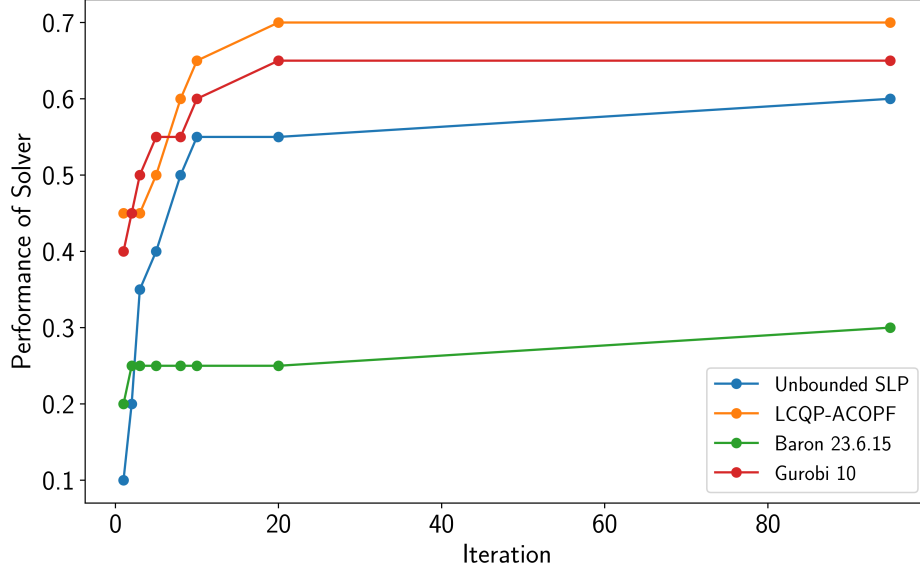


Figure 8: The solver performance profile concerning the cumulative duration for networks, ranging from 2 to 300 bus systems, within a time boundary of 3 hours for the proposed LCQP-ACOPF model, presented ACOPF model with unbounded-SLP in [47], Baron and Gurobi.

edges within the branch-and-bound algorithm.

In Tables 2 and 3, we conduct an exhaustive comparison between the LCQP-ACOPF and ACOPF methodologies, as well as the unbounded-SLP as delineated in [47]. The “Gap” column is defined by the equation:

$$\text{Gap} = \left| \frac{\text{Opt} - \text{Cont}}{\text{Opt}} \right| \times 100 \quad (56)$$

This section describes key metrics and observations from an experimental analysis within the branch-and-bound framework for solving optimization problems, specifically focusing on ACOPF challenge using the LCQP model. The terminology used includes:

- **Gap:** This term represents the initial difference encountered at the root node of the branch-and-bound process.
- **Cont:** Denotes the highest value achieved through rank relaxation.
- **Opt:** Defined according to Tables 2 and 3, referring to the optimal solution.

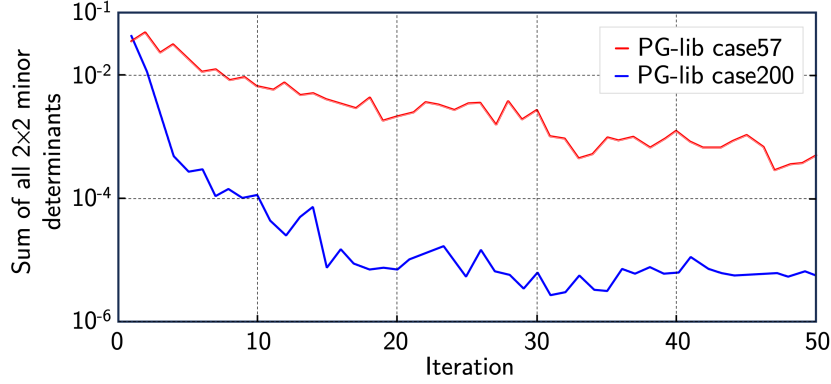


Figure 9: Total summation of all principal 2×2 minor determinants in the objective function over iterations of the 57-bus system and the 200-bus systems.

- The notation “-” indicates that the algorithm failed to find a feasible solution.
- **Aux**: Shows the total number of auxiliary variables added in the relaxation process, equivalent to the total number of cuts applied in the spatial branch-and-bound method.
- **UB** and **P-SDP**: Document the time in seconds required to establish an initial upper bound and to solve the rank-one constraint relaxation on a standard CPU, respectively.
- **CPU**: Aggregates the total CPU time used, with “-” marking cases unsolved within a three-hour limit.
- **Nodes**: Reflects the number of nodes explored during the branch-and-bound process.

Key findings from this analysis include the following.

1. The process of reformulating the problem is notably swift, often taking less than 428 seconds. This efficiency is attributed to the use of conventional semidefinite solvers to directly tackle rank relaxation.
2. In 9 out of 20 power test cases, there was no initial gap at the root node, highlighting the effectiveness of rank relaxation in addressing the OPF challenge. However, difficulties arise in precisely solving cases with a non-zero initial gap.

Test Case		LCQP-ACOPF				
ID	Gap	Aux	UB	P-SDP	CPU	BB Nodes
pglib_caseWB2	2.011	5	0	1	2	27
pglib_caseWB3	0.000	7	0	2	2	2
pglib_case3_Imbd	1.670	7	0	2	1	2
pglib_caseWB5	14.481	13	0	2	360	8279
pglib_case5_pjm	0.000	13	0	1	1	0
pglib_case6ww	0.000	14	5	1	6	0
pglib_case14_ieee	0.100	32	0	2	1	0
pglib_case24_ieee_rts	0.000	52	0	2	1	1
pglib_case30_as	0.050	62	8	0	8	0
pglib_case30_ieee	0.000	62	0	1	1	0
pglib_case39_epri	0.000	80	1	1	2	0
pglib_case57_ieee	0.05	115	167	1	-	2659
pglib_case73_ieee_rts	0.000	146	67	2	70	0
pglib_case89_pegase	0.000	180	31	3	34	0
pglib_case118_ieee	2.102	238	933	2	-	973
pglib_case162_ieee_dtc	2.470	326	334	6	-	326
pglib_case179_goc	0.024	360	1517	5	-	343
pglib_case200_activ	0.000	402	423	5	428	0
pglib_case240_pserc	2.470	482	3634	27	-	219
pglib_case300_ieee	1.090	612	3278	42	-	34

Table 2: The initial gap, sizes of the power network, CPU times, and the number of nodes used in the branch-and-bound algorithm (FBB) for the LCQP-ACOPF.

3. The approach cited in [47] had trouble solving smaller test cases with only 2 or 5 buses, due to its inability to improve the lower bound despite exploring many nodes.
4. Conversely, the LCQP-ACOPF method, while examining fewer nodes, was successful in progressively enhancing the lower bound for even the largest test cases. This success is largely due to the efficiency of the method in reducing the number of auxiliary variables and simplifying the relaxed equalities ($s_i = t_i^2$).
5. Achieving a feasible solution can be challenging, particularly when using a generic interior point algorithm. The CPU time to find a feasible solution varies significantly, exemplified by the `case200_activ` test

Test Case		Unbounded-SLP in [47]				
ID	Gap	Aux	UB	P-SDP	CPU	BB Nodes
pglib_caseWB2	2.011	10	0	1	-	44265
pglib_caseWB3	0.000	18	0	1	1	0
pglib_case3_Imbd	1.670	23	0	1	1	0
pglib_caseWB5	14.481	38	0	1	-	292450
pglib_case5_pjm	0.000	36	0	1	1	0
pglib_case6ww	0.000	60	0	2	2	0
pglib_case14_ieee	0.100	110	0	1	4	0
pglib_case24_ieee_rts	0.000	186	56	1	55	0
pglib_case30_as	0.050	226	0	1	1	0
pglib_case30_ieee	0.000	230	9	1	8	0
pglib_case39_epri	0.000	248	9	1	7	0
pglib_case57_ieee	0.05	445	35	1	-	2278
pglib_case73_ieee_rts	0.000	580	191	3	142	0
pglib_case89_pegase	0.000	1004	72	3	76	0
pglib_case118_ieee	2.102	952	2866	5	-	1023
pglib_case162_ieee_dtc	2.470	1444	524	9	-	1045
pglib_case179_goc	0.024	1246	3050	10	-	1132
pglib_case200_activ	0.000	1284	2468	10	2176	0
pglib_case240_pserc	2.470	1872	3330	29	-	819
pglib_case300_ieee	1.090	2334	3743	52	-	3171

Table 3: The initial gap, sizes of the power network, CPU times, and the number of nodes used in the LCQP-ACOPF with unbounded-SLP methods, as presented in the ACOPF from [47].

case, where it ranged from 400 to 1380 seconds.

These insights not only demonstrate the complexities and challenges involved in solving the ACOPF problem, but also underscore the potential of the LCQP-ACOPF method in achieving efficient and effective solutions under certain conditions.

7. Conclusion

The article explores the challenging LCQP model as it applies to the ACOPF problem, which involves complex variables with negative eigenvalues and is known for its NP-hard status, drawing from various fields. In

our study, we have combined a series of advanced techniques: a feasible successive linear programming (FSLP) approach, robust convex relaxation methods, initial setup processes, and the branch-and-bound (B&B) method to develop the FBB algorithm. This algorithm is specifically designed to find a globally optimal solution for the fundamental LCQP-based ACOPF model. Furthermore, we have developed a new joint algorithm that operates independently of the B&B framework. This innovative algorithm aims to discover a global optimum for the LCQP-ACOPF problem by integrating FSLP, convex relaxation, and a unique B&B approach. With certain basic assumptions, we have verified the global effectiveness of both the FBB and FSLP algorithms and estimated their computational complexity. Early experimental results confirm the success of FSLP and FBB in consistently finding the global optimum for all the power system scenarios described in our simulations. It is crucial to note that the complexity of the FBB algorithm increases exponentially with the number of negative eigenvalues in the Hessian matrix of the objective function, suggesting potential inefficiencies for power test cases with many negative eigenvalues. Our research shows that a specific implementation of our approach allows for the use of carefully penalized semidefinite relaxation, leading to much tighter convex relaxations compared to traditional methods. This significantly improves our handling of medium-sized real-world power test cases related to the LCQP-ACOPF problem. Future research could fruitfully explore advanced global algorithms for the broader ACOPF problem by integrating FSLP with additional branching and bound or local search strategies. Another promising area of investigation involves creating similar global solutions for generic quadratic programming problems with non-convex constraints.

References

- [1] M. B. Cain, R. P. O'Neill, and A. Castillo, *History of optimal power flow and formulations*, 2012.
- [2] J. Carpentier, *Contribution à l'étude du dispatching économique*, Bulletin de la Société Française des Électriciens, vol. 3, pp. 431-447, 1962.
- [3] D. Bienstock and A. Verma, *Strong np-hardness of ac power flows feasibility*, Oper. Res. Lett., vol. 47, pp. 494-501, 2019.

- [4] H. Konno, *A cutting plane algorithm for solving bilinear programs*, Mathematical Programming, vol. 11, pp. 14-27, 1976.
- [5] F. Palacios-Gómez, L. Lasdon, and M. Engquist, *Nonlinear optimization by successive linear programming*, Management Science, vol. 28, pp. 1106-1120, 1982.
- [6] Y. Ye, *On the complexity of approximating a kkt point of quadratic programming*, Mathematical Programming, vol. 80, pp. 195-211, 1998.
- [7] K. Anstreicher, *Semidefinite programming versus the reformulationlinearization technique for nonconvex quadratically constrained quadratic programming*, Journal of Global Optimization, vol. 43, pp. 471-484, 2009.
- [8] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization*, SIAM J. Optim., vol. 1, pp. 166-190, 1991.
- [9] J. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., vol. 11, pp. 796-817, 2001.
- [10] M. Kojima and L. Tunçel, *Cones of matrices and successive convex relaxations of nonconvex sets*, SIAM J. Optim., vol. 10, pp. 750-778, 2000.
- [11] M. Kojima and L. Tunçel, *Discretization and localization in successive convex relaxation methods for nonconvex quadratic optimization*, Mathematical Programming, vol. 89, pp. 79-111, 2000.
- [12] J. Peng and T. Zhu, *A nonlinear semidefinite optimization relaxation for the worst-case linear optimization under uncertainties*, Mathematical Programming, vol. 152, pp. 593-614, 2015.
- [13] A. Saxena, P. Bonami, and J. Lee, *Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations*, Mathematical Programming, vol. 130, pp. 359-413, 2011.
- [14] S. Burer and D. Vandenbussche, *A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations*, Mathematical Programming, vol. 113, pp. 259-282, 2008.

- [15] S. Burer and D. Vandenbussche, *Globally solving box-constrained non-convex quadratic programs with semidefinite-based finite branch-and-bound*, Computational Optimization and Applications, vol. 43, pp. 181-195, 2009.
- [16] J. Chen and S. Burer, *Globally solving nonconvex quadratic programming problems via completely positive programming*, Mathematical Programming Computation, vol. 4, pp. 33-52, 2012.
- [17] H. Luo, X. Bai, G. Lim, and J. Peng, *New global algorithms for quadratic programming with a few negative eigenvalues based on alternative direction method and convex relaxation*, Mathematical Programming Computation, vol. 11, pp. 119-171, 2019.
- [18] D. Molzahn and I. Hiskens, *A survey of relaxations and approximations of the power flow equations*, 2019.
- [19] J. Lavaei and S. H. Low, *Zero duality gap in optimal power flow problem*, IEEE Transactions on Power Systems, vol. 27, no. 1, pp. 92-107, 2012.
- [20] M. Ma, L. Fan, Z. Miao, B. Zeng, and H. Ghassempour, *A sparse convex AC OPF solver and convex iteration implementation based on 3-node cycles*, Electric Power Systems Research, vol. 180, 2020.
- [21] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, *Semidefinite programming for optimal power flow problems*, International Journal of Electrical Power & Energy Systems, vol. 30, pp. 383-392, 2008.
- [22] T. Liu, J. Sun, and D. Tsang, *Rank-one solutions for sdp relaxation of qcqps in power systems*, IEEE Transactions on Smart Grid, vol. 10, pp. 5-15, 2019.
- [23] H. Luo, S. Chen, and H. Wu, *A new branch-and-cut algorithm for non-convex quadratic programming via alternative direction method and semidefinite relaxation*, Numerical Algorithms, pp. 1-32, 2021.
- [24] L. Gan, N. Li, U. Topcu, and S. H. Low, *Exact convex relaxation of optimal power flow in radial networks*, IEEE Trans. Autom. Control, vol. 60, no. 1, pp. 72-87, Jan. 2015.

- [25] C. Coffrin and P. Van Hentenryck, *A linear-programming approximation of ac power flows*, INFORMS J. Comput., vol. 26, no. 4, pp. 718-734, 2014.
- [26] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, *The QC relaxation: A theoretical and computational study on optimal power flow*, IEEE Trans. Power Syst., vol. 31, no. 4, pp. 3008-3018, Jul. 2016.
- [27] R. A. Jabr, *Radial distribution load flow using conic programming*, IEEE Trans. Power Syst., vol. 21, no. 3, pp. 1458-1459, Aug. 2006.
- [28] B. Kocuk, S. S. Dey, and X. A. Sun, *Strong socp relaxations for the optimal power flow problem*, Oper. Res., vol. 64, no. 6, pp. 1177-1196, 2016.
- [29] F. Zohrizadeh, M. Kheirandishfard, E. Q. Jnr and R. Madani, *Penalized Parabolic Relaxation for Optimal Power Flow Problem*, 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 2018, pp. 1616-1623, doi: 10.1109/CDC.2018.8619374.
- [30] Y. Zhang, R. Madani and J. Lavaei, *Conic Relaxations for Power System State Estimation With Line Measurements*, in IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1193-1205, Sept. 2018, doi: 10.1109/TCNS.2017.2691473.
- [31] S. Sojoudi, R. Madani and J. Lavaei, *Low-rank solution of convex relaxation for optimal power flow problem*, 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm), Vancouver, BC, Canada, 2013, pp. 636-641, doi: 10.1109/SmartGridComm.2013.6688030.
- [32] R. Horst and N. V. Thoai, *DC programming: Overview*, J. Optim. Theory Appl., vol. 103, no. 1, pp. 1-43, 1999.
- [33] H. A. Le Thi and T. Pham Dinh, *Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms*, J. Global Optim., vol. 11, pp. 253-285, 1997.
- [34] P. M. Pardalos and S. A. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim., vol. 1, pp. 15-22, 1991.

- [35] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [36] F. Kittaneh and Y. Manasrah, *Improved Young and Heinz inequalities for matrices*, Journal of Mathematical Analysis and Applications, vol. 361, no. 1, pp. 262-269, 2010, [Online]. Available: <https://doi.org/10.1016/j.jmaa.2009.08.059>
- [37] C. A. Floudas and V. Visweswaran, *Quadratic optimization*, in Handbook of Global Optimization, R. Horst and P. M. Pardalos, Eds., Kluwer Academic Publishers, pp. 217-270, 1994.
- [38] J. Gorski, F. Pfeuffer, and K. Klamroth, *Biconvex sets and optimization with biconvex functions: A survey and extensions*, Math. Methods Oper. Res., vol. 66, pp. 373-407, 2007.
- [39] V. Visweswaran and C. Floudas, *New properties and computational improvement of the GOP algorithm for problems with quadratic objective function and constraints*, J. Global Optim., vol. 3, no. 3, pp. 439-462, 1993.
- [40] P. Loridan, *Necessary conditions for ϵ -optimality*, Math. Program. Stud., vol. 19, pp. 140-152, 1982.
- [41] The IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, *PGLib Optimal Power Flow Benchmarks*, Accessed: Feb. 2021.
- [42] N.V. Sahinidis and M. Tawarmalani, *Baron 9.0.4: Global optimization of mixed-integer nonlinear programs. User's Manual*, 2010.
- [43] IBM-ILOG, *IBM ILOG CPLEX 12.9 Reference Manual*, 2020.
- [44] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.2.*, 2019.
- [45] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., vol. 106, no. 1, pp. 25-57, Mar. 2006.
- [46] L. Gurobi Optimization, *Gurobi Optimizer Reference Manual 10.0*, 2023. [Online]. Available: <https://www.gurobi.com>

- [47] M. Zhao and M. Barati, *A Sensitive-eigenvalues based Global Algorithm for Penalized SDP Relaxation of ACOPF*, in 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2021, pp. 1-5, doi: 10.1109/ISGT49243.2021.9372234.