A Sequential $\ell_1$ Quadratic Programming Method for Robust Nonlinear Optimal Power Flow Solution

Vicente R. Simoni  
Companhia Hidro Elétrica do São Francisco  
Recife, Pernambuco  
Email: vicentes@chesf.gov.br

Geraldo L. Torres  
Universidade Federal de Pernambuco  
Recife, Pernambuco  
Email: gltorres@ufpe.br

Abstract—Large-scale nonlinear optimal power flow (OPF) problems have been solved lately by primal-dual interior point (IP) methods. In spite of their success, there are many situations in which IP-based OPF programs can fail to find a solution. On the other hand, with power systems operating heavily loaded there is an increasing need for globally convergent OPF solvers. Trust region schemes have been used to enforce convergence, but they are by nature computationally expensive. This paper aims at developing a trust region OPF algorithm less expensive than the one proposed by Sousa et al. The major difference lies in how they handle inconsistent constraints in the solution of the trust region subproblems. The algorithm proposed here employs a sequential $\ell_1$ quadratic programming ($S\ell_1$ QP) approach, while the one of Sousa et al. employs the Byrd-Omojokun technique. Thus, rather than solving two quadratic programming (QP) problems per iteration as in the Byrd-Omojokun technique, the $S\ell_1$ QP approach solves a single, but slightly larger, QP problem. The developed algorithm is tested on the IEEE test systems of up to 300-bus, with all QP problems solved by primal-dual IP algorithms. The numerical results indicate that the $S\ell_1$ QP method is competitive in processing time when compared to the Byrd-Omojokun approach.

Index Terms—Optimal Power Flow, Global Convergence, Trust Region Methods, Sequential $\ell_1$ Quadratic Programming.

I. INTRODUCTION

Optimal power flow (OPF) problems lay in the class of large-scale non convex nonlinear programming (NLP) problems and, with the increasing loading of power systems, there is a growing need for convergence robustness when solving OPF problems. In the last two decades, nonlinear OPF problems have been successfully solved by primal-dual interior point (IP) methods [1], mainly by the predictor-corrector [2] and the multiple centrality corrections [3], [4] variants. However, there are many situations under which locally convergent OPF solvers may fail to obtain a solution [5], e.g., when the initial guess is not sufficiently close to a solution. Fortunately, globally convergent techniques can be used to ensure convergence from remote starting points.

Line search and trust region techniques are two important descent schemes for guaranteeing global convergence [6]. Each iteration of a line search method computes a search direction and then decides how far to move along that direction by defining a step length. The success of a line search procedure depends on effective choices of both the search direction and the step length. On the other hand, trust region methods define a region around the current iterate within which they trust the model to be an adequate representation of the objective function and constraints and then choose the step to be an approximate minimizer of the model in this region.

Trust region methods [7], [8] have been used to provide global convergence to a great diversity of algorithms, from unconstrained to constrained optimization. They work by transforming a general NLP problem into a sequence of well-behaved subproblems, called trust region subproblems, by approximating the objective and constraints functions by quadratic and linear models, and then adding a trust region constraint to ensure the validity of this model. The resulting subproblems are commonly quadratic and can be solved by well-known methods for quadratic programming (QP). Thus, solving a general NLP problem by a trust region method involves outer and inner iterations. In each outer iteration a trust region subproblem is formulated. The inner iterations are those to solve the formulated trust region subproblems. Quite often, mainly in the initial outer iterations, the feasible set of the trust region subproblem is empty because the constraints are incompatible, i.e., the linearized constraints do not intercept the trust region constraint [9]. Several approaches have been proposed to handle inconsistent constraints in trust region subproblems [10].

Applications of trust region methods to power systems optimization gained attention in the last few years. Pajić and Clements [11], Hassaïne et al. [12] and Costa et al. [13] used trust region methods to increase the robustness of state estimation algorithms. Min et al. [14], Wang et al. [15], Giacomoni and Wollenberg [16] and Torres [17] presented applications of trust region methods to solve OPF problems, including market-based OPF and the classical active power losses minimization. Recently, Souza et al. [18] applied the Byrd-Omojokun trust region technique to solve nonlinear OPF problems. The Byrd-Omojokun technique handles incompatible constraints by dividing the subproblem into two subproblems, known as the vertical and horizontal subproblems. The vertical subproblem focus on achieving feasibility, as it minimizes the squared residuals of the linearized equality constraints within a reduced trust region. The goal of the horizontal subproblem is to make the same progress the vertical subproblem does towards satisfying the linearized equality constraints, while minimizing the objective function of the trust region subproblem.
This paper aims at reducing the computational effort of the trust region OPF algorithm in [18]. Towards this end, it solves only one QP problem per trust region iteration, instead of solving two as in Byrd-Omojokun approach. This is done by handling incompatible constraints in trust region subproblems using the sequential $\ell_1$ quadratic programming (S$\ell_1$QP) technique proposed by Fletcher [19]. In other words, the goal of this paper is to develop a globally convergent trust region OPF algorithm using the S$\ell_1$QP technique, as it appears to be computationally less expensive than the Byrd-Omojokun approach used in [18]. In each iteration, instead of splitting the trust region problem into two subproblems, like the Byrd-Omojokun technique does, the S$\ell_1$QP technique solves just one slightly larger QP problem, which invariably takes less time than solving the vertical and horizontal subproblems.

The paper unfolds as follows. Section II discusses the basic ideas of trust region strategies for constrained optimization. Section III briefly describes the Byrd-Omojokun approach. Section IV presents the proposed S$\ell_1$QP algorithm and the solution of QP problems by primal-dual IP methods. Section V presents the numerical results obtained with the proposed technique. Section VI closes the paper with final conclusions.

II. TRUST REGION METHODS

This paper is concerned with the robust solution of nonlinear OPF problems that can be mathematically stated in the form:

$$\begin{align*}
\min & \quad f(x) \\
\text{s. t.} & \quad g(x) = 0 \\
& \quad \underline{x} \leq x \leq \bar{x}
\end{align*}$$

where $x \in \mathbb{R}^n$ is a vector of decision variables; $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a vector of nonlinear functions including the power balance equations and other equality constraints; $\underline{x}$ and $\bar{x}$ are vectors of lower and upper bounds on the variables $x$, corresponding to physical and operational limits.

In order to solve (1), trust region methods generate a sequence of minimization steps with the help of a quadratic model. These methods define a region with radius $\Delta_k$ around the current point $x_k$ within which they trust the model to be an adequate representation of (1). For the NLP problem (1), the trust region subproblem can be defined as follows:

$$\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\
\text{s. t.} & \quad g(x_k) + \nabla g(x_k)^T d = 0 \\
& \quad \underline{x} \leq x_k + d \leq \bar{x} \\
& \quad \|d\| \leq \Delta_k
\end{align*}$$

where $H_k$ is the Hessian matrix of the Lagrange function associated with problem (1), i.e.,

$$H_k = \nabla^2 f(x_k) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x_k)$$

where $\lambda_i$ is the Lagrange multiplier associated with $g_i(x) = 0$.

In order to obtain a convex trust region subproblem, with globally convergent algorithms to solve it, the infinity norm is employed in the trust region constraint, as suggested in [18]. Thus, the trust region subproblem becomes the QP problem:

$$\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\
\text{s. t.} & \quad g(x_k) + \nabla g(x_k)^T d = 0 \\
& \quad \underline{x} - x_k \leq d \leq \bar{x} - x_k \\
& \quad \|d\|_{\infty} \leq \Delta_k
\end{align*}$$

Since $\|d\|_{\infty} = \max |d|$, the constraints (4c) and (4d) on step $d$ can be combined into a single simple bound constraint, so that problem (4) can be rewritten as:

$$\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\
\text{s. t.} & \quad g(x_k) + \nabla g(x_k)^T d = 0 \\
& \quad \max \{\delta, -\Delta_k\} \leq d \leq \min \{\bar{\delta}, \Delta_k\}
\end{align*}$$

where $\delta = \underline{x} - x_k$ and $\bar{\delta} = \bar{x} - x_k$.

In principle, problem (5) can be directly solved by any QP solver. However, the trust region constraint (5c) may limit the step $d$ in such a way that problem (5) becomes infeasible, i.e., there is no step $d$ that satisfies both (5b) and (5c). Several approaches have been proposed to handle the possible conflicts between satisfying the linearizations of the original constraints and the trust region constraint [10].

III. THE BYRD-OMOJOJUK TECHNIQUE

To resolve possible inconsistencies with constraints (5b) and (5c), the Byrd-Omojokun method [9] divides (5) into two subproblems, known as vertical and horizontal subproblems.

A. Vertical Subproblem

The vertical subproblem is defined as follows:

$$\begin{align*}
\min & \quad \frac{1}{2} \|g(x_k) + \nabla g(x_k)^T v\|^2 \\
\text{s. t.} & \quad \max \{\delta, -\xi \Delta_k\} \leq v \leq \min \{\bar{\delta}, \xi \Delta_k\}
\end{align*}$$

where $\xi \in (0, 1)$ is a reduction factor of the trust region. By ignoring the constant term $g(x_k)^T g(x_k)$ in the objective, problem (6) can be rewritten as follows:

$$\begin{align*}
\min & \quad (\nabla g(x_k) g(x_k))^T v + \frac{1}{2} v^T \nabla^2 g(x_k) \nabla g(x_k)^T v \\
\text{s. t.} & \quad \max \{\delta, -\xi \Delta_k\} \leq v \leq \min \{\bar{\delta}, \xi \Delta_k\}
\end{align*}$$

The role of the vertical subproblem is to find a vertical step $v_k$ within the reduced trust region $\xi \Delta_k$ that minimizes the violation of the equality constraint (5b).

B. Horizontal Subproblem

After solving (7) for the step $v_k$, the full step $d_k$ is obtained by solving the horizontal subproblem:

$$\begin{align*}
\min & \quad \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\
\text{s. t.} & \quad \nabla g(x_k)^T d = \nabla g(x_k)^T v_k \\
& \quad \max \{\delta, -\Delta_k\} \leq d \leq \min \{\bar{\delta}, \Delta_k\}
\end{align*}$$
The role of the horizontal subproblem is to make the same progress that the vertical does towards satisfying the linearized equality constraints while minimizing the objective function of the trust region problem. Note that problem (9) is always feasible, since \( d = v_k \) satisfies (8b) and \( ||v_k|| \leq \xi \Delta_k \).

IV. SEQUENTIAL \( \ell_1 \) QUADRATIC PROGRAMMING

An alternative approach to address the solution of (5) is to build the corresponding exact \( \ell_1 \) penalty function [19] and solve the following modified problem:

\[
\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} (p_i + q_i) \\
\text{s. t.:} & \quad \max \{\delta, -\Delta_k\} \leq d \leq \min \{\delta, \Delta_k\}
\end{align*}
\]

where \( \eta_k > 0 \) is a sufficiently large \( \ell_1 \) penalty parameter.

According to [6], difficulties to choose appropriate values of \( \eta_k \) caused non-smooth penalty methods to fall out of favor in the 1990s and also stimulated the development of filter methods, which do not require the choice of a penalty parameter. In spite of that, in the recent years, new approaches for updating the penalty parameter seem to have overcome these difficulties [20], [21]. As will be further discussed in this work, we propose a simple and effective procedure to update the \( \ell_1 \) penalty parameter based on the values of the problem’s Lagrange multipliers.

The objective function in (9) is non-smooth, so it may be difficult to handle the derivative discontinuities in such a minimization. Fortunately, this is not necessary and, as shown in [6], problem (9) is equivalent to the smooth problem:

\[
\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} (p_i + q_i) \\
\text{s. t.:} & \quad g(x_k) + \nabla g(x_k)^T d = p - q, \quad (p, q) \geq 0 \\
& \quad \max \{\delta, -\Delta_k\} \leq d \leq \min \{\delta, \Delta_k\}
\end{align*}
\]

where \( p \in \mathbb{R}^m \) and \( q \in \mathbb{R}^m \) are nonnegative elastic variables.

Problem (10) is feasible because a step \( d \) can be chosen within the limits (10b) and the elastic variables \( p \) and \( q \) can be chosen in a way that equality constraints (10a) are satisfied.

A. Quadratic Programming Interior-Point Solver

In order to solve QP problem (10), the primal-dual IP method operates on the modified problem:

\[
\begin{align*}
\min & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \mu_k \sum_{i=1}^{m} (p_i + q_i) \\
& \quad - \mu_k \sum_{i=1}^{m} (\ln p_i + \ln q_i) - \mu_k \sum_{i=1}^{n} (\ln r_i + \ln s_i) \\
\text{s. t.:} & \quad g(x_k) + \nabla g(x_k)^T d - p + q = 0, \quad (p, q) > 0 \\
& \quad d - d + r = 0, \quad r > 0 \\
& \quad d - \delta + s = 0, \quad s > 0
\end{align*}
\]

where \( \mu_k > 0 \) is a barrier parameter that goes to zero as iterations progress. The strict positivity \( (r, s, p, q) > 0 \) is handled implicitly through step length control.

If \( d_* \) is a local minimum of (11), then there are vectors of Lagrange multipliers \( \tau_s \in \mathbb{R}^m, \pi_s \in \mathbb{R}^n \) and \( \nu_s \in \mathbb{R}^n \) that satisfy the Karush-Kuhn-Tucker (KKT) first order optimality conditions, conveniently expressed in the form:

\[
\begin{align*}
R \pi - \mu_k e &= 0 \quad (12a) \\
S \nu - \mu_k e &= 0 \quad (12b) \\
P(\eta_k e - \tau) - \mu_k e &= 0 \quad (12c) \\
Q(\eta_k e + \tau) - \mu_k e &= 0 \quad (12d) \\
\frac{d - d}{d} + r &= 0 \quad (12e) \\
\frac{d - \delta}{d} + s &= 0 \quad (12f) \\
g(x_k) + \nabla g(x_k)^T d - p + q &= 0 \quad (12g) \\
\nabla f(x_k) + H_k d + \nabla g(x_k) \tau - \pi + v &= 0 \quad (12h)
\end{align*}
\]

where \( R, S, P \) and \( Q \) are diagonal matrices with \( R_{ii} = r_i \), \( S_{ii} = s_i \), \( P_{ii} = p_i \) and \( Q_{ii} = q_i \); and \( e \) is a vector of all ones of appropriate dimension.

In each iteration of the IP algorithm the following large indefinite linear system is solved:

\[
\begin{align*}
\Pi & 0 0 0 0 0 0 0 0 0 \\
0 & \Upsilon 0 0 0 0 0 0 0 0 \\
0 & 0 0 \Gamma 0 0 0 0 0 0 \\
0 & 0 0 0 \Theta 0 0 0 0 0 \\
I & 0 0 0 0 0 0 0 0 0 \\
0 & I 0 0 0 0 0 0 0 0 \\
0 & 0 0 0 0 0 0 0 0 0 \\
0 & 0 0 0 0 0 0 0 0 0 \\
0 & 0 0 0 0 0 0 0 0 0 \\
\begin{bmatrix}
\Delta r \\
\Delta s \\
\Delta p \\
\Delta q \\
\Delta \pi \\
\Delta \nu \\
\n\n& \begin{bmatrix}
R \pi - \mu_k e \\
S \nu - \mu_k e \\
P(\eta_k e - \tau) - \mu_k e \\
Q(\eta_k e + \tau) - \mu_k e \\
\frac{d - d}{d} + r \\
\frac{d - \delta}{d} + s \\
g(x_k) + \nabla g(x_k)^T d - p + q \\
\nabla f(x_k) + H_k d + \nabla g(x_k) \tau - \pi + v \\
\end{bmatrix}
\end{align*}
\]

where \( \Gamma \) and \( \Theta \) are diagonal matrices with \( \Gamma_{ii} = \eta_k - \tau_i \) and \( \Theta_{ii} = \eta_k + \tau_i \).

The dimensions of the linear systems solved in the vertical and horizontal subproblems of the Byrd-Obomokun technique are \( 5n \) and \( 5n + m \), respectively, while the dimension of the system (13) solved in the \( S \ell_1 QP \) approach is \( 5n + 3m \). Thus, system (13) has \( 2m \) more rows and columns than the system in the horizontal subproblem. However, the \( S \ell_1 QP \) approach solves a single linear system per iteration, while the Byrd-Obomokun technique solves two, one in the vertical and one in the horizontal subproblem. Since the coefficient matrices in both techniques are highly sparse, the time complexity of their factorization involves factors such as ordering and fill-in. For a good sparse matrix algorithm, the time required for sparse matrix operations depends on and should be proportional to the number of arithmetic operations on nonzero elements [22].
Although the coefficient matrix in (13) is slightly larger than analogous matrices in the vertical and horizontal subproblems, it has only 6m additional nonzero entries, corresponding to the diagonal block matrices related to KKT equations (12c), (12d) and (12g), along with the elastic variables. However, it is expected that the computational cost of solving (13) be much lower than the cost of solving two similar linear systems associated with the vertical and horizontal subproblems in the Byrd-Omojokun technique, as demonstrated with the help of the equivalent reduced linear systems presented below.

Note that, alternatively, the search direction can be obtained by solving the equivalent reduced linear system
\[
\begin{bmatrix}
H_k + R^{-1} \Pi + S^{-1} \Gamma & \nabla g(x_k) \\
\nabla g(x_k)^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta d \\
\Delta \tau
\end{bmatrix}
= -\begin{bmatrix}
r_d \\
r_r
\end{bmatrix}
\tag{14}
\]
for \(\Delta d\) and \(\Delta \tau\) first, and then computing
\[
\Delta r = -d + d - r + \Delta d \\
\Delta s = -d + d - s - \Delta d \\
\Delta \pi = -\pi + R^{-1}(\mu_k e - \Pi \Delta r) \\
\Delta v = -v + S^{-1}(\mu_k e - \Upsilon \Delta s) \\
\Delta p = -p + \Gamma^{-1}(\mu_k e + P \Delta \tau) \\
\Delta q = -q + \Theta^{-1}(\mu_k e + Q \Delta \tau)
\]
where \(\Xi = -\Gamma^{-1}P - \Theta^{-1}Q\) is a diagonal matrix, and
\[
r_d = \nabla f(x_k) + H_k d + \nabla g(x_k) \tau - R^{-1}(\mu_k e + \Pi r) \\
\n\nabla g(x_k)^T d + (\Theta^{-1} - \Gamma^{-1}) \mu_k e
\]
with \(r_r = d - d + r\) and \(r_v = d - d + s\).

Similarly, for the horizontal subproblem in Byrd-Omojokun technique (see [18]), the equivalent reduced linear system
\[
\begin{bmatrix}
H_k + R^{-1} \Pi + S^{-1} \Gamma & \nabla g(x_k) \\
\nabla g(x_k)^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta d \\
\Delta \tau
\end{bmatrix}
= -\begin{bmatrix}
r_d \\
r_r
\end{bmatrix}
\tag{23}
\]
is solved for \(\Delta d\) and \(\Delta \tau\) first, and then compute
\[
\Delta r = -d + d - r + \Delta d \\
\Delta s = -d + d - s - \Delta d \\
\Delta \pi = -\pi + R^{-1}(\mu_k e - \Pi \Delta r) \\
\Delta v = -v + S^{-1}(\mu_k e - \Upsilon \Delta s)
\]
where \(\bar{r}_r = \nabla g(x_k)(d - v_k)\) and \(r_d\) is computed from (21).

It is clear now, by comparing the coefficient matrices in the reduced linear systems (14) (in \(\bar{s}\ell_1 QP\) technique) and (23) (in horizontal subproblem of Byrd-Omojokun technique), that the matrix factorization efforts (by far the most time consuming task in an IP algorithm) to solve (14) and (23) are exactly the same, the minor difference in the processing time being the little extra effort to compute \(\Delta p\) and \(\Delta q\) (evaluate equations (19) and (20)), which by involving only simple diagonal matrices and vector operations, is of complexity \(O(n)\) operations only. Clearly, one iteration of the \(\bar{s}\ell_1 QP\) algorithm costs less than one iteration of Byrd-Omojokun technique, when the formulated trust region subproblems are solved by the primal-dual IP algorithm for QP.

\section{B. Merit Function}

Once \(d_k\) has been computed, a merit function is used to decide whether or not \(d_k\) sufficiently decreases the objective function \(f(x)\). The natural choice is the \(\ell_1\) merit function
\[
\psi(x, \eta) = f(x) + \eta \sum_{i=1}^{m} |g_i(x) |
\tag{28}
\]
and its corresponding model
\[
\tilde{\psi}(d, \eta_k) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} |g_i(x_k) + \nabla g_i(x_k)^T d |
\tag{29}
\]
where \(\eta_k > 0\) is the \(\ell_1\) penalty parameter that weighs constraint satisfaction against minimization of the objective.

Given a step \(d_k\), a reduction ratio factor is computed from
\[
\rho_k = \frac{\text{ar}(d_k)}{\text{pr}(d_k)} = \frac{\psi(x_k, \eta_k) - \psi(x_k + d_k, \eta_k)}{\psi(0, \eta_k) - \psi(d_k, \eta_k)}
\tag{30}
\]

where \(\text{ar}(d_k)\) and \(\text{pr}(d_k)\) are called the actual reduction and the predicted reduction in the merit function, respectively. The reduction ratio \(\rho_k\) is used as a measure for accepting or not the step \(d_k\) and for updating the trust region radius. If \(\rho_k\) is close to 1, then there is good agreement between the model \(\tilde{\psi}_k\) and the function \(\psi\) over this step, and thus it is safe to expand the trust region for the next iteration [6]. If \(\rho_k\) is positive but not close to 1, then the trust region is not altered, but if it is close to zero or negative, the trust region is reduced.

\section{C. Trust Region Radius Update}

Based on [18], if \(\rho_k \geq 0.1\) and, consequently, a step \(d_k\) is accepted, the decision of increasing the trust region can be taken as follows:
\[
\Delta_{k+1} = \begin{cases} 
\max \{2 \|d_k\|_{\infty}, \Delta_k\}, & \text{if } \rho_k > 0.9 \\
\max \{1.2 \|d_k\|_{\infty}, \Delta_k\}, & \text{if } 0.3 \leq \rho_k \leq 0.9 \\
\Delta_k, & \text{otherwise;}
\end{cases}
\tag{31}
\]
When a step \(d_k\) is rejected (i.e., \(\rho_k \approx 0\) or negative), the trust region radius is reduced to a fraction as \(\Delta_{k+1} = \gamma \Delta_k\), for \(\gamma \in (0, 1)\). As an alternative to the infinity norm in (31), other norms as the Euclidean can be used [6]. In addition, an upper bound \(\Delta\) on the trust region radius is considered.

\section{D. Initialization and \(\ell_1\) Penalty Parameter Update}

In addition to the usual initialization of primal and dual variables, the initial elastic variables \(p_0\) and \(q_0\) are set based on the complementarity conditions (12c) and (12d), as follows:
\[
p_0 = \frac{\mu_0}{\eta_0 - \tau_0} \tag{32}
\]
\[
q_0^i = \frac{\mu_0}{\eta_0 + \tau_0} \tag{33}
\]
Since the penalty parameter must be strictly positive, its initial value \(\eta_0\) should be different from any initial Lagrange multiplier \(\tau_0^i\) to avoid division by zero. According to [6], the
\( \ell_1 \) merit function (28) has a local minimizer if there is an \( \eta \) greater than the largest optimal Lagrange multiplier, i.e.,

\[
\eta = \max \{ \| \tau \|, \pi, u \} \tag{34}
\]

In this paper, the penalty parameter is chosen adaptively in terms of the current value \( \eta_{k} \) and the infinity norm of the Lagrange multipliers associated with the equality constraints. In the first trust region iteration, a user defined value \( \eta_0 \) is set and in the subsequent outer iterations it is updated as

\[
\eta_{k+1} = \max \{ \eta_k, 3\| \tau_k \|_\infty \} \tag{35}
\]

Additionally, the penalty parameter is also allowed to vary within the same trust region iteration, by smaller amounts, as

\[
\eta_{kj} = \max \{ \eta_{k-1}, \| \tau_{k-1} \|_\infty \} \tag{36}
\]

where the subscript \( k \) in \( \eta_{kj} \) denotes the number of the outer iteration and the subscript \( j \) denotes the number of the inner iteration. Since (10) is solved by IP methods, the current estimate \( \tau_k \) for \( \lambda_i \) is readily available. The proposed \( S\ell_1 \)QP approach is summarized in Algorithm 1.

1. Set \( k = 0 \), choose \( \Delta_0 \), \( \eta_0 \) and \( \xi \).
2. Solve (10) for \( d_k \) by using the primal-dual IP method and update the \( \ell_1 \) penalty parameter by (35) and (36).
3. Compute the reduction ratio \( \rho_k \) by (30).
4. If \( \rho_k \geq 0.1 \), then update \( x_{k+1} = x_k + d_k \) and choose \( \Delta_{k+1} \geq \Delta_k \) by (31). Otherwise, decrease the trust region radius \( \Delta_{k+1} = \gamma \Delta_k \) and set \( x_{k+1} = x_k \).
5. If \( x_{k+1} \) satisfies the convergence criteria of the NLP problem (1) then END. Otherwise, set \( k \leftarrow k + 1 \) and return to step 2.

Algorithm 1: Main steps of the \( S\ell_1 \)QP algorithm.

V. NUMERICAL EXPERIMENTS

The proposed trust region OPF solver is tested on the IEEE test systems of up to 300-bus. The OPF problem solved is the active transmission losses minimization problem is defined as follows:

\[
\begin{align*}
\min \quad & P_{\text{Losses}}(V, \theta, t) \\
\text{s. t.:} \quad & P_i(V, \theta, t) + P_{Di} - P_{Gi} = 0, \forall i \in \mathcal{N} \\
& Q_i(V, \theta, t) + Q_{Di} - Q_{Gi} = 0, \forall i \in \mathcal{G} \\
& Q_i(V, \theta, t) + Q_{Di} - Q_{Gi} = 0, \forall i \in \mathcal{F} \\
& V_i^\min \leq V_i \leq V_i^\max, \forall i \in \mathcal{N} \\
& P_i^\min \leq P_{Gi} \leq P_i^\max, \forall i \in \mathcal{G} \\
& Q_i^\min \leq Q_{Gi} \leq Q_i^\max, \forall i \in \mathcal{G} \\
& \rho_{ij}^\min \leq \rho_{ij} \leq \rho_{ij}^\max, \forall (i, j) \subseteq \mathcal{T} \\
& b_{ij}^\min \leq b_{ij} \leq b_{ij}^\max, \forall i \in \mathcal{C}
\end{align*}
\tag{37}
\]

where

\[
\begin{align*}
P_i(V, \theta, t) &= V_i \sum_{j \in \mathcal{N}} V_j \left( G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right) \tag{38} \\
Q_i(V, \theta, t) &= V_i \sum_{j \in \mathcal{N}} V_j \left( G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right) \tag{39}
\end{align*}
\]

\( V_i \) and \( \theta_i \) are the voltage magnitude and phase angle (\( \theta_{ij} = \theta_i - \theta_j \)) at the bus \( i \), \( P_{Gi} \) and \( P_{Di} \) are the active generation and demand, \( Q_{Gi} \) and \( Q_{Di} \) are the reactive generation and demand, and \( b_{ij} \) is the shunt susceptance, respectively, all at the bus \( i \). \( \mathcal{N} \) is the index set of all buses, \( \mathcal{G} \) is the set of all generation buses, \( \mathcal{C} \) is the set of buses with variable shunt capacitors (or reactors) and \( \mathcal{T} \) is the set of LTC transformers.

The implementations were performed in MATLAB\textsuperscript{®}. The used trust region and primal-dual IP parameters are: \( \Delta_0 = 2, \Delta = 5, \eta_0 = 2, \xi = 0.8, \mu_0 = 0.1, \gamma = 0.9995, \sigma = 0.2 \) and \( \epsilon = 10^{-4} \). The test systems and the active losses minimization problems dimensions are shown in Table I. The active and reactive systems loadings, and the initial active losses are presented in Table II.

| System  | \( |\mathcal{N}| \) | \( |\mathcal{G}| \) | \( |\mathcal{C}| \) | \( |\mathcal{T}| \) | \( n \) | \( m \) |
|---------|----------------|----------------|----------------|----------------|-------|------|
| IEEE-30 | 30             | 6              | 5              | 4              | 75    | 60   |
| IEEE-57 | 57             | 7              | 5              | 17             | 143   | 114  |
| IEEE-118| 118            | 54             | 12             | 9              | 311   | 236  |
| IEEE-300| 300            | 69             | 23             | 35             | 727   | 600  |

<table>
<thead>
<tr>
<th>System</th>
<th>( P_{Di} ) (MW)</th>
<th>( Q_{Di} ) (MVAr)</th>
<th>Losses (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>283.40</td>
<td>126.20</td>
<td>17.64</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>1250.80</td>
<td>336.40</td>
<td>27.86</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>3668.00</td>
<td>1438.00</td>
<td>132.86</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>23247.00</td>
<td>7788.00</td>
<td>408.31</td>
</tr>
</tbody>
</table>

As previously mentioned, locally convergent algorithms are expected to fail in some instances when submitted to a wide range of initial points. Thus, to evaluate the robustness of the proposed OPF solver, simulations were performed using four different types of initialization: (I) initial power flow solution, (II) flat start (i.e., \( V_i^0 = 1 \) and \( \theta_i^0 = 0 \)), (III) middle point of limits (e.g., \( V_i^0 = (V_i^\min + V_i^\max)/2 \)), and (IV) random points within limits (one hundred random points). The random points were generated using the MATLAB\textsuperscript{®}'s built-in function \texttt{rand()} using uniform distribution. Table III displays the number of IP iterations required to converge for initializations (I), (II) and (III), as well as the minimum active losses and the percentage reduction when the pure primal-dual IP algorithm is directly applied to solve the NLP problem (1).

Table IV shows the number of outer (trust region) iterations required by the Byrd-Omojokun and \( S\ell_1 \)QP algorithms to converge, using initializations (I), (II) and (III). The \( S\ell_1 \)QP algorithm requires nearly the same number of outer iterations as the Byrd-Omojokun algorithm. The numerical factorization
of the coefficient matrix in (13) is the most time consuming task in the proposed trust region algorithm. Considering the initialization (I) and the results for the IEEE 30-bus shown in Table IV, the overall elapsed CPU times to solve the Newton systems associated with the vertical and horizontal subproblems are 0.0690s and 0.1100s, respectively. On the other hand, the total elapsed time to solve the linear systems (13) is about 0.1580s, which represents a reduction of about 11.73%. Similarly, percentages equals to 4.37%, 21.52% and 3.04% were observed for the IEEE 57-, 118- and 300-bus test systems, respectively.

### Table IV
Number of outer iterations required by Byrd-Omojukun and Sf\_1 QP algorithms with initializations (I), (II) and (III).

<table>
<thead>
<tr>
<th>System</th>
<th>(I)</th>
<th>(II)</th>
<th>(III)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>10</td>
<td>11</td>
<td>16</td>
</tr>
</tbody>
</table>

Given the high number of random initial points used in the simulations (100 points), the practical performance of the IP algorithm, Byrd-Omojukun algorithm, and Sf\_1 QP algorithm is summarized in Tables V, VI and VII, respectively, using some important statistical data, like the number of converged cases (CC), the mode, the mean and the standard deviation \(\sigma\) of iteration counts, along with the minimum number \(k_{\text{min}}\) and maximum number \(k_{\text{max}}\) of outer iterations. As expected, concerning convergence robustness, the locally convergent IP algorithm directly applied to (I) is less robust than the Byrd-Omojukun and \(Sf\_1\) QP algorithms, since the numbers of CC when solving the IEEE 300-bus system is lower. Statistics for the Byrd-Omojukun and \(Sf\_1\) QP algorithms are quite similar, indicating that the performances of the algorithms, concerning convergence robustness and number of outer iterations, are quite similar. Thus, for the same number of outer iterations, the best performance in terms of processing time becomes related to the best performance per outer iteration. Clearly, the cost per outer iteration in the \(Sf\_1\) QP algorithms is lower than in the Byrd-Omojukun algorithm, as it solves a single QP problem of nearly the same size as one of the two QP subproblems (horizontal subproblem) solved in the Byrd-Omojukun algorithm. Thus, for the same number of outer iterations, total CPU time is lower in the \(Sf\_1\) QP algorithm than in the Byrd-Omojukun algorithm.

The three optimization algorithms were also submitted to tests in which the loads of the systems were increased in a way to obtain feasible but highly nonlinear cases. Tables VIII and IX present the numerical results. The number of outer iterations required to convergence has not significantly changed from those shown in Table IV, which indicates that the proposed \(Sf\_1\) QP algorithm is not very sensitive to the complexity of the OPF problem.

### Table V
Statistics for the IP algorithm using initialization (IV).

<table>
<thead>
<tr>
<th>System</th>
<th>CC</th>
<th>Mode</th>
<th>Mean</th>
<th>(\sigma)</th>
<th>(k_{\text{min}})</th>
<th>(k_{\text{max}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>100</td>
<td>14</td>
<td>13.56</td>
<td>0.50</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>100</td>
<td>13</td>
<td>13.07</td>
<td>0.26</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>100</td>
<td>15</td>
<td>14.97</td>
<td>0.44</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>80</td>
<td>20</td>
<td>21.27</td>
<td>1.60</td>
<td>13</td>
<td>22</td>
</tr>
</tbody>
</table>

### Table VI
Statistics for Byrd-Omojukun method with initialization (IV).

<table>
<thead>
<tr>
<th>System</th>
<th>CC</th>
<th>Mode</th>
<th>Mean</th>
<th>(\sigma)</th>
<th>(k_{\text{min}})</th>
<th>(k_{\text{max}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>100</td>
<td>4</td>
<td>5.37</td>
<td>2.50</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>100</td>
<td>6</td>
<td>6.56</td>
<td>1.04</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>100</td>
<td>4</td>
<td>4.00</td>
<td>0.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>93</td>
<td>15</td>
<td>14.76</td>
<td>2.11</td>
<td>11</td>
<td>24</td>
</tr>
</tbody>
</table>

### Table VII
Statistics for \(Sf\_1\) QP algorithm using initialization (IV).

<table>
<thead>
<tr>
<th>System</th>
<th>CC</th>
<th>Mode</th>
<th>Mean</th>
<th>(\sigma)</th>
<th>(k_{\text{min}})</th>
<th>(k_{\text{max}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>100</td>
<td>4</td>
<td>4.90</td>
<td>1.64</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>100</td>
<td>6</td>
<td>6.88</td>
<td>1.44</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>100</td>
<td>4</td>
<td>4.00</td>
<td>0.00</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>96</td>
<td>17</td>
<td>17.28</td>
<td>1.60</td>
<td>13</td>
<td>22</td>
</tr>
</tbody>
</table>

### Table VIII
Number of iterations for IP method, for increased load cases and initializations (I), (II) and (III).

<table>
<thead>
<tr>
<th>System</th>
<th>(I)</th>
<th>(II)</th>
<th>(III)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>11</td>
<td>13</td>
<td>27.36</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>14</td>
<td>13</td>
<td>60.43</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>17</td>
<td>18</td>
<td>221.77</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>17</td>
<td>21</td>
<td>444.93</td>
</tr>
</tbody>
</table>

### Table IX
Number of outer iterations for Byrd-Omojukun and \(Sf\_1\) QP algorithms, for increased load cases and initializations (I), (II) and (III).

<table>
<thead>
<tr>
<th>System</th>
<th>BO (Sf_1) QP</th>
<th>BO (Sf_1) QP</th>
<th>BO (Sf_1) QP</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE-30</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>IEEE-57</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>IEEE-118</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>IEEE-300</td>
<td>10</td>
<td>14</td>
<td>13</td>
</tr>
</tbody>
</table>

Table X details the convergence process of the \(Sf\_1\) QP approach for the IEEE 57-bus system with increased loads and power flow initialization. This case clearly illustrates the update process of the \(\ell_1\) penalty parameter. The initial value set to \(\eta_0 = 2\) slightly increases within the first outer iteration by using (36) and at the beginning of the second iteration using (35). Furthermore, from the beginning of the second outer iteration until the process converge, the \(\ell_1\) penalty parameter remained constant and equal to \(\eta_0 = 6.0424\).
TABLE X

<table>
<thead>
<tr>
<th>Iter.</th>
<th>Infeasibility Residuals (I)</th>
<th>η_k</th>
<th>µ_k</th>
<th>Δ_k</th>
<th>f(x_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out.</td>
<td>Primal</td>
<td>Dual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>4.28×10^{-4}</td>
<td>3.31×10^{-5}</td>
<td>2.0000</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>9.66×10^{-5}</td>
<td>2.09×10^{-2}</td>
<td>2.0000</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2.25×10^{-4}</td>
<td>3.90×10^{-6}</td>
<td>2.0120</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2.47×10^{-2}</td>
<td>9.49×10^{-7}</td>
<td>2.0132</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2.29×10^{-3}</td>
<td>4.79×10^{-8}</td>
<td>2.0141</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>6.66×10^{-9}</td>
<td>1.38×10^{-9}</td>
<td>2.0141</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1.72×10^{-1}</td>
<td>3.11×10^{-7}</td>
<td>2.0141</td>
<td>0.94</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>5.27×10^{-2}</td>
<td>1.54×10^{-2}</td>
<td>6.0424</td>
<td>0.83</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>2.08×10^{-3}</td>
<td>5.12×10^{-4}</td>
<td>6.0424</td>
<td>0.97</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>8.06×10^{-5}</td>
<td>1.68×10^{-5}</td>
<td>6.0424</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Figure 1 illustrates the variation of the ℓ₁ penalty parameter η_k along with the primal and dual infeasibilities for the IEEE 118-bus with increased loads and power flow initialization. Similarly to the results obtained for the IEEE 57-bus, the ℓ₁ penalty parameter increases within the first two outer iterations and then remains constant until the process converges. Numerical experiments with the other test systems indicate analogous patterns of convergence.

VI. CONCLUSIONS

This paper has presented a StQP trust region algorithm applied to the solution of the minimum active transmission losses problem. The proposed OPF solver was developed based on a general OPF problem form. Therefore, its application to other OPF instances, such as the minimization of generation costs, is straightforward. The proposed OPF algorithm has demonstrated to be competitive in processing time with the Byrd-Omojokun OPF solver proposed by Sousa et al. [18]. In each iteration, instead of dividing the trust region subproblem into two QP subproblems, as the Byrd-Omojokun algorithm does, the StQP technique solves just one slightly larger QP problem, which invariably takes less time. Additionally, the paper also describes a simple and effective procedure to update the ℓ₁ penalty parameter based on the problem’s Lagrange multipliers. The computational experiments on the used class of OPF problems indicate that this procedure can properly estimate a value for the ℓ₁ penalty parameter during the iterative process.

REFERENCES