VICENTE RIBEIRO SIMONI

PERFORMANCE IMPROVEMENT OF A TRUST REGION INTERIOR POINT METHOD TO SOLVE NONLINEAR OPTIMAL POWER FLOWS

Recife - Pernambuco - Brazil

March 2014

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PERFORMANCE IMPROVEMENT OF A TRUST REGION INTERIOR POINT METHOD TO SOLVE NONLINEAR OPTIMAL POWER FLOWS

by

VICENTE RIBEIRO SIMONI

Thesis presented to the Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

SUPERVISOR: PROF. GERALDO LEITE TORRES, PhD

Recife, March 2014.

Abstract of Thesis Presented to the
Universidade Federal de Pernambuco (UFPE) as Partial Fulfillment of the
Requirements for the Degree of Doctor in Electrical Engineering.

PERFORMANCE IMPROVEMENT OF A TRUST REGION INTERIOR POINT METHOD TO SOLVE NONLINEAR OPTIMAL POWER FLOWS

Vicente Ribeiro Simoni

March 2014

Supervisor: Prof. Geraldo Leite Torres, PhD
Area of Concentration: Energy Processing
Keywords: Optimal Power Flow, Interior Point Methods, Global Convergence, Trust Region Methods, Sequential ℓ_1 Quadratic Programming.
Number of Pages: 105

Large-scale nonlinear Optimal Power Flow (OPF) problems have been solved lately by primal-dual interior point (IP) methods and, despite 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 a need for globally convergent OPF solvers. Trust region schemes have been used to enforce convergence, but they are by nature computationally expensive. The main goal of the Thesis is to reduce the computational effort of the Byrd-Omojokun (BO) trust region OPF algorithm proposed by Sousa et al.. The BO technique handles possible inconsistencies in the trust region subproblems by solving a sequence of quadratic programming (SQP) problems, known as the *vertical* and *horizontal* subproblems. The idea exploited to reduce the computation time is to avoid the solution of the vertical subproblem after the iteration its optimal objective value becomes zero, since this means that the constraints of the trust region subproblem are consistent. The proposed algorithm is called *Modified Byrd-Omojokun* (MBO) and, by directly solving the trust region subproblems, it is expected to reduce the computation time to nearly half from that iteration on. Additionally, an alternative procedure based on *sequential* ℓ_1 *quadratic programming* (Sℓ_1QP) is devised. Rather than solving two quadratic programming (QP) problems per iteration as in the Byrd-Omojokun technique, the Sℓ_1QP approach solves a single, but slightly larger, QP problem. The Thesis makes a contribution towards the solution of nonlinear OPF problems in real time by addressing the discussed issues. The proposed methodologies are tested in the IEEE test systems of up to 300-bus and two actual subtransmission systems.

Resumo da Tese apresentada à
Universidade Federal de Pernambuco (UFPE) como parte dos requisitos
necessários para obtenção do grau de Doutor em Engenharia Elétrica.

PERFORMANCE IMPROVEMENT OF A TRUST REGION INTERIOR POINT METHOD TO SOLVE NONLINEAR OPTIMAL POWER FLOWS

Vicente Ribeiro Simoni

Março de 2014

Orientador: Prof. Geraldo Leite Torres, PhD
Área de Concentração: Processamento da Energia
Keywords: Fluxo de Potência Ótimo, Métodos de Pontos Interiores, Convergência Global, Métodos de Região de Confiança, Programação Sequencial ℓ_1 Quadrática.
Número de Páginas: 105

Problemas não-lineares de Fluxo de Potência Ótimo (FPO) de grande dimensão têm sido resolvidos por métodos primais-duais de pontos interiores (PI) e, apesar do sucesso, existem várias situações em que programas baseados nesses métodos podem falhar em encontrar a solução. Por outro lado, com sistemas de potência operando extremamente carregados, existe a necessidade de programas de FPO globalmente convergentes. Esquemas de região de confiança têm sido usados para assegurar convergência, mas eles são por natureza computacionalmente onerosos. O principal objetivo da Tese é reduzir o esforço computacional do algoritmo de Byrd-Omojokun (BO) de região de confiança proposto por Sousa et al.. A técnica de BO trata possíveis inconsistências nos subproblemas de região de confiança resolvendo uma sequência de problemas de programação quadrática (PQ) conhecidos como subproblemas *vertical* e *horizontal*. A ideia explorada para reduzir o tempo computacional é evitar a solução do subproblema vertical após a iteração em que o valor ótimo do seu objetivo torna-se zero, uma vez que isso indica que as restrições do subproblema de região de confiança são consistentes. O algoritmo proposto é chamado Byrd-Omojokun modificado e, por resolver diretamente os subproblemas de região de confiança, espera-se que ele reduza o tempo computacional para cerca da metade a partir daquela iteração. Adicionalmente, um procedimento alternativo baseado em programação sequencial ℓ_1 quadrática (PSℓ_1Q) é desenvolvido. Em vez de resolver dois problemas de PQ por iteração como ocorre na técnica de BO, o método de PSℓ_1Q resolve um único, porém ligeiramente maior, problema de PQ. A Tese contribui para a solução de problemas de FPO não-linear desenvolvendo os aspectos discutidos. As metodologias propostas são testadas nos sistemas do IEEE até o 300-bus e dois sistemas reais de subtransmissão.

Contents

List of Acronyms vii

List of Symbols ix

List of Figures xi

List of Tables xii

List of Algorithms xv

1 Introduction 1

1.1 Optimal Power Flow 2

1.1.1 Active Transmission Losses Minimization 3

1.2 Solution Methods 4

1.3 Objectives of the Thesis 8

1.4 Contents of the Thesis 9

2 Penalty Methods 10

2.1 The Quadratic Penalty Method 11
2.2 The ℓ_1 Penalty Method 12
2.3 The Augmented Lagrangian Method 16
2.4 Final Remarks 17

3 Interior Point Methods for Nonlinear Programming 18
3.1 The Primal-Dual Interior Point Method 19
3.1.1 Computation of the Search Directions 21
3.1.2 Update of Variables 22
3.1.3 Reducing the Barrier Parameter 23
3.1.4 Convergence Criteria 24
3.2 The Predictor-Corrector Interior Point Method 24
3.2.1 The Predictor Step 26
3.2.2 The Corrector Step 27
3.3 Infeasibility Detection and Handling 28
3.3.1 Primal-Dual Logarithmic Indicators 30
3.4 Final Remarks 31

4 Trust Region Methods 32
4.1 Nonlinear Least Squares Problems 33
4.2 Trust Region Methods for Unconstrained Optimization 35
4.2.1 The Cauchy Point 36
4.2.2 The Dogleg Method 38
4.3 Trust Region Methods for Constrained Optimization 40
4.3.1 The Byrd-Omojokun Method 42
4.3.1.1 Vertical Subproblem 42
4.3.1.2 Horizontal Subproblem 42
4.3.1.3 Merit Function 43
4.3.1.4 Trust Region Radius Update 44

4.3.2 Sequential ℓ_1 Quadratic Programming 44
4.4 Final Remarks 46

5 Trust Region Methods for Improved Performance 47
5.1 Modified Byrd-Omojokun Approach 48
5.1.1 Direct Solution of the Trust Region Problem 49
5.1.1.1 Tolerance Criteria on Inner Iterations and Early Stop 52
5.2 The Sℓ_1QP Method 53
5.2.1 Solving the Reduced System 55
5.2.2 Decreasing the Barrier Parameter 57
5.2.3 Initialization and ℓ_1 Penalty Parameter Update 58
5.3 Final Remarks 59

6 Numerical Experiments 60
6.1 Test Systems and General Parameters 61
6.2 Primal-Dual IP Algorithms 61
6.3 Trust Region IP Algorithms 63
6.3.1 Trust Region Problems Solved by the PD Algorithm 64
6.3.2 Trust Region Problems Solved by the PC Algorithm 67
6.3.3 Performance For Different Sets of Parameters 70
6.3.4 Less Restrictive Tolerance Criteria on Inner Iterations 71
6.3.5 Monitoring by Primal-Dual Logarithmic Indicators 74
6.4 Final Remarks 76

7 Conclusions 78
7.1 Summary and Contributions 79
7.2 Perspectives for Future Research 80

References 82

Appendix **89**

A Parametric Grid **90**

List of Acronyms

BO	Byrd-Omojokun Method.
IP	Interior Point Method.
KKT	Karush-Kuhn-Tucker.
LP	Linear Programming.
LTC	Load Tap Changer.
NLP	Nonlinear Programming.
OPF	Optimal Power Flow.
PD	Pure Primal-Dual Interior Point Method.
PC	Primal-Dual Predictor-Corrector Interior Point Method.
QP	Quadratic Programming.
MBO	Modified Byrd-Omojokun Method.
SQP	Sequential Quadratic Programming.
Sℓ_1QP	Sequential ℓ_1 Quadratic Programming.

List of Symbols

$\mathcal{N}$	Set of all buses of an electrical system.
$\mathcal{G}$	Set of all generation buses.
$\mathcal{T}$	Set of pairs of buses of LTC transformers.
$\mathcal{C}$	Set of all buses with variable shunt capacitors or reactors.
$\mathcal{F}$	Set of all buses with fixed reactive sources.
P_i^G	Active power generation at bus i.
P_i^D	Active power load at bus i.
P_i	Active power injection at bus i.
Q_i^G	Reactive power generation at bus i.
Q_i^D	Reactive power load at bus i.
Q_i	Reactive power injection at bus i.
P_{Losses}	Total active power losses in the transmission system.
x	Vector of decision variables of an optimization problem.
$f(x)$	Nonlinear objective function.
$g(x)$	Vector of nonlinear equality constraints.
$\underline{x}$	Vector of lower bound constraints.

$\overline{x}$	Vector of upper bound constraints.
μ	Barrier parameter.
η	Penalty parameter.

List of Figures

2.1 The exact ℓ_1 penalty function for problem (2.4) (Adapted from [1, 2]). 13

2.2 The exact ℓ_1 penalty function for problem (2.6) with $\eta = -1$ (Adapted from [1]). 14

2.3 The exact ℓ_1 penalty function for problem (2.6) with $\eta = 1$ (Adapted from [1]). 14

3.1 Primal-Dual logarithmic indicators of infeasibility. 30

4.1 Concave function $m_k(\tau d_k^s)$. 38

4.2 Convex function $m_k(\tau d_k^s)$. 38

4.3 Exact trajectory and dogleg path (Adapted from [1]). 39

5.1 Taylor polynomial approximation of $c(x)$ around x_k. 49

6.1 Variation of the ℓ_1 penalty parameter along with the primal and dual infeasibilities for IEEE-118 with increased loads. 68

6.2 Boxplot for initial trust region radius Δ_0 and the number of CC, for BO method and IEEE 300-bus. 72

6.3 Boxplot for initial trust region radius Δ_0 and the number of CC, for MBO method and IEEE 300-bus. 72

6.4 Logarithmic indicators profile for a feasible trust region subproblem. 75

6.5 Logarithmic indicators profile for an infeasible trust region subproblem. 76

List of Tables

6.1 Test systems and active losses minimization problems dimensions. . . 61

6.2 Active and reactive loads, initial and minimum active losses and percentage reduction. 61

6.3 Number of iterations for IP methods with three different initializations. 62

6.4 Descriptive statistics for the PD method using initialization (IV). . . . 62

6.5 Descriptive statistics for the PC method using initialization (IV). . . . 62

6.6 Number of iterations for IP methods for increased load cases and initializations (I), (II) and (III). 63

6.7 Number of iterations for initializations (I), (II) and (III) with trust region problems solved by the PD method. 64

6.8 Convergence process of the MBO method for the IEEE-57 test system with (II). 65

6.9 Descriptive statistics for the BO method with PD algorithm using initialization (IV). 65

6.10 Descriptive statistics for the MBO method with PD algorithm using initialization (IV). 66

6.11 Descriptive statistics for the $S\ell_1QP$ method with PD algorithm using initialization (IV). 66

6.12 Number of outer iterations for the BO, the MBO and the Sℓ_1QP methods for increased load cases and initializations (I), (II) and (III). . . . 66

6.13 Convergence process of the Sℓ_1QP method for IEEE-57 system with increased loads and initialization (I). 67

6.14 Number of iterations for initializations (I), (II) and (III) with trust region problems solved by the PC method. 68

6.15 Failed convergence process for the BO method and IEEE 300-bus system with PC algorithm and initialization (II). 69

6.16 Descriptive statistics for the BO method with PC algorithm using initialization (IV). 70

6.17 Descriptive statistics for the MBO method with PC algorithm using initialization (IV). 70

6.18 Descriptive statistics for the $S\ell_1QP$ method with PC algorithm using initialization (IV). 70

6.19 Less restrictive tolerance criterion on inner iterations for BO method and initialization (I). 73

6.20 Less restrictive tolerance criterion on inner iterations for BO method and initialization (II). 73

6.21 Less restrictive tolerance criterion on inner iterations for BO method and initialization (III). 74

A.1 Parametric grid for the IEEE 30-bus solved by the BO Method. 91

A.2 Parametric grid for the IEEE 57-bus solved by the BO Method. 92

A.3 Parametric grid for the IEEE 118-bus solved by the BO Method. . . . 93

A.4 Parametric grid for the IEEE 300-bus solved by the BO Method. . . . 94

A.5 Parametric grid for the REAL-A solved by the BO Method. 95

A.6 Parametric grid for the IEEE REAL-R solved by the BO Method. . . . 96

A.7 Parametric grid for the IEEE 30-bus solved by the MBO Method. . . . 97

A.8 Parametric grid for the IEEE 57-bus solved by the MBO Method. . . . 98

A.9 Parametric grid for the IEEE 118-bus solved by the MBO Method. . . 99
A.10 Parametric grid for the IEEE 300-bus solved by the MBO Method. . . 100
A.11 Parametric grid for the REAL-A solved by the MBO Method. 101
A.12 Parametric grid for the REAL-R solved by the MBO Method. 102
A.13 Parametric grid for the IEEE 30-bus solved by the Sℓ_1QP Method. . . 103
A.14 Parametric grid for the IEEE 57-bus solved by the Sℓ_1QP Method. . . 103
A.15 Parametric grid for the IEEE 118-bus solved by the Sℓ_1QP Method. . . 104
A.16 Parametric grid for the IEEE 300-bus solved by the Sℓ_1QP Method. . . 104
A.17 Parametric grid for the REAL-A solved by the Sℓ_1QP Method. 105
A.18 Parametric grid for the REAL-R solved by the Sℓ_1QP Method. 105

List of Algorithms

3.1 Primal-dual IP method to solve (1.1). 21
3.2 Primal-dual predictor-corrector IP method to solve (1.1). 28
5.1 Modified Byrd-Omojokun method to solve (1.1). 52
5.2 Proposed Sℓ_1QP method to solve (1.1). 59

Chapter 1

Introduction

In the past years, the time required to execute some power systems analysis during the operation and planning studies has been significantly reduced. A great number of computational programs have been designed to perform network analysis and to guide new investments. Nowadays, power grid operators can count on advanced energy management systems for monitoring and controlling almost all instances from generation to electric power distribution, providing data for online security analysis. Additionally, with the advent of distributed generation systems and advanced metering infrastructure, massive real-time power systems calculations are becoming very usual. Thus, computational tools for power systems analysis must be prepared to face future challenges, providing a fast, secure and reliable solution for electrical engineering problems.

1.1 Optimal Power Flow

Among the computational tools that are generally included in supervisory systems, the *optimal power flow* (OPF) [3–9] can be considered one of the most important and prominent. According to [10], OPF can be defined as the determination of the best operating point (that is minimum fuel cost, maximum power transferred between areas, minimum load shedding, among others) satisfying security constraints.

Several OPF problems have been studied since its first mathematical formulation by Carpentier [11] in the 1960's. Research topics mainly concentrate on new formulations, which attempt to address some special characteristics such as the treatment of discrete variables [12–22], handling of infeasible OPF problems [23], reduction of the number of control actions [24–27], etc. In addition, a great effort has been made in order to improve the solution techniques [28] and to model power systems equipments [29].

OPF problems are inherently *nonconvex nonlinear programming* (NLP) problems, which can be formulated as follows:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x} \leq x \leq \overline{x} \end{aligned} \tag{1.1}$$

where,

- $x \in \mathbb{R}^n$ is a vector of decision variables, including state and control variables such as voltages at generation buses, tap of power transformers and reactive shunt compensation;
- $f : \mathbb{R}^n \mapsto \mathbb{R}$ is the objective function representing a planning or operational goal, such as active transmission losses minimization;
- $g : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a vector of nonlinear equality constraints including the power balance equations;
- $\underline{x} \in \mathbb{R}^n$ and $\overline{x} \in \mathbb{R}^n$ are vectors of simple bounds, which represent equipment or system's operational limits.

1.1.1 Active Transmission Losses Minimization

A basic formulation of the *active transmission losses minimization* problem generally includes the active and reactive power balance equations, the operational limits related to voltage levels in all buses, the active and reactive power limits of generators and the physical limits of transformers with load tap changer (LTC) and of shunt compensation devices.

With the exception of the *swing* bus, the active power generation in all buses are assumed constant and equal to the values defined by the economic dispatch problem [11]. The control variables, i.e., the variables that can be controlled by the electrical grid operators, are terminal voltages of generators, shunt susceptancies of switchable capacitors and reactors and taps of LTC transformers. The state variables are the voltage magnitudes at load buses, the phases of nodal voltages and reactive power of generators.

In a compact form, the active transmission losses minimization problem can be expressed as follows:

$$
\begin{aligned}
\min \quad & P_{Losses}(V,\theta,t) \\
\text{subject to} \quad & P_i(V,\theta,t) + P_{D_i} - P_{G_i} = 0, && \forall i \in \mathcal{N} \\
& Q_i(V,\theta,t) + Q_{D_i} - Q_{G_i} = 0, && \forall i \in \mathcal{G} \\
& Q_i(V,\theta,t) + Q_{D_i} - Q_{G_i} = 0, && \forall i \in \mathcal{F} \\
& Q_i(V,\theta,t) + Q_{D_i} - Q_{G_i} - b_i^{sh} V_i^2 = 0, && \forall i \in \mathcal{C} \\
& V_i^{\min} \leq V_i \leq V_i^{\max}, && \forall i \in \mathcal{N} \\
& P_i^{\min} \leq P_{G_i} \leq P_i^{\max}, && \forall i \in \mathcal{G} \\
& Q_i^{\min} \leq Q_{G_i} \leq Q_i^{\max}, && \forall i \in \mathcal{G} \\
& t_{ij}^{\min} \leq t_{ij} \leq t_{ij}^{\max}, && \{(i,j)\} \subseteq \mathcal{T} \\
& b_i^{\min} \leq b_i^{sh} \leq b_i^{\max}, && \forall i \in \mathcal{C}
\end{aligned}
\tag{1.2}
$$

where $\mathcal{N}$ is the set of all buses, $\mathcal{G}$ is the set of all generation buses, $\mathcal{F}$ is the set of all buses with fixed reactive sources, $\mathcal{C}$ is the set of buses with variable shunt capacitors (or reactors), $\mathcal{T}$ is the set of LTC transformers. Additionally, P_{G_i} and P_{D_i} are the active generation and load, Q_{G_i} and Q_{D_i} are the reactive generation and demand, and b_i^{sh} is the shunt susceptance at bus i, respectively.

As described in [30], the active and reactive power injections $P_i(V,\theta,t)$ and

$Q_i(V,\theta,t)$ are defined as follows:

$$P_i(V,\theta,t) = V_i^2 G_{ii} + V_i \sum_{k\in\mathcal{N}_i} V_k \left(G_{ik}\cos\theta_{ik} + B_{ik}\,\mathrm{sen}\,\theta_{ik}\right) \tag{1.3}$$

$$Q_i(V,\theta,t) = -V_i^2 B_{ii} + V_i \sum_{k\in\mathcal{N}_i} V_k \left(G_{ik}\,\mathrm{sen}\,\theta_{ik} - B_{ik}\cos\theta_{ik}\right) \tag{1.4}$$

where V_i and θ_i are the voltage magnitude and phase angle ($\theta_{ij} = \theta_i - \theta_j$), and $\mathcal{N}_i$ is the set of all buses connected to the bus i.

The objective function of (1.2) is the total active loss in the transmission system $P_{Losses}(V,\theta,t)$, which is calculated as follows:

$$P_{Losses} = \sum_{\{(i,j)\}\in\mathcal{B}} g_{ij}\left(V_i^2 + V_j^2 - 2V_iV_j\cos\theta_{ij}\right) \tag{1.5}$$

where $\mathcal{B} = \{(i,j) \mid i \in \mathcal{N},\ j \in \mathcal{N}_i \text{ and } j > i\}$ is the set of terminal buses of all system's branches (transmission lines and transformers, among others).

The active losses minimization problem is used throughout this work mainly because such an implementation is readily available in the actual stage of the designed OPF code. Besides, this minimization problem can provide, without loss of generality, all the required tools to extensively test our developed trust region IP methods. Additionally, the main set of nonlinear network equations is already present in this problem. For further details regarding the modeling of other OPF problems the reader can consult [31, 32].

1.2 Solution Methods

Lately, large-scale OPF problems have been successfully solved by primal-dual interior point (IP) methods [33], mainly for the *primal-dual predictor-corrector* [34] and *multiple centrality corrections* [35, 36] variants. However, there are many situations in which OPF programs can fail to find a solution [37]. In fact, locally convergent algorithms may fail to converge when the initial guess is not near to a solution. Fortunately, globally convergent techniques can be used to obtain convergence from remote starting points.

Line search and *trust region* methods are two important descent schemes for guaranteeing global convergence [1]. Each iteration of a line search method com-

putes 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 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 then choose the step to be the approximate minimizer of the model in this region.

Trust region methods [38, 39] have been used to provide global convergence to a great diversity of algorithms from unconstrained to constrained optimization. They can be used to transform a general NLP problem into a sequence of well-behaved subproblems by adding a trust region constraint to the original problem and by approximating its functions by linear and quadratic models. The resulting subproblems are commonly quadratic and can be solved by classical methods for *quadratic programming* (QP). Therefore, 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, that is, the linearized constraints do not intercept the trust region constraint [40]. Several approaches have been proposed in order to resolve the possible conflict between satisfying the linearized constraints and the trust region constraint [41].

The application of trust region methods to power systems problems has increased over the last few years. Pajić and Clements [42, 43] used a trust region method for unconstrained optimization, along with a QR factorization, to obtain a reliable and more robust state estimator. Numerical simulations were carried out with the IEEE test systems up to the 118-bus and the results indicate that the globally convergent algorithm has superior performance when compared to the Gauss-Newton method [1]. Similarly, Hassaïne et al. [44] used the trust region Levenberg-Marquardt method [1] to increase the robustness of a new M-estimator. Costa et al. [45] proposed a combination of trust region and sequential-orthogonal methods in order to devise globally convergent state estimators.

Min and Shengsong [46] proposed a trust region IP approach for *linear programming* (LP) to solve OPF problems up to 662 buses. They used a relaxation of the linearized constraints in the trust region subproblem in order to overcome possible infeasibility. Wang et al. [47] proposed three separate techniques, including

a trust region based augmented Lagrangian method, for reliable and efficient computation of large-scale market-based OPF. Giacomoni et al. [48] used a trust region method to solve an LP OPF problem. El-Sobky et al. [49] introduced a trust region algorithm for unconstrained optimization to solve a reactive power compensation problem. Bisheh et al. [50] presented a trust region based augmented Lagrangian method for solving the economic power dispatch problem. Biehl [51] used the Steihaug's conjugate gradient method [1] to solve the trust region problems that arise from constrained power flow.

Torres [52] used complementarity functions to obtain an unconstrained optimization problem that is solved by the dogleg trust region method [1]. Souza et al. [53] developed a globally convergent trust region IP algorithm and applied it to the classical active losses minimization problem. They used the Byrd-Omojokun approach [40] to handle possible inconsistencies in the formulation of the trust region problems. The simulations were performed on the IEEE test systems up to 300-bus and a real system with 1211 buses. Additionally, cases with an increased load illustrate the robustness of the proposed algorithm. Four types of initializations, including random initialization, were also successfully tested and a detailed discussion of the numerical results was provided.

This work follows the Byrd-Omojokun trust region approach by Souza et al. [53]. This method consists in solving a sequence of QP problems and handles incompatible constraints by dividing the trust region problem into two subproblems, which are known as *vertical* and *horizontal* subproblems. The vertical and horizontal subproblems have nearly the same order of the trust region problem and they are solved alternately until convergence is obtained. The resolution of the vertical subproblem aims at finding a feasible point that is strictly inside the trust region. If the trust region problem is infeasible, the vertical subproblem reduces the primal infeasibility by minimizing the squared Euclidean norm of the equality constraints. As discussed in [41], it is the vertical subproblem that relaxes the original constraints enough to allow consistency. In the same way, 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. Additionally, if a feasible point is found for the vertical subproblem, the horizontal subproblem is feasible as well.

Even though the focus of a globally convergent trust region algorithm is on robustness, it is relevant to note that the Byrd-Omojokun technique is computation-

ally more expensive than the primal-dual IP algorithms. However, as pointed out in [53], the computational performance of the Byrd-Omojokun algorithm, as far as processing time is concerned, can be much further improved. Towards this end, note that when the optimal objective of the vertical subproblem (squared residuals of the linearized constraints) becomes zero, then a feasible point to the trust region subproblem has been found, that is, the linearized constraints and the trust region constraint are consistent. Thus, under certain assumptions, it may be possible to directly solve the trust region subproblem from that outer iteration on. The direct solution of the trust region subproblems, that is, the solution of the trust region subproblem without dividing it into the vertical and horizontal subproblems, can significantly reduce the total computational effort because a single QP problem is solved in each outer iteration. Additionally, note that if a feasible point is found during the solution of the first vertical problem, the total algorithm run time can be reduced by nearly half.

Another way to improve the performance of the Byrd-Omojokun algorithm is to reduce the computational effort in the solution of each trust region subproblem. Considering that a sequence of QP problems is solved, looser tolerance criteria on the vertical and horizontal subproblems may give the necessary approximation to the minimizer while reducing the overall number of matrix factorizations along with forwards and backwards solves. Similarly, the application of higher-order primal-dual IP methods to the solution of the trust region subproblems can also reduce the run time while increasing the robustness.

The main objective of this research is to investigate possible ways to improve the computational performance of the Byrd-Omojokun strategy proposed by Souza et al. [53] in the context of the solution of OPF problems. The motivation is to develop faster globally convergent trust region algorithms, which can be used when the available IP methods fail to converge. In addition, based on the works of Fletcher et al. [54, 55] and Gould et al. [56] concerning the use of *sequential quadratic programming* (SPQ) methods and *exact* ℓ_1 *penalty functions*, an alternative methodology is analyzed. This approach can handle incompatible constraints in the trust region problems and is expected to be as robust as and faster than the Byrd-Omojokun method described in [53].

1.3 Objectives of the Thesis

The objectives and main contributions of this research proposal are summarized as follows:

- To develop methodologies to reduce the computational effort of the Byrd-Omojokun trust region algorithm proposed in [53]. With this goal, a procedure to directly solve the trust region problem is investigated.
- To analyze the influence of using looser tolerance criteria on the vertical and the horizontal subproblems. It may be possible to reduce the number of matrix factorizations by approximating the solution of the trust region subproblems.
- To devise a globally convergent algorithm based on SQP methods and exact ℓ_1 penalty functions that is competitive in processing time with the Byrd-Omojokun strategy.
- To investigate the impact of the trust region parameters on the performance of the proposed techniques. As discussed in [57], the numerical efficiency of trust region algorithms can be further improved with a better parameter selection. Thus, it is relevant to examine different sets of parameters for the class of nonlinear OPF problems.
- To use infeasibility indicators to monitor the proposed trust region algorithms. The indicators can be employed to infer whether or not a trust region problem is infeasible.
- To implement the optimization methods used to solve the OPF problems. As discussed by [58], efficient computational codes of IP methods for LP can be written using the MATLAB® environment and their performance are comparable with a good Fortran routine. Additionally, [59] provides closed forms to efficiently implement OPF second derivatives.
- To discuss the numerical results obtained by the proposed techniques as well as possible enhancements of the methodologies and developed computational codes.

1.4 Contents of the Thesis

The Thesis is organized as follows. In Chapter 2, an introduction to penalty functions is given. The most popular techniques such as the quadratic and the ℓ_1 penalty methods, in addition to the augmented Lagrangian method, are briefly described.

In Chapter 3, the pure primal-dual and the primal-dual predictor-corrector IP methods for NLP are presented. These locally convergent IP methods, along with a trust region strategy, can provide a globally convergent algorithm to solve OPF problems.

In Chapter 4, the main concepts of trust region methods for unconstrained and constrained optimization are given. The general formulation of nonlinear least-squares problems and the most popular solution methods, named the Gauss-Newton and the Levenberg-Marquardt methods [1], are presented. Finally, the Byrd-Omojokun approach [40] and the sequential ℓ_1 quadratic programming [54] technique are described.

In Chapter 5, some modifications on the Byrd-Omojokun technique proposed by Souza et al. [53] are described with the aim of reducing its computational time. Additionally, a sequential ℓ_1 quadratic programming approach is developed as an alternative to the Byrd-Omojokun method.

In Chapter 6, the numerical experiments involving the proposed methods are presented and the main results are discussed. The simulations are carried out using different starting points, including randomly generated ones, and descriptive statistics are furnished.

Finally, in Chapter 7, the main conclusions of the research and the perspectives for future research are presented.

Chapter 2

Penalty Methods

THIS CHAPTER presents penalty methods that can be used to solve constrained optimization problems. Some emphasis is given to *the exact ℓ_1 penalty function*, which provides the main insights for the devised SQP method that will be discussed later in this work. Additionally, the quadratic penalty method and the augmented Lagrangian method are briefly described. These penalty methods are somehow related to the *logarithmic barrier* method [60], in which logarithmic terms prevent feasible iterates from moving too close to the boundary of the feasible region. This technique also forms part of the foundation for modern IP methods detailed in Chapter 3.

This chapter follows closely the discussion presented in [1] and tries to capture the most important points of it for the developed work. For a more comprehensive explanation concerning penalty methods the reader is referred to [1, 54, 61].

2.1 The Quadratic Penalty Method

Consider a general NLP problem in the following form:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & h(x) \leq 0 \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$, $f(x) : \mathbb{R}^n \to \mathbb{R}$, $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ and $h(x) : \mathbb{R}^n \to \mathbb{R}^p$.

Penalty methods usually define a sequence of penalty functions in which the penalty terms for constraint violations are multiplied by a positive coefficient and added to the original objective of the problem. By enlarging this coefficient, the constraint violations are penalized more severely in way that the minimizer of the penalty function tends to remain within the feasible region of (2.1).

According to [1], the simplest penalty function is the *quadratic penalty function*, in which the penalty terms are the squares of the constraint violations. For the general NLP problem (2.1), the quadratic penalty function can be defined as follows:

$$\psi(x, \eta) = f(x) + \frac{\eta}{2} \sum_{i=1}^{m} g_i^2(x) + \frac{\eta}{2} \sum_{i=1}^{p} \left([h_i(x)]^+ \right)^2 \tag{2.2}$$

where $[h_i(x)]^+$ denotes $\max(0, h_i(x))$ and η is the penalty parameter.

By driving η to ∞, the constraint violations are penalized with increasing severity. The general idea of penalty methods is to consider a sequence of values $\{\eta_k\}$ with $\eta_k \to \infty$ as $k \to \infty$ and to find the approximate minimizer x_k of $\psi(x, \eta_k)$ for each k. The parameter sequence $\{\eta_k\}$ can be chosen adaptively, based on the difficulty of minimizing the penalty function at each iteration and on estimates of the Lagrange multipliers.

One must observe that when only equality constraints are present in (2.2), $\psi(x, \eta)$ is smooth. On the other hand, when also considering inequalities, the penalty function $\psi(x, \eta)$ may be less smooth than the objective and constraint functions. Additionally, as η_k becomes larger, the minimization of $\psi(x, \eta_k)$ generally becomes more difficult to perform due to ill-conditioning of the Hessian $\nabla^2_{xx}\psi(x, \eta_k)$ [1]. Thus, when solving (2.2) by the Newton's method, the steps that it generates may not make a fast progress toward the minimizer of $\psi(x, \eta_k)$ since this method is based on the quadratic model. Furthermore, even when x is close to the mini-

mizer of $\psi(x, \eta_k)$, the quadratic Taylor series approximation to $\psi(x, \eta_k)$ around x is a good estimate of the true function only in a small neighborhood of x. This difficulty can be reduced in some way by a proper choice of the starting point in the next linearization or by defining a slightly larger penalty parameter.

2.2 The ℓ_1 Penalty Method

As discussed in [54], an attractive approach to NLP is to attempt to determine an *exact penalty function* $\psi(x, \eta)$ that is locally minimized by the solution x^* of (2.1). This holds out the possibility that the solution x^* can be found by a single application of an unconstrained minimization technique to $\psi(x, \eta)$ instead of requiring a sequential process. Besides, the property of being exact is desirable because it makes the performance of penalty methods less dependent on the strategy for updating the penalty parameter.

A popular nonsmooth penalty function for the general NLP problem (2.1) is *the exact ℓ_1 penalty function* defined as follows:

$$\psi(x, \eta) = f(x) + \eta \sum_{i=1}^{m} |g_i(x)| + \eta \sum_{i=1}^{p} [h_i(x)]^+ \tag{2.3}$$

where $[h_i(x)]^+$ stands for $\max(0, h_i(x))$ and η is called the ℓ_1 penalty parameter. Its name derives from the fact that the penalty term is η times the ℓ_1 norm of the constraint violations. In addition, one should observe that $\psi(x, \eta)$ is not differentiable at some x because of the presence of the absolute value $|g_i(x)|$ and $[h_i(x)]^+$ functions.

To illustrate how the ℓ_1 penalty method works and its dependency on the value of the penalty parameter η, consider two simple examples. First, consider the following minimization problem in one varible,

$$\begin{aligned} \min \quad & x \\ \text{subject to} \quad & 1 - x \leq 0 \end{aligned} \tag{2.4}$$

and its corresponding ℓ_1 penalty function:

$$\psi(x, \eta) = x + \eta [1 - x]^+ \tag{2.5}$$

As shown in Figure 2.1, the ℓ_1 penalty function (2.5) has a minimizer at $x^* = 1$ when $\eta > 1$, but is a monotone increasing function when $\eta < 1$.

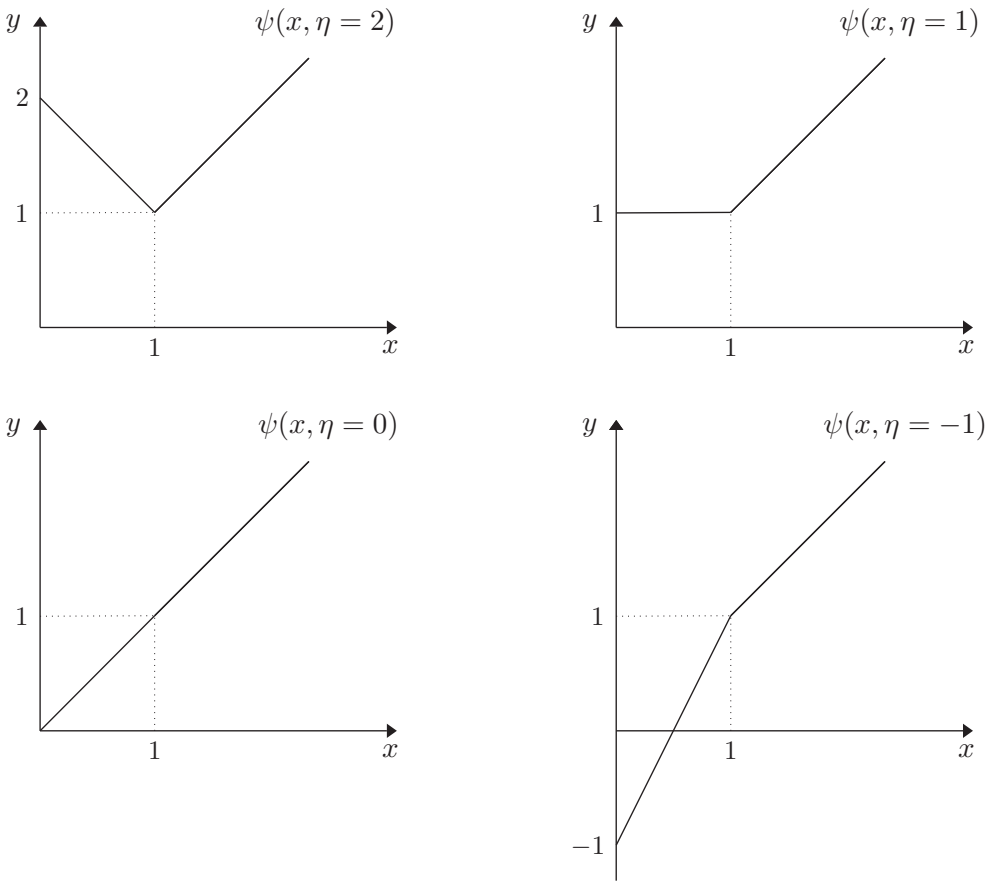

Figure 2.1: The exact ℓ_1 penalty function for problem (2.4) (Adapted from [1, 2]).

Similarly, consider the following problem:

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1^2 + x_2^2 - 2 = 0 \end{aligned} \tag{2.6}$$

and the related ℓ_1 penalty function:

$$\psi(x, \eta) = x_1 + x_2 + \eta |x_1^2 + x_2^2 - 2| \tag{2.7}$$

As can be seen in Figures 2.2 and 2.3, the ℓ_1 penalty function (2.7) is nonsmooth along the boundary of the circle defined by $x_1^2 + x_2^2 = 2$. Additionally, by considering that $|\lambda^*| = 0.5$ is the absolute value of the Lagrange multiplier corresponding to the equality constraint (2.6), function (2.7) has a minimizer x^* that coincides with the solution of problem (2.6) for all $\eta > |\lambda^*|$.

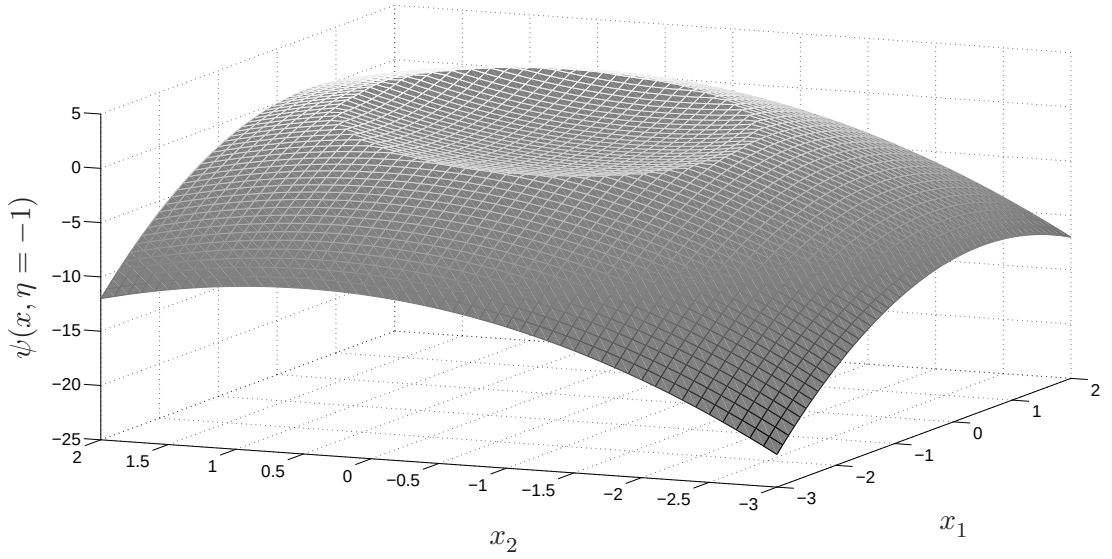

Figure 2.2: The exact ℓ_1 penalty function for problem (2.6) with $\eta = -1$ (Adapted from [1]).

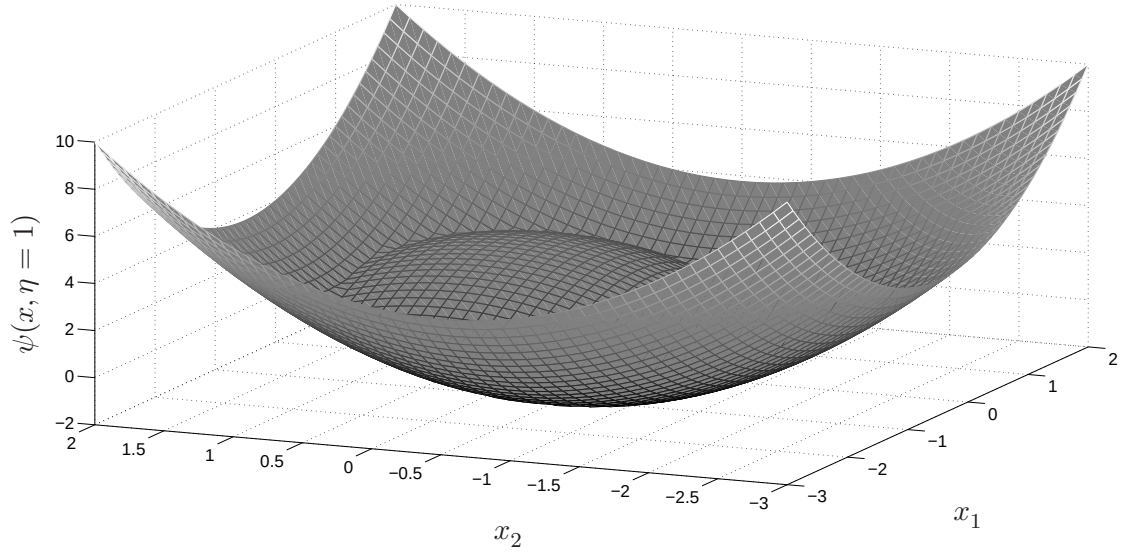

Figure 2.3: The exact ℓ_1 penalty function for problem (2.6) with $\eta = 1$ (Adapted from [1]).

According to [1], the simplest strategy for updating the ℓ_1 penalty parameter is to increase it by a constant multiple. If the initial value η_0 is too small, many cycles may be necessary to determine an appropriate value. On the other hand, if η_0 is excessively large, the penalty function will be difficult to minimize, possible requiring a large number of iterations. As will be discussed later in this work, a simple and effective scheme to update the ℓ_1 penalty parameter, based on the values of the problem's Lagrange multipliers, is devised in the context of a SQP method.

The exact ℓ_1 penalty function is nonsmooth and, consequently, nondifferentiable for some x. Instead of minimizing directly a nonsmooth function, the minimizer of $\psi(x,\eta)$ can be obtained by forming a simplified model of this function and finding the minimizer of this model. A model function can be defined by linearizing the constraints $g_i(x)$ and $h_i(x)$ of the NLP problem (2.1) and replacing the nonlinear objective $f(x)$ by a quadratic function, as follows:

$$\begin{aligned}\widetilde{\psi}(d,\eta) =& f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H d + \eta \sum_{i=1}^{m} |g_i(x) + \nabla g_i(x)^T d| + \\ &+ \eta \sum_{i=1}^{p} \left[h_i(x) + \nabla h_i(x)^T d \right]^+ \end{aligned} \tag{2.8}$$

where H is a symmetric matrix which generally contains second derivative information about $f(x)$, $g_i(x)$ and $h_i(x)$, $\forall i$. The model $\widetilde{\psi}(d,\eta)$ is still nonsmooth, but fortunately one can turn the minimization of $\widetilde{\psi}(d,\eta)$ into the solution of a smooth QP problem by introducing *elastic* variables $(p,q) \in \mathbb{R}^m$ and $t \in \mathbb{R}^p$, as follows:

$$\min \quad f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H d + \eta \sum_{i=1}^{m} (p_i + q_i) + \eta \sum_{i=1}^{p} t_i \tag{2.9a}$$

$$\text{subject to} \quad g(x) + \nabla g(x)^T d = p - q, \quad (p,q) \geq 0 \tag{2.9b}$$

$$h_i(x) + \nabla h_i(x)^T d \leq t, \quad t \geq 0 \tag{2.9c}$$

As discussed in [1], problem (2.9) can be solved by using a standard QP solver. Besides, even with the addition of a "box-shaped" trust region constraint of the form $\|d\|_\infty \leq \Delta$, problem (2.9) remains a QP problem. Additionally, it is not difficult to show that if the general NLP problem (2.1) assumes the particular form (1.1), that is, with only simple bound constraints on the variable x, the corresponding smooth QP problem can be reduced to the following:

$$\min \quad f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H d + \eta \sum_{i=1}^{m} (p_i + q_i) \tag{2.10a}$$

$$\text{subject to} \quad g(x) + \nabla g(x)^T d = p - q, \quad (p,q) \geq 0 \tag{2.10b}$$

$$\underline{x} \leq x_k + d \leq \overline{x} \tag{2.10c}$$

In this case, it is important to observe that the starting point d_0 for problem (2.10) should be in the closed interval $[\underline{x} - x_k, \overline{x} - x_k]$, which is a relatively simple

task because of the simple bounds.

2.3 The Augmented Lagrangian Method

As previously discussed, the quadratic penalty function (2.2) may suffer from ill-conditioning as η_k becomes larger. This section briefly shows the bound constrained formulation of the *augmented Lagrangian method* [1], which reduces the possibility of ill-conditioning by introducing explicit Lagrange multiplier estimates into the function to be minimized.

Given the general NLP problem (2.1), one can transform it to a problem with equality and simple bound constraints by introducing slack varibles $z \in \mathbb{R}^p$ and replacing the general inequalities $h(x) \leq 0$ by $h(x) + z = 0$ and $z \geq 0$. That is, problem (2.1) can be rewritten in the following form:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & h(x) + z = 0, \quad z \geq 0 \end{aligned} \tag{2.11}$$

The bound constrained augmented Lagrangian method incorporates only the equality constraints from (2.11) into the augmented Lagrangian function, as follows:

$$\begin{aligned} L(x, z, \lambda, \upsilon, \eta) = & f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \frac{\eta}{2} \sum_{i=i}^{m} g_i^2(x) + \\ & + \sum_{i=1}^{p} \upsilon_i \left[h_i(x) + z_i \right] + \frac{\eta}{2} \sum_{i=i}^{p} \left[h_i(x) + z_i \right]^2 \end{aligned} \tag{2.12}$$

where λ_i and υ_i are Lagrange multipliers. Additionally, the bound constraints are considered explicitly in the minimization problem, as follows:

$$\begin{aligned} \min \quad & L(x, z, \lambda, \upsilon, \eta) \\ \text{subject to} \quad & z \geq 0 \end{aligned} \tag{2.13}$$

The procedure consists in approximately solve (2.13), update the Lagrange multiplier vectors λ and υ, along with the penalty parameter η, and test for convergence. If the tolerance criteria are met, the algorithm stops. Otherwise, the process

is repeated. An efficient technique to solve the bound constrained problem (2.13) is the nonlinear version of the *gradient projection method*. The interested reader is referred to [1, 62] for more details.

2.4 Final Remarks

This chapter briefly presented penalty methods that are suitable for the solution of general NLP problems. The simplest penalty function is the quadratic one, but the quadratic penalty method suffers from ill-conditioning as the penalty parameter becomes larger. The bound constrained augmented Lagrangian method is a closed related technique that reduces the possibility of ill-conditioning by introducing explicit Lagrange multiplier estimates into the augmented Lagrangian function.

The ℓ_1 penalty method uses the nonsmooth exact ℓ_1 penalty function to handle constraint violations. The main concepts regarding this penalty method, along with its important smooth reformulation as a QP problem, are used later in the Thesis to devise a SQP method in the context of the solution of OPF problems by trust region IP methods.

Chapter 3

Interior Point Methods for Nonlinear Programming

THERE IS a great diversity of optimization methods to solve NLP problems in the general form (1.1) [1, 33, 39, 54]. Particularly, the solution of OPF problems has been successfully addressed by primal-dual IP methods [8, 59]. Modern IP methods have their origins in the *logarithmic barrier* methods attributed to Frisch [60] and later extended by Fiacco and McCormick [63] for nonlinear inequality problems. However, at least for large-scale LP problems, the practical performance of the *simplex method* [61, 64] appeared to be insuperable until the breakthrough by Karmarkar [65] in 1984. After Karmarkar's paper about his new polynomial-time algorithm, many variants of IP methods were developed. This chapter briefly describes two well known variants called the *pure primal-dual* (or simply primal-dual) method and the *primal-dual predictor-corrector* method [34].

3.1 The Primal-Dual Interior Point Method

In order to derive the primal-dual algorithm, first consider an NLP problem in the form (1.1):

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x} \leq x \leq \overline{x} \end{aligned}$$

where $x \in \mathbb{R}^n$, $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^m$. The primal-dual method operates on the following modified problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x} + s - x = 0, \quad s \geq 0 \\ & x + z - \overline{x} = 0, \quad z \geq 0 \end{aligned} \tag{3.1}$$

where all inequality constraints are transformed into equalities by adding vectors of slack variables $s \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$. The nonnegativity conditions $s \geq 0$ and $z \geq 0$ in (3.1) are incorporated into logarithmic barrier functions that are added to the objective function and the following problem is obtained:

$$\begin{aligned} \min \quad & f(x) - \mu_k \sum_{i=1}^{n} (\ln s_i + \ln z_i) \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x} + s - x = 0, \quad s > 0 \\ & x + z - \overline{x} = 0, \quad z > 0 \end{aligned} \tag{3.2}$$

where $\mu_k > 0$ is the barrier parameter, which is monotonically reduced to zero as the iterative process evolves, i.e., $\mu_0 > \mu_1 > \cdots > \mu_k > \cdots > \mu_\infty = 0$. The strict positivity conditions $s > 0$ and $z > 0$ must be imposed in order to define the logarithmic terms. However, these conditions are implicitly handled through a step length control.

The first-order necessary optimality conditions for the modified problem (3.2), for a fixed barrier parameter μ_k, can be derived from the Lagrangian function

$L(y;\mu_k)$ associated with problem (3.2), which is defined as follows:

$$L(y;\mu_k) = f(x) - \mu_k \sum_{i=1}^{n} (\ln s_i + \ln z_i) + \lambda^T g(x) + \pi^T(\underline{x} + s - x) + \upsilon^T(x + z - \overline{x}) \quad (3.3)$$

where $\lambda \in \mathbb{R}^m$, $\pi \in \mathbb{R}^n_+$ and $\upsilon \in \mathbb{R}^n_+$ are vectors of Lagrange multipliers, also called dual variables, and $y = (s, z, \pi, \upsilon, \lambda, x)$ is a column vector of all variables.

Under the linear independence constraint qualification, a local minimum of (3.2) is a stationary point of the Lagrangian function, which must satisfy the first order KKT conditions:

$$\nabla_s L = \pi - \mu_k S^{-1} e = 0 \quad (3.4a)$$
$$\nabla_z L = \upsilon - \mu_k Z^{-1} e = 0 \quad (3.4b)$$
$$\nabla_\pi L = \underline{x} + s - x = 0 \quad (3.4c)$$
$$\nabla_\upsilon L = x + z - \overline{x} = 0 \quad (3.4d)$$
$$\nabla_\lambda L = g(x) = 0 \quad (3.4e)$$
$$\nabla_x L = \nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon = 0 \quad (3.4f)$$

where $S = \mathrm{diag}(s_1, \ldots, s_n)$, $Z = \mathrm{diag}(z_1, \ldots, z_n)$ and $e = (1, 1, \ldots, 1)$.

The system of nonlinear equations (3.4) can be conveniently expressed as:

$$S\pi - \mu_k e = 0 \quad (3.5a)$$
$$Z\upsilon - \mu_k e = 0 \quad (3.5b)$$
$$\underline{x} + s - x = 0 \quad (3.5c)$$
$$x + z - \overline{x} = 0 \quad (3.5d)$$
$$g(x) = 0 \quad (3.5e)$$
$$\nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon = 0 \quad (3.5f)$$

In order to solve (3.5), an iteration of the primal-dual IP method applies a single iteration of the Newton's method for root finding, then calculates the step length along the Newton's direction, updates the variables and reduces the barrier parameter μ_k. The iterative process ends when the primal and dual infeasibilities, along with the complementarity gap, are smaller than specified tolerances. The main steps of the primal-dual IP method are given by Algorithm 3.1.

1. Set $k = 0$, choose $\mu_0 > 0$ and an initial point y_0 that satisfies the strict positivity conditions $(s_0, z_0, \pi_0, \upsilon_0) > 0$.
2. Apply the Newton's method to (3.5) at the current point,

$$\nabla^2_{yy} L(y_k; \mu_k)\Delta y = -\nabla_y L(y_k; \mu_k),$$

   and solve the resulting linear system for the Newton's direction Δy.
3. Calculate the step length α_k in the Newton's direction and obtain the new estimate of the solution by $y_{k+1} = y_k + \alpha_k \Delta y$.
4. If y_{k+1} satisfies the convergence criteria, then STOP. Otherwise, set $k \leftarrow k + 1$, reduce the barrier parameter μ_k and return to step 2.

Algorithm 3.1: Primal-dual IP method to solve (1.1).

3.1.1 Computation of the Search Directions

Even though the system resulting from the KKT conditions (3.5) is nonlinear, its solution is usually approximated by a single iteration of the Newton's method. In fact, the Newton's direction is used to obtain an approximation $\widetilde{y}(\mu_k)$ for the local solution $y(\mu_k)$ as the barrier parameter is reduced. If the Newton's method were applied until convergence and μ_k reduced in infinitesimal steps, the trajectory described by these points would be called the *primal-dual barrier trajectory* and would converge to y^* as $\mu_k \to 0$ [1].

By taking the first order terms of the Taylor series expansion for (3.5) around y_k, the following sparse linear system is obtained:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta \upsilon \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_k e \\ Z\upsilon - \mu_k e \\ \underline{x} + s - x \\ x + z - \overline{x} \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon \end{bmatrix} \tag{3.6}$$

where $\Pi = \mathrm{diag}(\pi_1, \ldots, \pi_n)$, $\Upsilon = \mathrm{diag}(\upsilon_1, \ldots, \upsilon_n)$, and $\nabla^2_{xx} L(y)$ is the Hessian of the

Lagrangian function, defined as

$$\nabla^2_{xx} L(y) = \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x) \tag{3.7}$$

It is important to observe that the computation of the Hessian $\nabla^2_{xx} L(y)$ can be time-consuming if it is not efficiently implemented. An efficient procedure to evaluate the Hessian $\nabla^2_{xx} L(y)$ for OPF problems is described in [59].

3.1.2 Update of Variables

The new estimates for the primal and dual variables are calculated as:

$$\begin{aligned}
x_{k+1} &= x_k + \alpha_k^P \Delta x \\
s_{k+1} &= s_k + \alpha_k^P \Delta s \\
z_{k+1} &= z_k + \alpha_k^P \Delta z \\
\lambda_{k+1} &= \lambda_k + \alpha_k^D \Delta \lambda \\
\pi_{k+1} &= \pi_k + \alpha_k^D \Delta \pi \\
\upsilon_{k+1} &= \upsilon_k + \alpha_k^D \Delta \upsilon
\end{aligned} \tag{3.8}$$

where $\alpha_k^P \in (0, 1]$ and $\alpha_k^D \in (0, 1]$ are the primal and dual step lengths, respectively.

The maximum step lengths that are possible to be taken in the Newton's direction are determined by:

$$\alpha_k^P = \min \left\{ 1,\ \gamma \times \min_i \left\{ \frac{-s_i^k}{\Delta s_i} \middle| \Delta s_i < 0,\ \frac{-z_i^k}{\Delta z_i} \middle| \Delta z_i < 0 \right\} \right\} \tag{3.9a}$$

$$\alpha_k^D = \min \left\{ 1,\ \gamma \times \min_i \left\{ \frac{-\pi_i^k}{\Delta \pi_i} \middle| \Delta \pi_i < 0,\ \frac{-\upsilon_i^k}{\Delta \upsilon_i} \middle| \Delta \upsilon_i < 0 \right\} \right\} \tag{3.9b}$$

The scalar $\gamma \in (0, 1)$ is a safeguard factor to ensure that the next point will satisfy the strict positivity conditions (typically, $\gamma = 0.99995$).

The use of different step lengths allows the search to be performed in both the primal and dual spaces and this is one of the advantages of primal-dual IP methods. Moreover, for practical problems, the use of distinct primal and dual steps usually results in faster convergence. In NLP problems, however, the primal and dual variables are also correlated by the dual feasibility condition (3.5f), which

does not allow rigorously distinct steps in the primal and dual spaces. Therefore, an unique (identical) step length should be used to update the primal and dual variables, which is calculated as:

$$\alpha_k^P = \alpha_k^D \leftarrow \min\left\{\alpha_k^P,\ \alpha_k^D\right\} \tag{3.10}$$

In practice, different and equal step lengths have both been used [59]. However, preliminary numerical experiments on OPF problems have indicated that algorithms based on equal steps are more likely to fail when a problem is nearly infeasible, that is, when the feasible region is almost empty. In such a case, the primal step α_k^P tends to zero as the variables approach the boundaries.

3.1.3 Reducing the Barrier Parameter

The scheme used to reduce the barrier parameter μ_k is an extension of those successfully implemented for LP and QP IP methods[33]. At iteration k, the residual of the complementarity conditions, called *complementarity gap*, is obtained as follows:

$$\rho_k = s_k^T \pi_k + z_k^T \upsilon_k \tag{3.11}$$

If the iterative process converges to an optimal solution, then $\rho_k \to 0$. The relation between ρ_k and μ_k is implicitly defined in equations (3.5a), (3.5b) and (3.11) in the following form:

$$\sum_{i=1}^{n} s_i \pi_i + \sum_{i=1}^{n} z_i \upsilon_i = 2n\mu_k = \rho \tag{3.12}$$

from which μ_k can be rewritten as a function of the complementarity gap:

$$\mu_{k+1} = \sigma \frac{\rho_k}{2n} \tag{3.13}$$

where σ is a reduction factor on average complementarity gap. The parameter $\sigma \in [0, 1]$ is called *centering parameter* and the KKT system (3.5) defines a step towards a point at the barrier trajectory. On the other hand, $\sigma = 0$ provides a step in the pure Newton's direction, which is called *affine-scaling* direction. In computational practice, in order to balance the two objectives of reducing μ_k while improving the centrality of the iterates, σ is chosen within the open interval $(0, 1)$.

3.1.4 Convergence Criteria

The convergence criteria for the primal-dual IP method can be established as follows:

$$\max\left\{\max_i\left\{\underline{x}_i - x_i^k\right\}, \max_i\left\{x_i^k - \overline{x}_i\right\}, \|g(x_k)\|_\infty\right\} \leq \epsilon_1 \tag{3.14a}$$

$$\frac{\|\nabla f(x_k) + \nabla g(x_k)\lambda - \pi_k + \upsilon_k\|_\infty}{1 + \|x_k\|_2 + \|\lambda_k\|_2 + \|\pi_k\|_2 + \|\upsilon_k\|_2} \leq \epsilon_1 \tag{3.14b}$$

$$\frac{\rho_k}{1 + \|x_k\|_2} \leq \epsilon_2 \tag{3.14c}$$

If the criteria (3.14a), (3.14b) and (3.14c) are satisfied, then the primal and dual feasibility conditions, along with the complementarity condition, are all satisfied and the solution point y^* has precision equal to ϵ_1. Typically, the tolerance values are set to $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-2}\epsilon_1$.

3.2 The Predictor-Corrector Interior Point Method

The factorization of the coefficient matrix in (3.6) is the most time-consuming task of an IP algorithm [35, 59]. Therefore, each matrix factorization must be used as many times as possible before obtaining a new estimate. The predictor-corrector IP method [34] follows this idea by using the same factorization to calculate two different directions: the predictor and the corrector. The final direction can be obtained by the sum of these two directions. By doing that, the method improves the centrality, allows larger step lengths to be taken and reduces the overall number of iterations.

The predictor-corrector IP algorithm is obtained by substituting the new point expression $y_{k+1} = y_k + \Delta y$ directly into (3.5) to yield the approximation:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta\pi \\ \Delta\upsilon \\ \Delta\lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi \\ Z\upsilon \\ \underline{x} + s - x \\ x + z - \overline{x} \\ g(x) \\ \nabla_x L(y) \end{bmatrix} + \begin{bmatrix} \mu_k e \\ \mu_k e \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \Delta S \Delta\pi \\ \Delta Z \Delta\upsilon \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.15}$$

where $\nabla_x L(y) = \nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon$, $\Delta S = \mathrm{diag}(\Delta s_1, \ldots, \Delta s_n)$, and $\Delta Z = \mathrm{diag}(\Delta z_1, \ldots, \Delta z_n)$.

The major difference between systems (3.6) and (3.15) is that the right-hand side vector cannot be determined beforehand because of the nonlinear Δ-terms $\Delta S \Delta \pi$ and $\Delta Z \Delta \upsilon$. The full direction Δy obtained from (3.15) can be divided into three components:

$$\Delta y = \Delta y_{\text{af}} + \Delta y_{\text{ce}} + \Delta y_{\text{co}} \tag{3.16}$$

where each component is determined by one of the three vectors in the right-hand side of (3.15). These three components of the full direction Δy can be understood separately as follows:

Δy_{af} : is the *affine-scaling* direction, which is obtained by setting $\mu = 0$ in (3.6). This direction is determined by the first vector on the right-hand side of (3.15), that is, it is the solution of the system:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{\text{af}} \\ \Delta z_{\text{af}} \\ \Delta \pi_{\text{af}} \\ \Delta \upsilon_{\text{af}} \\ \Delta \lambda_{\text{af}} \\ \Delta x_{\text{af}} \end{bmatrix} = - \begin{bmatrix} S\pi \\ Z\upsilon \\ \underline{x} + s - x \\ x + z - \overline{x} \\ g(x) \\ \nabla_x L(y) \end{bmatrix} \tag{3.17}$$

The affine-scaling direction focuses on reducing the primal and dual infeasibilities, and the complementarity gap.

Δy_{ce} : is the *centering direction*, whose length is determined by the barrier parameter μ_k. This direction is defined by the second vector on the right-hand side of (3.15), that is, by solving the linear system:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{\text{ce}} \\ \Delta z_{\text{ce}} \\ \Delta \pi_{\text{ce}} \\ \Delta \upsilon_{\text{ce}} \\ \Delta \lambda_{\text{ce}} \\ \Delta x_{\text{ce}} \end{bmatrix} = \begin{bmatrix} \mu_k e \\ \mu_k e \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

The centering direction maintains the point sufficiently far from the boundaries of the feasible region and ideally near to the barrier trajectory. By

doing this, the method increases the chance of taking a larger step in the next iteration.

Δy_{co} : is the *corrector direction*, which tries to compensate some of the nonlinearities in the affine-scaling direction Δy_{af}. The corrector direction is determined by the third vector on the right-hand side of (3.15), which is found by solving the following system:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{\text{co}} \\ \Delta z_{\text{co}} \\ \Delta \pi_{\text{co}} \\ \Delta \upsilon_{\text{co}} \\ \Delta \lambda_{\text{co}} \\ \Delta x_{\text{co}} \end{bmatrix} = - \begin{bmatrix} \Delta S \Delta \pi \\ \Delta Z \Delta \upsilon \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.19}$$

Clearly, the combination of the directions Δy_{af} and Δy_{ce} is given by the solution of (3.6). However, to deal with the nonlinearities in (3.15), the direction Δy_{af} is obtained separately from and before Δy_{ce}. This furnishes the possibility to calculate approximations to the barrier parameter μ_{k+1} and to the second order terms $\Delta S \Delta \pi$ and $\Delta Z \Delta \upsilon$.

3.2.1 The Predictor Step

In order to find an step that approximates the solution of (3.15), it is enough to solve (3.17) for the affine-scaling direction:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s_{\text{af}} \\ \Delta z_{\text{af}} \\ \Delta \pi_{\text{af}} \\ \Delta \upsilon_{\text{af}} \\ \Delta \lambda_{\text{af}} \\ \Delta x_{\text{af}} \end{bmatrix} = - \begin{bmatrix} S\pi \\ Z\upsilon \\ \underline{x} + s - x \\ x + z - \overline{x} \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon \end{bmatrix} \tag{3.20}$$

The direction Δy_{af} is used in two distinct ways: (a) to approximate the Δ-terms on the right-hand side of (3.15); and (b) to dynamically estimate the barrier parameter μ_{k+1}.

A new estimate μ_{k+1} is calculated by considering (3.9) to determine the step α_{af}

if the direction Δy_{af} were applied:

$$\alpha_{\text{af}}^{P} = \min\left\{1,\ \gamma \times \min_{i}\left\{\frac{-s_i^k}{\Delta s_i^{\text{af}}}\middle|\Delta s_i^{\text{af}} < 0,\ \frac{-z_i^k}{\Delta z_i^{\text{af}}}\middle|\Delta z_i^{\text{af}} < 0\right\}\right\} \tag{3.21a}$$

$$\alpha_{\text{af}}^{D} = \min\left\{1,\ \gamma \times \min_{i}\left\{\frac{-\pi_i^k}{\Delta \pi_i^{\text{af}}}\middle|\Delta \pi_i^{\text{af}} < 0,\ \frac{-\upsilon_i^k}{\Delta \upsilon_i^{\text{af}}}\middle|\Delta \upsilon_i^{\text{af}} < 0\right\}\right\} \tag{3.21b}$$

Once the primal and dual steps in the affine-scaling direction are available, a new estimate for the complementarity gap is obtained as:

$$\rho_{\text{af}} = (s_k + \alpha_{\text{af}}^{P}\Delta s_{\text{af}})^T(\pi_k + \alpha_{\text{af}}^{D}\Delta \pi_{\text{af}}) + (z_k + \alpha_{\text{af}}^{P}\Delta z_{\text{af}})^T(\upsilon_k + \alpha_{\text{af}}^{D}\Delta \upsilon_{\text{af}}) \tag{3.22}$$

Finally, the approximation μ_{af} for μ_{k+1} is calculated as:

$$\mu_{\text{af}} = \min\left\{\left(\frac{\rho_{\text{af}}}{\rho_k}\right)^2, \sigma\right\}\frac{\rho_{\text{af}}}{2n} \tag{3.23}$$

This procedure chooses μ_{af} small if the direction Δy_{af} produces a significant decrease on the complementarity gap, that is, if $\rho_{\text{af}} \ll \rho_k$. Otherwise, it defines μ_{af} to be a larger value.

3.2.2 The Corrector Step

Instead of calculating the combined direction $\Delta y_{\text{ce}} + \Delta y_{\text{co}}$ and then add it to Δy_{af}, the full direction Δy is obtained in one step by solving the following system:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & \nabla g(x)^T \\ 0 & 0 & -I^T & I^T & \nabla g(x) & \nabla^2_{xx} L(y) \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta \upsilon \\ \Delta \lambda \\ \Delta x \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_{\text{af}} e + \Delta S_{\text{af}} \Delta \pi_{\text{af}} \\ Z\upsilon - \mu_{\text{af}} e + \Delta Z_{\text{af}} \Delta \upsilon_{\text{af}} \\ \underline{x} + s - x \\ x + z - \overline{x} \\ g(x) \\ \nabla f(x) + \nabla g(x)\lambda - \pi + \upsilon \end{bmatrix} \tag{3.24}$$

Both the predictor and the corrector steps make use of the same matrix factorization. Therefore, the additional computational effort in the predictor-corrector method, when compared to the pure primal-dual approach, is the solution of an additional linear system to calculate Δy_{af} and the computation of the approximation

μ_{af}. The benefits of using this method usually compensate the extra effort and it generally converges faster than the pure primal-dual variant. The main steps of the predictor-corrector IP method are given by Algorithm 3.2.

1. Set $k = 0$, choose $\mu_0 > 0$ and a start point y_0 that satisfies the strict positivity conditions $(s_0, z_0, \pi_0, \upsilon_0) > 0$.
2. Form the matrix $\nabla^2_{yy} L(y_k; \mu_k)$ and obtain its factorization.
   (a) Solve the system (3.17) for the direction Δy_{af}, calculate α_{af} from (3.21) and obtain μ_{af} from (3.22).
   (b) Solve the system (3.24) for the full direction Δy.
3. Calculate the step length α_k along the direction Δy and obtain the new point as $y_{k+1} = y_k + \alpha_k \Delta y$.
4. If y_{k+1} satisfies the convergence criteria, then STOP. Otherwise, set $k \leftarrow k + 1$, reduce the barrier parameter μ_k, and return to step 2.

Algorithm 3.2: Primal-dual predictor-corrector IP method to solve (1.1).

3.3 Infeasibility Detection and Handling

Optimization algorithms can fail to converge due to several factors, including parameter selection, initialization, ill-conditioning and scaling problems. Particularly, primal-dual IP point methods may present some difficulties to converge when the iterates are badly centered, which motivated the development of the multiple centrality corrections method [35].

An important aspect when solving an optimization problem is to efficiently detect and handle infeasibility. A problem is called infeasible when it has an empty feasible set, that is, there is no point that can simultaneously satisfy all constraints. For primal-dual IP methods, an infeasible simple bound can cause its respective slack variable to rapidly decrease to zero (limited by the logarithmic barrier) in an attempt to become negative. As a direct consequence of very small slack variables, the primal step calculated from (3.9) decreases and the algorithm fails to converge. At the same time, the corresponding Lagrange multiplier increases very fast, providing the sensitivity that an active constraint has been found.

In [21, 66, 67], a methodology to detect and handle infeasible simple bounds is proposed in the context of the solution of OPF problems by IP methods. This

approach starts to monitor infeasible bounds whenever the primal step α_k^P becomes too small. That is, if $\alpha_k^P \leq \xi$ (typically, $\xi = 0.0001$) than problem (1.1) is possibly infeasible and the algorithm defines the following sets of lower and upper infeasible bounds:

$$\mathcal{I}_L = \{i \mid s_i^k \leq \tau_1 \text{ and } \pi_i^k \geq \tau_2,\ i = 1, 2, \ldots, n\} \tag{3.25}$$

$$\mathcal{I}_U = \{i \mid z_i^k \leq \tau_1 \text{ and } \upsilon_i^k \geq \tau_2,\ i = 1, 2, \ldots, n\} \tag{3.26}$$

All the infeasible bounds must be included in the set $\mathcal{I}_L \cup \mathcal{I}_U$, whereas the feasible ones must not be wrongly included. The methodology to handle infeasible bounds consists in solving the following modified problem:

$$\begin{aligned} \min \quad & f(x) + c\sum_{i\in\mathcal{I}_L} w_i(x_i - \underline{x}_i)^2 + c\sum_{i\in\mathcal{I}_U} w_i(x_i - \overline{x}_i)^2 \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, \quad \forall i \notin \mathcal{I}_L \cup \mathcal{I}_U \end{aligned} \tag{3.27}$$

where c is a large positive number that balances the minimization of the objective function against the minimization of the violation of infeasible bounds and w_i's are weights that differentiates the relative importance among these bounds.

A practical formulation of (3.27) keeps all the original constraints in order to preserve the structure of the Newton's system, but it changes the levels of the infeasible simple bounds so as to relax them. Therefore, the following problem is solved:

$$\begin{aligned} \min \quad & f(x) + c\sum_{i\in\mathcal{I}_L} w_i(x_i - \underline{x}_i)^2 + c\sum_{i\in\mathcal{I}_U} w_i(x_i - \overline{x}_i)^2 \\ \text{subject to} \quad & g(x) = 0 \\ & \widetilde{\underline{x}} \leq x \leq \widetilde{\overline{x}} \end{aligned} \tag{3.28}$$

where $\widetilde{\underline{x}}_i = \underline{x}_i - \delta_i$ if $i \in \mathcal{I}_L$ and $\widetilde{\overline{x}}_i = \overline{x}_i + \delta_i$ if $i \in \mathcal{I}_U$, and δ_i's are the values by which the infeasible bounds are relaxed.

3.3.1 Primal-Dual Logarithmic Indicators

As discussed in [68], the identification of subgroups of variables can be used in several ways to obtain computational advantage. Extending the idea to infeasible problems, infeasibility is monitored by using logarithmic primal-dual indicators, which are functions of the slack variables and the Lagrange multipliers, defined as follows:

$$PDL_i^L = \log\left(\frac{s_i^k}{\pi_i^k}\right) \tag{3.29}$$

$$PDL_i^U = \log\left(\frac{z_i^k}{v_i^k}\right) \tag{3.30}$$

When k goes to infinity, the limits of PDL_i^U and PDL_i^U behave as follows:

$$\lim_{k\to\infty} PDL_i^L(s_i^k, \pi_i^k) = \lim_{k\to\infty} PDL_i^U(z_i^k, v_i^k) = \begin{cases} -\infty, i \in \mathcal{Z} \\ +\infty, i \notin \mathcal{Z} \end{cases} \tag{3.31}$$

where $\mathcal{Z}$ is the index set of active bound constraints.

Similarly to strong active simple bound constraints, infeasible bounds might accomplish the behavior predicted by (3.31). Figure 3.1 illustrates the profile of the PDL_i^U indicator for an infeasible OPF problem before and after the activation, at iteration 9, of the infeasibility handling algorithm proposed in [67].

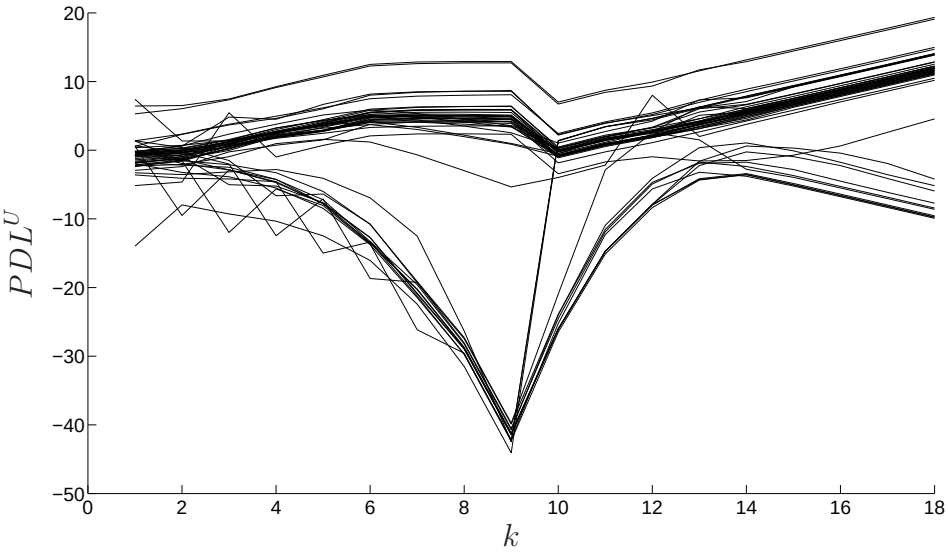

Figure 3.1: Primal-Dual logarithmic indicators of infeasibility.

It can be seen from Figure 3.1 that once the feasibility has been restored by relaxing the infeasible simple bounds, the primal-dual logarithmic indicators do not return to their previous levels. In fact, when the algorithm converges, subgroups of active and inactive constraints can be easily identified. The main observation is that the absolute value of the primal-dual logarithmic indicator for an active constraint is much smaller and different from that found for an infeasible constraint.

3.4 Final Remarks

This chapter presented the pure primal-dual and the primal-dual predictor-corrector IP methods for NLP. The solution of the Newton's system (3.6) is by far the most time consuming task of these algorithms. Thus, to obtain efficient IP codes, it is important to concentrate efforts to improve the factorization, forward and backward solves. In this work, the solution of the linear system (3.6) is performed by MATLAB's built-in functions `mldivide()` and `lu()`, which invariably use well-established routines from the Linear Algebra PACKage (LAPACK) [69] or the UMFPACK [70] packages.

Together with the multiple centrality corrections approach [36], the pure primal-dual and the primal-dual predictor-corrector IP methods have been successfully applied to OPF problems. An important practical feature of these optimization techniques is that the number of iterations does not seem to be very sensitive to the network size or to the number of control variables [8, 59], so they are suitable for large-scale OPF problems.

Infeasibility detection and handling is one of the most important resources of an optimization program. Particularly, in the context of the solution by primal-dual IP methods, the corresponding primal-dual logarithmic indicator can readily provide the information on whether or not an infeasible constraint is found.

Chapter 4

Trust Region Methods

OPF PROBLEMS lie in the class of nonlinear nonconvex constrained optimization. As power systems increase and operate heavy loaded, the solution of these problems become more difficult. Moreover, locally convergent algorithms may fail to converge when the initial point is not near to a solution. Fortunately, globally convergent techniques can be used to obtain convergence from remote starting points.

Trust region methods are one of the globalization techniques that has been used to provide global convergence properties to a great diversity of algorithms. According to [71], the idea of defining a region of trust for the search direction was first suggested for nonlinear least squares problems and later extended to general unconstrained optimization. Thus, it is worth to briefly describe the formulation and solution techniques for these problems before introducing the basic concepts of trust region.

4.1 Nonlinear Least Squares Problems

Nonlinear least squares problems can be stated in the following classical form:

$$\min \quad f(x) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x) = \frac{1}{2}\|r(x)\|^2 \tag{4.1}$$

where $r : \mathbb{R}^n \mapsto \mathbb{R}^m$ is continuously differentiable and $\|\cdot\|$ denotes the ℓ_2 vector norm. These problems became very popular because of their special features when compared to general unconstrained problems. The first and second order derivatives of $f(x)$ can be expressed in terms of the residual functions $r_i(x)$ as follows:

$$\nabla f(x) = \nabla r(x) r(x) \tag{4.2}$$

$$\nabla^2 f(x) = \nabla r(x) \nabla r(x)^T + \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x) \tag{4.3}$$

It is clear from (4.2) that the gradient $\nabla f(x)$ can be obtained if the first derivative of each function $r_i(x)$ is easily calculated. Furthermore, by knowing the gradient $\nabla r(x)$, the first part of the Hessian $\nabla^2 f(x)$ is immediately available. The second part of the Hessian is expected to be small when the residuals r_i are small or when each function r_i is nearly linear ($\|\nabla^2 r_i(x)\| \approx 0$).

The Gauss-Newton method [1], which can be understood as a modification of the Newton's method with line search, exploit those features by neglecting the second order term from (4.3), that is, $\nabla^2 f(x) \approx \nabla r(x) \nabla r(x)^T$, and by defining a search direction d_k^{GN} as:

$$\nabla r(x_k) \nabla r(x_k)^T d_k^{\text{GN}} = -\nabla r(x_k) r(x_k) \tag{4.4}$$

Interestingly, the Gauss-Newton direction d_k^{GN} can also be obtained from the linear model function $r(x_k + p) \approx r(x_k) + \nabla r(x_k)^T d$, by substituting it into the expression $f(x) = \frac{1}{2}\|r(x)\|^2$ and minimizing over d. That is, d_k^{GN} is the solution of the linear least squares problem

$$\min \quad \frac{1}{2}\|r(x_k) + \nabla r(x_k)^T d\|^2 \tag{4.5}$$

which is also equivalent to replace the objective function in (4.1) by the quadratic

model

$$m_k(d) = \frac{1}{2}\|r(x_k)\|^2 + [\nabla r(x_k) r(x_k)]^T d + \frac{1}{2} d^T \nabla r(x_k) \nabla r(x_k)^T d \tag{4.6}$$

On the other hand, instead of using a line search procedure, the Levenberg-Marquardt method [1, 72, 73] applies a trust region strategy. The main idea of this approach is to restrict the search to a region around the current iterate x_k within the quadratic model $m_k(d)$ in (4.6) closely approximates the original nonlinear objective function $f(x)$. For a spherical trust region, the problem to be solved at each iteration is:

$$\begin{aligned} \min \quad & \frac{1}{2}\|r(x_k)\|^2 + [\nabla r(x_k) r(x_k)]^T d + \frac{1}{2} d^T \nabla r(x_k) \nabla r(x_k)^T d \\ \text{subject to} \quad & \|d\| \leq \Delta_k \end{aligned} \tag{4.7}$$

where $\Delta_k > 0$ is the radius of the trust region. When the solution d^* of (4.7) reaches the trust region, that is, $\|d^*\| = \Delta_k$, then it also solves the following equality constrained problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|r(x_k)\|^2 + [\nabla r(x_k) r(x_k)]^T d + \frac{1}{2} d^T \nabla r(x_k) \nabla r(x_k)^T d \\ \text{subject to} \quad & \|d\| = \Delta_k \end{aligned} \tag{4.8}$$

If d^* is a local minimum of (4.8), then there is a strictly positive Lagrange multiplier $\lambda^* \in \mathbb{R}_+$ that satisfy the KKT first order optimality conditions:

$$\left[\nabla r(x_k) \nabla r(x_k)^T + \lambda I\right] d = -\nabla r(x_k) r(x_k) \tag{4.9a}$$

$$\|d\| - \Delta_k = 0 \tag{4.9b}$$

Otherwise, if d^* remains strictly inside the trust region, that is, $\|d^*\| < \Delta_k$, then $d^* = d^{\text{GN}}$ is the solution of the unconstrained problem (4.5) and $\lambda^* = 0$, which represents an inactive constraint. Finally, the constraints $\|d\| - \Delta_k \leq 0$ and $\lambda \geq 0$ can be combined as a single *complementarity condition* and the KKT first order optimality conditions for problem (4.7) become:

$$\left[\nabla r(x_k) \nabla r(x_k)^T + \lambda I\right] d^{\text{LM}} = -\nabla r(x_k) r(x_k) \tag{4.10a}$$

$$\lambda \left(\|d^{\text{LM}}\| - \Delta_k\right) = 0 \tag{4.10b}$$

It is important to mention that both Gauss-Newton and Levenberg-Marquardt methods use the same Hessian approximations, thus their local convergence properties are similar. In addition, these methods are expected to perform poorly on large-residual problems, in which the second-order part of the Hessian $\nabla^2 f(x)$ in (4.3) is too significant to be ignored. For a detailed discussion about these two methods the reader is referred to [1].

4.2 Trust Region Methods for Unconstrained Optimization

Consider general unconstrained optimization problems of the following form:

$$\min \quad f(x) \tag{4.11}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a nonlinear function that is assumed to be at least twice continuously differentiable. An approximation for the minimizer x^* of (4.11) can be determined by means of the Taylor series expansion around the point x_k

$$f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d \tag{4.12}$$

which is a local quadratic model of $f(x)$. If x_k is in the vicinity of x^*, then Newton type methods are expected to converge with quadratic rate. However, if x_k is far from the solution of (4.11), locally convergent methods can fail to converge as the model (4.12) no longer preserves the curvature characteristics of $f(x)$. Thus, to globalize the convergence of the algorithm, a trust region constraint is added and the trust region problem to be solved at each iteration is:

$$\begin{aligned} \min \quad & m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \\ \text{subject to} \quad & \|d\| \leq \Delta_k \end{aligned} \tag{4.13}$$

where $H_k \in \mathbb{R}^{n \times n}$ is the Hessian $\nabla^2 f(x_k)$ or some approximation to it.

When defining a trust region method, the first issue that deserves attention is the way the trust region radius Δ_k is chosen and how it directly affects the performance of the approach. If the chosen radius is too small, the model accurately approximates the objective function but only small steps can be taken. Otherwise,

if a large radius is chosen, the trust region constraint would not limit the progress of the algorithm. However, the minimizer of the quadratic model may be far from the minimizer of the objective function, thus the radius has to be reduced and a new attempt of the algorithm must be performed.

The most common way of choosing the trust region radius is based on the agreement between the model function $m_k(d)$ and the objective function $f(x_k)$ at the previous iteration. For a given step d_k, we define the ratio

$$\varrho_k = \frac{\text{ar}(d_k)}{\text{pr}(d_k)} = \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d)} \tag{4.14}$$

where the numerator is called the *actual reduction* and the denominator is the *predicted reduction*. Observe that since d_k is obtained by minimizing the model $m_k(d)$ within a region that includes $d = 0$, the predicted reduction will always be nonnegative. Thus, if ϱ_k is negative, the new objective value $f(x_k + d_k)$ is larger than the current value f_k, so the step d_k must be rejected. On the contrary, if ϱ_k is close to 1, there is a good agreement between the model $m_k(d)$ and the function $f(x)$ over this step. In this case, the step d_k is accepted and it is safe to expand the trust region for the next iteration. If ϱ_k is positive but significantly smaller than 1, the trust region is not altered. Additionally, if ϱ_k is close to zero or negative, we shrink the trust region by reducing Δ_k at the next iteration [1].

4.2.1 The Cauchy Point

Although in principle the solution of problem (4.13) is the goal, for global convergence purposes it is sufficient to find an approximate solution d_k that lies within the trust region and gives a *sufficient reduction* in the model. The sufficient reduction can be quantified in terms of the *Cauchy point*, which is simply the minimizer of $m_k(d)$ along the steepest descent direction $-\nabla f(x_k)$, subject to the trust region constraint.

Before analyzing the Cauchy point, it is convenient to define the vector

$$d_k^s = -\Delta_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \tag{4.15}$$

that represents the maximum point along the steepest descent direction subject to

the trust region bound (note that $\|d_k^s\| = \Delta_k$). The Cauchy point is then defined as

$$d_k^c = \tau_k d_k^s \tag{4.16}$$

where $\tau_k \in (0, 1]$ is the minimizer of $m_k(\tau d_k^s)$ constrained to the trust region bound, that is, the solution of the following problem:

$$\begin{aligned} \min \quad & m_k(\tau d_k^s) = f(x_k) + \nabla f(x_k)^T(\tau d_k^s) + \frac{1}{2}(\tau d_k^s)^T H_k(\tau d_k^s) \\ \text{subject to} \quad & \|\tau d_k^s\| \leq \Delta_k \end{aligned} \tag{4.17}$$

The unconstrained minimizer of (4.17) can be determined by taking the first derivative of $m_k(\tau d_k^s)$ with respect to τ, as follows:

$$\frac{d}{d\tau} m_k(\tau d_k^s) = \nabla f(x_k)^T d_k^s + \tau_k {d_k^s}^T H_k d_k^s = 0 \quad \therefore \quad \tau_k = -\frac{\nabla f(x_k)^T d_k^s}{{d_k^s}^T H_k d_k^s} \tag{4.18}$$

By substituting (4.15) into (4.18) and remembering that $\nabla f(x_k)^T \nabla f(x_k) = \|\nabla f(x_k)\|^2$, the result is:

$$\tau_k = \frac{\|\nabla f(x_k)\|^3}{\Delta_k \nabla f(x_k)^T H_k \nabla f(x_k)} \tag{4.19}$$

To clearly understand the effect of the trust region constraint on the objective function of problem (4.17), two curvature cases should be considered. Firstly, if $\nabla f(x_k)^T H_k \nabla f(x_k) \leq 0$, the function $m_k(\tau d_k^s)$ is concave and, therefore, it decreases monotonically with τ from its maximum point at $\nabla f(x_k) = 0$ (see Figure 4.1). Thus, τ_k is simply the largest value that satisfies the trust region bound, that is, $\tau_k = 1$. On the other hand, if $\nabla f(x_k)^T H_k \nabla f(x_k) > 0$, then $m_k(\tau d_k^s)$ is a convex quadratic function in τ, so τ_k is either the unconstrained minimizer (4.19) or the boundary value $\tau_k = 1$ (see Figure 4.2).

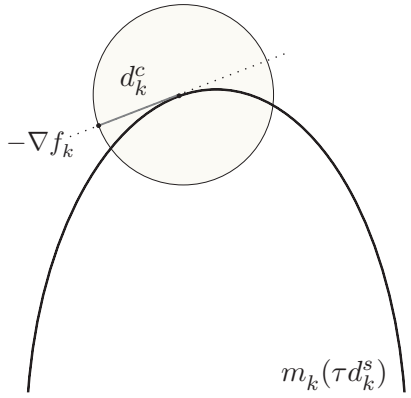

Figure 4.1: Concave function $m_k(\tau d_k^s)$.

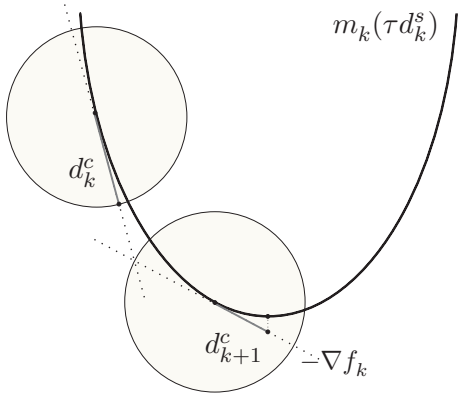

Figure 4.2: Convex function $m_k(\tau d_k^s)$.

In summary, we have

$$d_k^c = \begin{cases} \min\left\{ d_k^s, -\dfrac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T H_k \nabla f(x_k)} \nabla f(x_k) \right\} & \text{, if } \nabla f(x_k)^T H_k \nabla f(x_k) > 0; \\ d_k^s & \text{, otherwise.} \end{cases} \tag{4.20}$$

A trust region method will be globally convergent if the steps d_k attain a sufficient reduction in the model $m_k(d)$ that is at least some fixed multiple of the decrease obtained by the Cauchy step d_k^c at each iteration. Despite of being very simple to implement and inexpensive to calculate, the approximation provided by the Cauchy point is based on the steepest decent method, which implies linear rates of convergence. Thus, in order to achieve faster convergence, practical algorithms start by computing the Cauchy point and then try to improve on it.

According to [1], an improvement strategy is often designed so that the unconstrained minimum $d_k^B = -H_k^{-1}\nabla f(x_k)$ of the quadratic model $m_k(d)$ is used whenever H_k is positive definite and $\|d_k^B\| \leq \Delta_k$. Additionally, when H_k is the exact Hessian or a quasi-Newton approximation, this procedure is expected to achieve superlinear convergence. The next section briefly introduces one of the possible improvements on the Cauchy point, called the *dogleg method* [74].

4.2.2 The Dogleg Method

In order to better understand the dogleg method, it is important to analyze the influence of the trust region radius Δ_k on the solution d^* of (4.13). When the radius is small relative to d_k^B, the constraint $\|d\| \leq \Delta_k$ ensures that the quadratic

term in $m_k(d)$ has little effect on the solution of (4.13). That is, the curvature information of the model function $m_k(d)$ is not very significant when searching within a small vicinity, so a linear approximation to it can be used. In this case, one simple approximation to the step d^* that solves (4.13) is the maximum point along the steepest descent direction subject to the trust region bound, which is calculated by (4.15). On the other hand, when Δ_k assumes intermediate values, the solution d^* typically follows a curved trajectory such as the one illustrated in Figure 4.3.

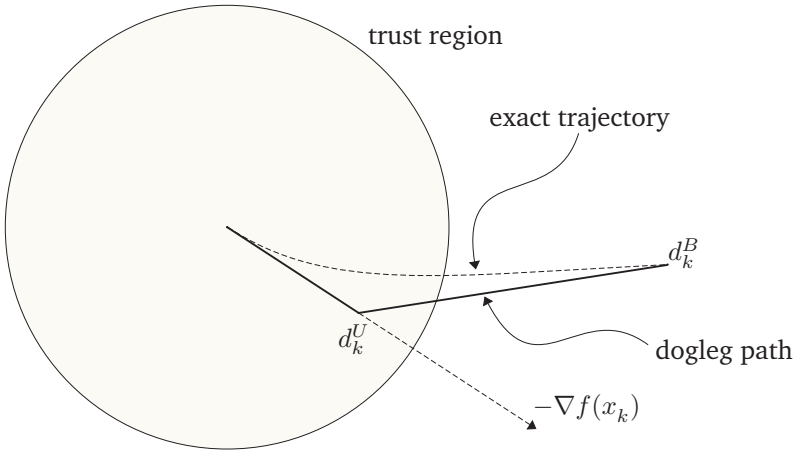

Figure 4.3: Exact trajectory and dogleg path (Adapted from [1]).

In order to find an estimate solution to (4.13), the dogleg method replaces the curved trajectory for d^* with a path consisting of two line segments. The first line segment goes from the current point to the minimizer of $m_k(d)$ along the steepest descent direction, which is given by

$$d_k^U = -\frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T H_k \nabla f(x_k)} \nabla f(x_k) \tag{4.21}$$

The second line segment goes from d_k^U to d_k^B, as illustrated in Figure 4.3. The *dogleg path* can be rigorously denoted by $\widetilde{d}(\tau)$ for $\tau \in [0, 2]$, where

$$\widetilde{d}(\tau) = \begin{cases} \tau d_k^U, & 0 \leq \tau \leq 1 \\ d_k^U + (\tau - 1)\left(d_k^B - d_k^U\right), & 1 \leq \tau \leq 2 \end{cases} \tag{4.22}$$

The dogleg approach chooses d to minimize the model $m_k(d)$ along this path, subject to the trust region constraint.

It is important to observe that $\widetilde{d}(\tau)$ intersects the trust region boundary at ex-

actly one point if and only if $\|d_k^B\| \geq \Delta_k$. In addition, since H_k is supposed to be positive definite, the model $m_k(d)$ is a decreasing function along the dogleg path. Thus, the chosen value of d will be at d_k^B if the unconstrained minimizer is strictly inside the trust region. Otherwise, the value of d is chosen as the point of intersection of the dogleg path and the trust region boundary. In this case, the value of τ is computed by solving the following scalar quadratic equation:

$$\|d_k^U + (\tau - 1)(d_k^B - d_k^U)\|^2 = \Delta_k^2 \tag{4.23}$$

Note that if the exact Hessian $\nabla^2 f(x_k)$ is available and positive definite, one can simply set $H_k = \nabla^2 f(x_k)$ and follow the procedure above to find the Newton-dogleg step.

4.3 Trust Region Methods for Constrained Optimization

In this work, most of the attention is devoted to general nonlinear programming problems of the form (1.1):

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) = 0 \\ & \underline{x} \leq x \leq \overline{x} \end{aligned}$$

Instead of directly solving the nonlinear problem (1.1), trust region methods generate steps with the help of a quadratic model. As seen in previous sections, these methods define a region with radius Δ_k around the current point x_k within which they *trust* the model to be an adequate representation of (1.1). The related trust region problem can be defined as follows:

$$\begin{aligned} \min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d && \text{(4.24a)} \\ \text{subject to} \quad & g(x_k) + \nabla g(x_k)^T d = 0 && \text{(4.24b)} \\ & \underline{x} \leq x_k + d \leq \overline{x} && \text{(4.24c)} \\ & \|d\| \leq \Delta_k && \text{(4.24d)} \end{aligned}$$

where H_k is the Hessian matrix of the Lagrangian function associated with the problem (1.1), that is,

$$H_k = \nabla^2 f(x_k) + \sum_{i=1}^{m} \lambda_i^k \nabla^2 g_i(x_k), \tag{4.25}$$

and λ_i is the Lagrange multiplier associated to the constraint $g_i(x) = 0$.

Problem (4.24) is not a QP problem due to the Euclidean norm used in the trust region constraint. As discussed in [1], there has not been much research on the relative performance of methods that use trust regions of different shapes on large problems. Additionally, other norms such as the ℓ_1 norm and the infinity norm offer no obvious advantages for small-medium unconstrained problems, but they may be useful for constrained ones. For instance, to overcome the difficulty and turn problem (4.24) into a QP problem, Souza et al. [53] used the infinity norm instead of the Euclidean to rewrite the trust region subproblem as follows:

$$\begin{aligned}
\min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d && \text{(4.26a)} \\
\text{subject to} \quad & g(x_k) + \nabla g(x_k)^T d = 0 && \text{(4.26b)} \\
& \underline{x} - x_k \leq d \leq \overline{x} - x_k && \text{(4.26c)} \\
& \|d\|_\infty \leq \Delta_k && \text{(4.26d)}
\end{aligned}$$

As described by [53], since $\|d\|_\infty = \max_i |d_i|$, the constraints (4.26c) and (4.26d) on step d can be combined into a single simple bound constraint, so that the problem (4.26) can be rewritten as:

$$\begin{aligned}
\min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d && \text{(4.27a)} \\
\text{subject to} \quad & g(x_k) + \nabla g(x_k)^T d = 0 && \text{(4.27b)} \\
& \max\{\underline{\delta}, -\Delta_k\} \leq d \leq \min\{\overline{\delta}, \Delta_k\} && \text{(4.27c)}
\end{aligned}$$

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

Problem (4.27) could be directly solved by a general optimization algorithm for QP. However, the trust region constraint (4.27c) could limit the step d in a way such that problem (4.27) becomes infeasible, that is, there may be no step d that satisfies both the equality constraint (4.27b) and the trust region constraint (4.27c). Several approaches were proposed to resolve the possible conflict between satisfying the

linearizations of the original constraints and the additional trust region constraint [41]. The next section presents the Byrd-Omojokun method.

4.3.1 The Byrd-Omojokun Method

In order to resolve possible inconsistencies with constraints (4.27b) and (4.27c), the Byrd-Omojokun method [40] divides problem (4.27) into two subproblems, which are known as *vertical* and *horizontal* subproblems.

4.3.1.1 Vertical Subproblem

The vertical subproblem is defined as follows:

$$\min \quad \frac{1}{2}\|g(x_k) + \nabla g(x_k)^T v\|^2 \tag{4.28a}$$

$$\text{subject to} \quad \max\{\underline{\delta}, -\xi\Delta_k\} \leq v \leq \min\{\overline{\delta}, \xi\Delta_k\} \tag{4.28b}$$

where $\xi \in (0, 1)$ is a reduction factor of the trust region (typically $\xi = 0.8$).

By ignoring the constant term $g(x_k)^T g(x_k)$ in the objective function, problem (4.28) can be rewritten as follows:

$$\min \quad [\nabla g(x_k) g(x_k)]^T v + \frac{1}{2} v^T \nabla g(x_k) \nabla g(x_k)^T v \tag{4.29a}$$

$$\text{subject to} \quad \max\{\underline{\delta}, -\xi\Delta_k\} \leq v \leq \min\{\overline{\delta}, \xi\Delta_k\} \tag{4.29b}$$

The aim 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 (4.27b).

4.3.1.2 Horizontal Subproblem

After solving (4.29) for the step v_k, the full step d_k is obtained by solving the horizontal subproblem:

$$\min \quad \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \tag{4.30a}$$

$$\text{subject to} \quad \nabla g(x_k)^T d = \nabla g(x_k)^T v_k \tag{4.30b}$$

$$\max\{\underline{\delta}, -\Delta_k\} \leq d \leq \min\{\overline{\delta}, \Delta_k\} \tag{4.30c}$$

It must be observed that problem (4.30) is always feasible, once $d = v_k$ is a solution for (4.30b) and $\|v_k\| \leq \xi\Delta_k$.

4.3.1.3 Merit Function

Once d_k has been calculated, a *merit function* is used to decide whether or not this step sufficiently decreases the objective function $f(x)$. Based on [53], the ℓ_2 merit function is considered:

$$\psi(x, \eta) = f(x) + \eta\|g(x)\| \tag{4.31}$$

where $\eta > 0$ is a penalty parameter that weighs the constraint satisfaction against minimization of the objective function. In [41], the author furnishes a strategy to update the value of the penalty parameter. However, some preliminary experiments on OPF problems have shown that a typical value (for example, $\eta = 2$) can be used without significantly changing the performance of the Byrd-Omojokun method.

Given a step d_k, we define the merit function model

$$\widetilde{\psi}(d, \eta) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta\|g(x_k) + \nabla g(x_k)^T d\| \tag{4.32}$$

and calculate the predicted reduction in the merit function (4.31), which is defined as

$$\begin{aligned} \text{pr}(d_k) &= \widetilde{\psi}(0, \eta) - \widetilde{\psi}(d_k, \eta) \\ &= -\nabla f(x_k)^T d_k - \frac{1}{2} d_k^T H_k d_k + \eta(\|g(x_k)\| - \|g(x_k) + \nabla g(x_k)^T v_k\|) \end{aligned} \tag{4.33}$$

Similarly, the actual reduction is

$$\begin{aligned} \text{ar}(d_k) &= \psi(x_k, \eta) - \psi(x_k + d_k, \eta) \\ &= f(x_k) - f(x_k + d_k) + \eta(\|g(x_k)\| - \|g(x_k + d_k)\|) \end{aligned} \tag{4.34}$$

As previously discussed for unconstrained optimization, given a step d_k, we calculate the reduction ratio as follows:

$$\varrho_k = \frac{\text{ar}(d_k)}{\text{pr}(d_k)} = \frac{\psi(x_k, \eta_k) - \psi(x_k + d_k, \eta_k)}{\widetilde{\psi}(0, \eta_k) - \widetilde{\psi}(d_k, \eta_k)} \tag{4.35}$$

Since the merit function (4.31) is nonsmooth and suffers from Maratos effect

[1], it may perform poorly on some problems. However, this can be avoided by computing a *second-order correction* term that is added to d_k, which yields the following trial step:

$$d_{soc} = d_k - \nabla g(x_k) \left(\nabla g(x_k)^T \nabla g(x_k)\right)^{-1} g(x_k + d_k) \tag{4.36}$$

4.3.1.4 Trust Region Radius Update

Based on [41, 53], if $\varrho_k \approx 1$ and, consequently, a step d_k is accepted, the decision of increasing the trust region can be taken as

$$\Delta_{k+1} = \begin{cases} \min\left\{\kappa_1 \|d_k\|_\infty, \bar{\Delta}\right\}, \text{ if } \varrho_k \geq \overline{\varrho}; \\ \min\left\{\kappa_2 \|d_k\|_\infty, \bar{\Delta}\right\}, \text{ if } \underline{\varrho} \leq \varrho_k \leq \overline{\varrho}; \\ \Delta_k, \text{ otherwise}; \end{cases} \tag{4.37}$$

where $\kappa_1 > \kappa_2 > 1$ and the scalars $\underline{\varrho}$ and $\overline{\varrho} \in (0, 1)$ are lower and upper reference values for the ratio ϱ_k, respectively, and $\bar{\Delta}$ is the maximum trust region size. On the other hand, if the step d_k is rejected (that is, $\varrho_k \approx 0$ or negative), we simply shrink the trust region as $\Delta_{k+1} = \gamma \Delta_k$, for $\gamma \in (0, 1)$. Note that, as problem (4.27) was defined using the infinity norm, the definition of (4.37) may use this norm as well. As an alternative to $\|d_k\|_\infty$ in (4.37), the value of the current trust region radius Δ_k can also be used [1].

4.3.2 Sequential ℓ_1 Quadratic Programming

Since (4.27) may have incompatible constraints, another way of attempting to address it is to build the corresponding *exact* ℓ_1 *penalty function* [54] and solve the following modified problem:

$$\begin{aligned} \min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} |g(x_k) + \nabla g(x_k)^T d| \\ \text{subject to} \quad & \max\{\underline{\delta}, -\Delta_k\} \leq d \leq \min\{\overline{\delta}, \Delta_k\} \end{aligned} \tag{4.38}$$

where η_k is a positive and sufficiently large penalty parameter.

The idea of using the exact ℓ_1 penalty function to overcome the possible inconsistencies among the linearized constraints, in the context of the solution of SQP problems with the addition of a trust region, was first suggested by Fletcher [54].

The resulting algorithm is known in the literature as the *sequential* ℓ_1 *quadratic programming* (Sℓ_1QP) approach. According to [1], the difficulties of choosing appropriate values of η_k caused nonsmooth penalty methods to fall out of favor during 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 [2, 56]. As will be further discussed in this work, we propose a simple and effective procedure to update the ℓ_1 penalty parameter based on the values of the problem's Lagrange multipliers.

The objective function in (4.38) is nonsmooth, so it may be difficult to handle the derivative discontinuities in such a minimization. Fortunately, this is not necessary and, as discussed in [1], (4.38) is equivalent to the smooth problem

$$\min \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} (p_i + q_i) \tag{4.39a}$$

$$\text{subject to} \quad g(x_k) + \nabla g(x_k)^T d = p - q, \quad (p, q) \geq 0 \tag{4.39b}$$

$$\max\{\underline{\delta}, -\Delta_k\} \leq d \leq \min\{\overline{\delta}, \Delta_k\} \tag{4.39c}$$

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

Problem (4.39) is feasible because one can always choose a point within the limits (4.39c) and set the pair (p, q) in a way that the constraints (4.39b) and (4.39c) are satisfied. Additionally, to determinate the acceptance of a step d_k, the natural choice is the ℓ_1 merit function

$$\psi(x, \eta) = f(x) + \eta_k \sum_{i=1}^{m} |g_i(x)| \tag{4.40}$$

and the corresponding ℓ_1 merit function model

$$\widetilde{\psi}(d, \eta) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \eta_k \sum_{i=1}^{m} |g(x_k) + \nabla g(x_k)^T d| \tag{4.41}$$

Similarly to the Byrd-Omojokun approach, the predicted and the actual reductions can be defined as functions of the ℓ_1 merit function (4.40) and of its respective model (4.41).

4.4 Final Remarks

This chapter presented the main concepts of trust region methods. Due to the possibility of defining infeasible trust region problems when solving constrained optimization, several approaches have been proposed to handle this difficulty [41]. The Byrd-Omojokun and the Sℓ_1QP are examples of techniques to impose the trust region constraint without generating inconsistent problems. The former method divides the solution into two subproblems, the vertical and the horizontal, that are solved until convergence. The latter approach uses the exact ℓ_1 penalty function to overcome possible inconsistencies among the linearized constraints. These techniques are used in this work, along with the primal-dual IP methods discussed in Chapter 3, to devise globally convergent trust region IP algorithms.

Chapter 5

Trust Region Methods for Improved Performance

THE MAIN contributions of the Thesis are presented in this chapter, in which algorithms based on trust region IP methods are investigated. These algorithms are expected to be globally convergent and to succeed even if the initial point is far from the solution, a situation that can cause locally convergent methods fail to converge. With this aim, two trust region techniques are proposed. Firstly, a modified Byrd-Omojokun approach is detailed. Then, based on the works of Fletcher [54], Gould et al. [56] and Byrd et al. [2], a trust region IP method based on the exact ℓ_1 penalty function is described.

5.1 Modified Byrd-Omojokun Approach

Motivated by the recent works of Souza et al. [53, 75] regarding a trust region Byrd-Omojokum globalization strategy, this section describes some ideas to improve its performance in the context of the solution of trust region subproblems via IP methods. As discussed in [53], both vertical (4.29) and horizontal (4.30) subproblems can be rewritten in the following general form:

$$\begin{aligned} \min \quad & b^T w + \frac{1}{2} w^T A w && \text{(5.1a)} \\ \text{subject to} \quad & M^T w - c = 0 && \text{(5.1b)} \\ & \underline{w} \leq w \leq \overline{w} && \text{(5.1c)} \end{aligned}$$

The vertical subproblem (4.29) is solved by defining $w = v$, $c = []$ and $M = []$, along with

$$\begin{aligned} b &= \nabla g(x_k) g(x_k) && \text{(5.2a)} \\ A &= \nabla g(x_k) \nabla g(x_k)^T && \text{(5.2b)} \\ \underline{w} &= \max\{\underline{\delta}, -\xi \Delta_k\} && \text{(5.2c)} \\ \overline{w} &= \min\{\overline{\delta}, \xi \Delta_k\} && \text{(5.2d)} \end{aligned}$$

Similarly, to solve the horizontal subproblem (4.30), the assumed values are $w = d$ and

$$\begin{aligned} b &= \nabla f(x_k) && \text{(5.3a)} \\ A &= H_k && \text{(5.3b)} \\ c &= \nabla g(x_k)^T v_k && \text{(5.3c)} \\ M &= \nabla g(x_k) && \text{(5.3d)} \\ \underline{w} &= \max\{\underline{\delta}, -\Delta_k\} && \text{(5.3e)} \\ \overline{w} &= \min\{\overline{\delta}, \Delta_k\} && \text{(5.3f)} \end{aligned}$$

Since both vertical and horizontal problems have the same general formulation, very similar optimization routines can be written to solve these problems. Despite of the great diversity of available QP techniques to solve (5.1), the methodologies developed in this work focus on the aspects of the solution via primal-dual IP meth-

ods.

It must be observed that the same IP algorithms described in Chapter 3 for NLP can be used to solve the QP problem (5.1). In fact, the major difference is that the Gradient and Hessian matrices of the nonlinear functions $f(x)$ and $g(x)$ are constant for inner trust region iterations.

5.1.1 Direct Solution of the Trust Region Problem

In order to illustrate the idea, consider the very simple case shown in Figure 5.1, in which $c(x) : \mathbb{R} \rightarrow \mathbb{R}$ is a one-dimensional continuously differentiable function and $m_{x_k}(x) = c(x_k) + c'(x_k)\,(x - x_k)$ is the first order Taylor polynomial approximation of it around x_k.

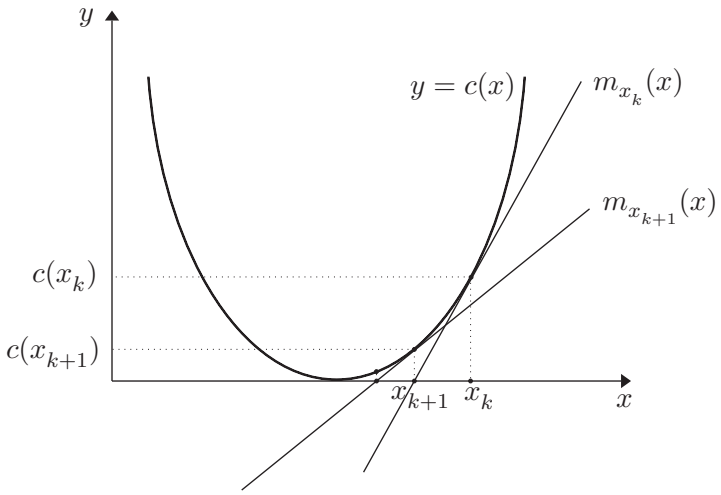

Figure 5.1: Taylor polynomial approximation of $c(x)$ around x_k.

As shown in Figure 5.1, the Taylor linearization $m_{x_k}(x)$ becomes zero at x_{k+1}. However, the value $c(x_{k+1})$ of the function $c(x)$ evaluated at x_{k+1} is still not null. Such as in the Newton's method to find roots of a real-valued function, the error between the function's value $c(x_{k+1})$ and its approximation $m_{x_k}(x_{k+1})$ is $O(d^2)$, in which $d = x_{k+1} - x_k$. Subsequent linearizations $m_{x_{k+1}}(x)$, $m_{x_{k+2}}(x)$, $\ldots$, $m_{x_{k+n}}(x)$ would give smaller errors as d tends to zero and the process converges to the root. A similar procedure is used here to derive the proposed modification on the Byrd-Omojokun method.

When the value of objective function of the vertical subproblem reaches zero, it means that a feasible point $d = v^*$ of (4.27) has been found and that the constraints

(4.27b) and (4.27c) do not define an empty feasible set. However, it is important to observe that $x_{k+1} = x_k + v^*$ is not necessarily a feasible point to the original NLP problem. Firstly, note that from the Taylor series approximation the following equality holds:

$$g(x_{k+1}) = g(x_k) + \nabla g(x_k)^T d + O(\|d\|^2) \tag{5.4}$$

Additionally, considering that $d = v^*$ is feasible for (4.27), then

$$g(x_k) + \nabla g(x_k)^T v^* = 0 \quad \Rightarrow \quad g(x_{k+1}) \text{ is } O(\|v^*\|^2) \tag{5.5}$$

which means that the original nonlinear equality constraint $g(x) = 0$ evaluated at x_{k+1} may be only nearly satisfied.

Assuming that for practical purposes the second order term in (5.5) is negligible, that is, v^* is small and $g(x_{k+1}) = O(\|v^*\|^2) \approx 0$, subsequent Taylor linearizations of the nonlinear constraints around this feasible point will be compatible since $d = 0$ will always be a minimizer of (4.27). Therefore, supposing that from this point on a sequence of feasible points can be found, the trust region problem (4.27) can be directly solved and, by dropping the term $f(x_k)$ of (4.27a), the constants of the QP problem (5.1) are defined as

$$b = \nabla f(x_k) \tag{5.6a}$$
$$A = H_k \tag{5.6b}$$
$$c = -g(x_k) \tag{5.6c}$$
$$M = \nabla g(x_k) \tag{5.6d}$$
$$\underline{w} = \max\{\underline{\delta}, -\Delta_k\} \tag{5.6e}$$
$$\overline{w} = \min\{\overline{\delta}, \Delta_k\} \tag{5.6f}$$

along with $w = d$.

As seen before for NLP, the minimizer of the QP problem (5.1) can be obtained

by primal-dual IP methods by solving the following problem:

$$\min \quad b^T w + \frac{1}{2} w^T A w - \mu_k \sum_{i=1}^{n} (\ln s_i + \ln z_i) \tag{5.7a}$$

$$\text{subject to} \quad M^T w - c = 0 \tag{5.7b}$$

$$\underline{w} - w + s = 0, \quad s > 0 \tag{5.7c}$$

$$w - \overline{w} + z = 0, \quad z > 0 \tag{5.7d}$$

where $s \in \mathbb{R}^n_+$ and $z \in \mathbb{R}^n_+$ are slack variables and μ_k is the barrier parameter.

After all the algebraic manipulations described in Chapter 3, the Newton's system that needs to be solved in each iteration is as follows:

$$\begin{bmatrix} \Pi & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & Z & 0 & 0 \\ I & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 & M^T \\ 0 & 0 & -I & I & M & A \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta \pi \\ \Delta \upsilon \\ \Delta \tau \\ \Delta w \end{bmatrix} = - \begin{bmatrix} S\pi - \mu_k e \\ Z\upsilon - \mu_k e \\ \underline{w} - w + s \\ w - \overline{w} + z \\ M^T w - c \\ Aw + b + M\tau - \pi + \upsilon \end{bmatrix} \tag{5.8}$$

where Υ and Π are diagonal matrices with $\Upsilon_{ii} = \upsilon_i$ and $\Pi_{ii} = \pi_i$, respectively, and $\tau \in \mathbb{R}^m$.

The proposed globally convergent algorithm firstly solves the vertical subproblem using the suggestion by Souza et al. [53], in which $w_0 = 0$. If a feasible point is not found, the horizontal subproblem is started at the minimizer of the vertical subproblem, i.e., $w_0 = v_k$. On the other hand, if the vertical subproblem converges to a minimum equal to zero, then the trust region problem (4.27) is solved instead.

The dimensions of the Newton's systems solved in each iteration of the vertical and the horizontal subproblems are $5n$ and $5n + m$, respectively. Thus, considering that the order of the linear system (5.8) is $5n + m$, the direct solution of the trust region problem can reduce the total algorithm run time up to nearly 50% if a feasible point is found during the solution of the first vertical subproblem [53]. Furthermore, it must be observed that switching from solving alternately the vertical and horizontal subproblems to directly solve the trust region problem is straightforward. Once the vertical objective approaches zero, the horizontal subproblem can be transformed into the trust region problem by simply changing the level of the equality constraints (5.1b) from $c = \nabla g(x_k)^T v_k$ to $c = -g(x_k)$. The modified

Byrd-Omojokun algorithm is summarized in Algorithm 5.1.

1. Set $k = 0$, choose Δ_0, η and ξ.
2. Solve the vertical subproblem for v_k by using (5.2).
3. If $\|g(x_k) + \nabla g(x_k)^T v_k\|^2 \leq \epsilon_v$, then solve the full trust region subproblem by using (5.6). Otherwise, solve the horizontal subproblem by using (5.3).
4. Calculate the reduction ratio ϱ_k by (4.35).
5. If $\varrho_k \geq 0.1$, then update $x_{k+1} = x_k + d_k$ and choose $\Delta_{k+1} \geq \Delta_k$ by (4.37). Otherwise, decrease the trust region radius $\Delta_{k+1} = \gamma \Delta_k$ and set $x_{k+1} = x_k$.
6. If x_{k+1} satisfies the convergence criteria of the original nonlinear problem (1.1) then END. Otherwise, set $k \leftarrow k + 1$ and return to step 2.

Algorithm 5.1: Modified Byrd-Omojokun method to solve (1.1).

5.1.1.1 Tolerance Criteria on Inner Iterations and Early Stop

The globalization strategy used in this work consists in solving a sequence of QP subproblems until the convergence criteria of the original NLP problem have been accomplished. Thus, another way to reduce the total computational effort of the Byrd-Omojokun method could be to approximately solve each QP subproblem, reducing the number of inner trust region iterations and, consequently, the number of matrix factorizations.

According to [76, 77], there is a trade-off between computational cost per subproblem and the number of overall trust region iterations. The more accurate the subproblem solver, the fewer overall iterations required. In addition, for a given optimization problem, an improved performance may be obtained by comparing the cost of evaluating the function and its derivatives against the cost of the linear algebra concerning the resolution of the trust region problem (4.27).

Regarding the solution of OPF problems, the cost of solving the Newton's system is much higher than the cost of evaluating the first and second derivatives. Therefore, numerical simulations are performed to analyze the behavior of the modified Byrd-Omojokun method when using less restrictive tolerance criteria on inner iterations.

5.2 The Sℓ_1QP Method

The main goal of this research is to develop a globally convergent algorithm that is competitive in run time when compared to the Byrd-Omojokun approach. The modified Byrd-Omojokun technique aims at reducing the computational effort by directly solving the trust region problem (4.27) once the vertical objective attains zero. Similarly to the Byrd-Omojokun approach, the $S\ell_1QP$ method [54] can also handle inconsistencies among the linearized constraints in (4.27). This technique uses an exact ℓ_1 penalty function to incorporate the equality constraints (4.27b) into the objective function. The resulting trust region problem (4.38) is a box constrained minimization of a nonsmooth function, which can be transformed into a smooth function by the addition of elastic variables.

This section details the solution of the $S\ell_1QP$ problem (4.39) via primal-dual IP methods for QP. In order to address (4.39), these methods solve the following modified problem:

$$\begin{aligned}
\min \quad & f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d + \\
& + \sum_{i=1}^{m} \left[\eta_k \left(p_i + q_i \right) - \mu_k \left(\ln p_i + \ln q_i \right) \right] - \mu_k \sum_{i=1}^{n} \left(\ln s_i + \ln z_i \right) && \text{(5.9a)} \\
\text{subject to} \quad & g(x_k) + \nabla g(x_k)^T d - p + q = 0, \quad (p, q) > 0 && \text{(5.9b)} \\
& \underline{w} - d + s = 0, \quad s > 0 && \text{(5.9c)} \\
& d - \overline{w} + z = 0, \quad z > 0 && \text{(5.9d)}
\end{aligned}$$

where $\underline{w}$ and $\overline{w}$ are defined by equations (5.3e) and (5.3f), respectively, $s \in \mathbb{R}^n_+$ and $z \in \mathbb{R}^n_+$ are slack variables, $p \in \mathbb{R}^m_+$ and $q \in \mathbb{R}^m_+$ are elastic variables, $\mu_k > 0$ is the barrier parameter and η_k is the ℓ_1 penalty parameter. The strict positivity conditions on the slack variables $(s, z) > 0$ and also on the elastic variables $(p, q) > 0$, are implicitly handled by controlling the step length during the update.

Under the linear independence constraints qualification, if d^* is a local minimum of (5.9), then there are vectors of Lagrange multipliers $\tau^* \in \mathbb{R}^m$, $\pi^* \in \mathbb{R}^n$ and

$\upsilon^* \in \mathbb{R}^n$ that satisfy the KKT first order optimality conditions:

$$\nabla_s L = -\mu_k S^{-1} e + \pi = 0 \tag{5.10a}$$
$$\nabla_z L = -\mu_k Z^{-1} e + \upsilon = 0 \tag{5.10b}$$
$$\nabla_p L = \eta_k e_m - \mu_k P^{-1} e_m - \tau = 0 \tag{5.10c}$$
$$\nabla_q L = \eta_k e_m - \mu_k Q^{-1} e_m + \tau = 0 \tag{5.10d}$$
$$\nabla_\pi L = \underline{w} - d + s = 0 \tag{5.10e}$$
$$\nabla_\upsilon L = d - \overline{w} + z = 0 \tag{5.10f}$$
$$\nabla_\tau L = g(x_k) + \nabla g(x_k)^T d - p + q = 0 \tag{5.10g}$$
$$\nabla_d L = \nabla f(x_k) + H_k d + \nabla g(x_k)\tau - \pi + \upsilon = 0 \tag{5.10h}$$

where P and Q are diagonal matrices with $P_{ii} = p_i$ and $Q_{ii} = q_i$, and $e \in \mathbb{R}^n$ and $e_m \in \mathbb{R}^m$ are column vectors of all ones. The system of equations (5.10) can be rearranged as:

$$r_{s\pi} = S\pi - \mu e = 0 \tag{5.11a}$$
$$r_{z\upsilon} = Z\upsilon - \mu e = 0 \tag{5.11b}$$
$$r_p = P(\eta_k e_m - \tau) - \mu_k e_m = 0 \tag{5.11c}$$
$$r_q = Q(\eta_k e_m + \tau) - \mu_k e_m = 0 \tag{5.11d}$$
$$r_s = \underline{w} - d + s = 0 \tag{5.11e}$$
$$r_z = d - \overline{w} + z = 0 \tag{5.11f}$$
$$r_\tau = g(x_k) + \nabla g(x_k)^T d - p + q = 0 \tag{5.11g}$$
$$r_d = \nabla f(x_k) + H_k d + \nabla g(x_k)\tau - \pi + \upsilon = 0 \tag{5.11h}$$

By applying the Newton's method to (5.11), the following linear system is obtained:

$$\begin{bmatrix} \Pi & 0 & 0 & 0 & S & 0 & 0 & 0 \\ 0 & \Upsilon & 0 & 0 & 0 & Z & 0 & 0 \\ 0 & 0 & \Gamma & 0 & 0 & 0 & -P & 0 \\ 0 & 0 & 0 & \Theta & 0 & 0 & Q & 0 \\ I & 0 & 0 & 0 & 0 & 0 & 0 & -I \\ 0 & I & 0 & 0 & 0 & 0 & 0 & I \\ 0 & 0 & -I & I & 0 & 0 & 0 & \nabla g(x_k)^T \\ 0 & 0 & 0 & 0 & -I & I & \nabla g(x_k) & H_k \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta z \\ \Delta p \\ \Delta q \\ \Delta \pi \\ \Delta \upsilon \\ \Delta \tau \\ \Delta d \end{bmatrix} = - \begin{bmatrix} r_{s\pi} \\ r_{z\upsilon} \\ r_p \\ r_q \\ r_s \\ r_z \\ r_\tau \\ r_d \end{bmatrix} \tag{5.12}$$

where Γ and Θ are diagonal matrices with $\Gamma_{ii} = \eta_k - \tau_i$ and $\Theta_{ii} = \eta_k + \tau_i$, respectively.

The dimensions of the Newton's systems solved in each iteration of the vertical and the horizontal subproblems are $5n$ and $5n + m$, respectively, which nearly corresponds to the cost of a single outer iteration of the Byrd-Omojokun method. On the other hand, the dimension of the linear system (5.12) for the S$\ell_1 QP$ method is $5n + 3m$. Therefore, the linear system (5.12) has extra $2m$ rows and columns when compared to the system solved in the horizontal subproblem in each iteration. However, the S$\ell_1 QP$ approach solves a single linear system whereas the Byrd-Omojokun method solves both the vertical and the horizontal subproblems per iteration. Additionally, it is very important to observe that the coefficient matrix in (5.12) is sparse and the time complexity of its factorization involves factors such as ordering and fill-in [78]. For a good sparse matrix algorithm, the time required for sparse matrix operation depends on and should be proportional to the number of arithmetic operations on nonzero elements.

Despite being slightly larger than the matrices formed for the vertical and horizontal subproblems in the Byrd-Omojokun method, the coefficient matrix in (5.12) has only $6m$ more nonzero elements, which correspond to the diagonal block matrices that come from equalities (5.11c), (5.11d) and (5.11g) for the elastic variables. Thus, considering that in our OPF algorithm the matrix factorizations are handled by the MATLAB's built-in function `mldivide()`, which can efficiently perform operations on sparse matrices, the computational cost of solving (5.12) is expected to be smaller than the cost of solving similar linear systems associated with the vertical and horizontal subproblems in the Byrd-Omojokun technique. Furthermore, the sparsity pattern of the coefficient matrix in (5.12) for the S$\ell_1 QP$ approach is essentially the same of that found in analogous matrices during the solution of the vertical and horizontal subproblems by the primal-dual IP method.

5.2.1 Solving the Reduced System

Alternatively, $\Delta y = (\Delta s, \Delta z, \Delta p, \Delta q, \Delta \pi, \Delta \upsilon, \Delta \tau, \Delta d)$ can be calculated by an equivalent reduced system that is obtained as follows. From the fifth and sixth linear equations in (5.12):

$$\Delta s = \Delta d - \underline{w} + d - s \tag{5.13}$$

$$\Delta z = -\Delta d - d + \overline{w} - z \tag{5.14}$$

By substituting (5.13) and (5.14) in the first and second equations of (5.12):

$$\begin{aligned}\Delta\pi &= -S^{-1}\Pi\Delta d + S^{-1}\left[\mu_k e + \Pi\left(\underline{w} - d\right)\right] \\ &= -S^{-1}\Pi\Delta d + r_{\Delta\pi} \qquad (5.15)\\ \Delta\upsilon &= Z^{-1}\Upsilon\Delta d + Z^{-1}\left[\mu_k e + \Upsilon\left(d - \overline{w}\right)\right] \\ &= Z^{-1}\Upsilon\Delta d + r_{\Delta\upsilon} \qquad (5.16)\end{aligned}$$

By replacing (5.15) and (5.16) in the last equation of (5.12):

$$\left(S^{-1}\Pi + Z^{-1}\Upsilon + H_k\right)\Delta d + \nabla g(x_k)\Delta\tau = -r_{\Delta d}, \qquad (5.17)$$

where $r_{\Delta d} = \nabla f(x_k) + H_k d + \nabla g(x_k)\tau - \pi + \upsilon - r_{\Delta\pi} + r_{\Delta\upsilon}$. Additionally, from the third and fourth equations in (5.12):

$$\Delta p = -p + \Gamma^{-1}\mu_k e + \Gamma^{-1}P\Delta\tau \qquad (5.18)$$

$$\Delta q = -q + \Theta^{-1}\mu_k e - \Theta^{-1}Q\Delta\tau \qquad (5.19)$$

By replacing (5.18) and (5.19) in the seventh equation of (5.12):

$$\nabla g(x_k)^T\Delta d - \left(\Gamma^{-1}P + \Theta^{-1}Q\right)\Delta\tau = -r_{\Delta\tau}, \qquad (5.20)$$

where $r_{\Delta\tau} = g(x_k) + \nabla g(x_k)^T d + \left(\Theta^{-1} - \Gamma^{-1}\right)\mu_k e$.

Finally, using (5.17) and (5.20), the following reduced equivalent system is obtained:

$$\begin{bmatrix} S^{-1}\Pi + Z^{-1}\Upsilon + H_k & \nabla g(x_k) \\ \nabla g(x_k)^T & \Lambda \end{bmatrix}\begin{pmatrix}\Delta d \\ \Delta\tau\end{pmatrix} = -\begin{pmatrix} r_{\Delta d} \\ r_{\Delta\tau}\end{pmatrix} \qquad (5.21)$$

where $\Lambda = -\Gamma^{-1}P - \Theta^{-1}Q$ is a diagonal matrix.

Thus, Δy is obtained by solving (5.21) for Δd and $\Delta\tau$, and then substituting Δd in (5.13), (5.14), (5.15) and (5.16) to get Δs, Δz, $\Delta\pi$ and $\Delta\upsilon$, respectively, along with replacing $\Delta\tau$ in (5.18) and (5.19) to obtain Δp and Δq, in this order.

Analogous algebraic manipulations can be carried out with the equations of the linear system resulting from the horizontal subproblem in the Byrd-Omojokun method to achieve a system of the same dimension of (5.21), as follows:

$$\begin{bmatrix} S^{-1}\Pi + Z^{-1}\Upsilon + H_k & \nabla g(x_k) \\ \nabla g(x_k)^T & 0 \end{bmatrix}\begin{pmatrix}\Delta d \\ \Delta\tau\end{pmatrix} = -\begin{pmatrix} r_{\Delta d} \\ \widetilde{r}_{\Delta\tau}\end{pmatrix} \qquad (5.22)$$

where $\widetilde{r}_{\Delta\tau} = \nabla g(x_k)(d - v_k)$.

Firstly, note that linear system (5.21) has up to m more nonzero elements, which corresponds to the diagonal matrix Λ. It is easy to perceive then, by comparing the coefficient matrices in the equivalent reduced linear systems (5.21) (for the S$\ell_1 QP$ technique) and (5.22) (for the horizontal subproblem of Byrd-Omojokun technique), that the matrix factorization efforts (by far the most time consuming task in an IP algorithm) to solve (5.21) and (5.22) are exactly the same. The minor difference in the processing time consists in the little extra effort to compute Δp and Δq by equations (5.18) and (5.19), respectively, which by involving only simple diagonal matrices and vector operations, is of complexity $O(n)$ operations only. Clearly, one outer iteration of the 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.

5.2.2 Decreasing the Barrier Parameter

Adapting the idea previously discussed in Chapter 3, the complementarity gap can be obtained as follows:

$$\rho_k = s_k^T \pi_k + z_k^T \upsilon_k + p_k^T(\eta_k e_m - \tau_k) + q_k^T(\eta_k e_m + \tau_k) \tag{5.23}$$

The relation between ρ_k and μ_k is implicitly defined in equations (5.11a), (5.11b), (5.11c), (5.11d) and (5.23) in the following form:

$$\sum_{i=1}^{n} s_i \pi_i + \sum_{i=1}^{n} z_i \upsilon_i + \sum_{i=1}^{m} p_i(\eta_k - \tau_i) + \sum_{i=1}^{m} q_i(\eta_k + \tau_i) = (2n + 2m)\,\mu_k = \rho \tag{5.24}$$

from which μ_k can be rewritten as a function of the complementarity gap:

$$\mu_{k+1} = \sigma \frac{\rho_k}{2n + 2m} \tag{5.25}$$

Once the primal and dual steps in the affine-scaling direction are available, the complementarity gap (5.23) can also be used in the predictor-corrector IP method to obtain the approximation μ_{af} for μ_{k+1}.

5.2.3 Initialization and ℓ_1 Penalty Parameter Update

In addition to the standard initialization, the elastic variables p and q are started by expressions based on the complementarity conditions (5.11c) and (5.11d), respectively:

$$p_0^i = \frac{\mu_0}{\eta_0 - \tau_0^i} \tag{5.26}$$

$$q_0^i = \frac{\mu_0}{\eta_0 + \tau_0^i} \tag{5.27}$$

Once the penalty parameter allows only strictly positive values, its initial value η_0 should be different from the initial guess for the Lagrange multipliers τ_0 to avoid ill-posed expressions. According to [1], the ℓ_1 merit function (4.40) has a local minimizer if there is an η^* that is greater than the greatest optimal value for the Lagrange multipliers associated with the equality and inequality constraints of (4.39), that is,

$$\eta^* = \max\{|\tau^*|, \pi^*, \upsilon^*\} \tag{5.28}$$

In this work, the penalty parameter is chosen adaptively as a function of the current value η_k and of the infinity norm of Lagrange multipliers related to the equality constraints. With the exception of the first trust region iteration, in which an user defined value η_0 must be set for the ℓ_1 penalty parameter, each subsequent initialization is performed based on the following heuristic:

$$\eta_k = \max\{\eta_{k-1}, 3\|\tau_{k-1}\|_\infty\} \tag{5.29}$$

Additionally, the penalty parameter is also allowed to vary within the same trust region iteration. In this case, however, only a small increase is permitted, as follows:

$$\eta_{k_j} = \max\{\eta_{k_{j-1}}, \|\tau_{k_{j-1}}\|_\infty\} \tag{5.30}$$

in which the subscript k in η_{k_j} denotes the number of the outer iteration whereas the subscript j denotes the number of the inner iteration.

Although the update strategy of η_k depends on the value of the largest Lagrange multiplier and this may cause numerical instability if the multiplier becomes too large, the practical performance of this strategy on the used class of OPF problems has demonstrated to be fairly good. In addition, as (4.39) is solved by IP methods, the current estimate τ_k for τ^* is readily available. The proposed Sℓ_1QP approach is

summarized in Algorithm 5.2.

1. Do $k = 0$, choose Δ_0, η_0 and ξ.
2. Solve (4.39) for d_k by using the primal-dual IP method and update the ℓ_1 penalty parameter by (5.29) and (5.30).
3. Calculate the reduction ratio ϱ_k by (4.35) with $\psi(x, \eta)$ and $\widetilde{\psi}(d, \eta)$ given by (4.40) and (4.41), respectively.
4. If $\varrho_k \geq 0.1$, then update $x_{k+1} = x_k + d_k$ and choose $\Delta_{k+1} \geq \Delta_k$ by (4.37). Otherwise, decrease the trust region radius $\Delta_{k+1} = \gamma\Delta_k$ and do $x_{k+1} = x_k$.
5. If x_{k+1} satisfy the convergence criteria of the original NL problem (1.1) then END. Otherwise, do $k \leftarrow k + 1$ and return to step 2.

Algorithm 5.2: Proposed Sℓ_1QP method to solve (1.1).

5.3 Final Remarks

This chapter presented some modifications on the Byrd-Omojokun method in order to reduce its computational cost. The objective is to reduce the computation time of the Byrd-Omojokun algorithm proposed in [53]. The main idea is to not solve the vertical subproblem after its objective has become zero, for which can be assumed that the trust region problem is feasible. Hence, from this point on, the modified approach solves directly the trust region problem (4.27).

The dimensions of the Newton's systems solved in each iteration of the vertical and the horizontal subproblems are $5n$ and $5n + m$, respectively. Thus, to directly solve the trust region problem can reduce by nearly half the total algorithm running time. Additionally, it must be observed that the horizontal subproblem (4.30) has the same form of the trust region problem (4.27), so switching between these problems is very simple.

The implementation details of the Sℓ_1QP method was also described. This methodology is able to handle inconsistent constraints and is an alternative to the Byrd-Omojokun method. Additionally, by considering that the linear system (5.12) of order $5n + 3m$ is solved in each iteration, this method can be faster than the Byrd-Omojokun approach for the same number of trust region outer iterations.

Chapter 6

Numerical Experiments

In the previous chapters globally convergent trust region IP methods were presented and the corresponding algorithms were fully described. In this chapter, the main results and a discussion about the application of these methods on OPF problems are given. To test the proposed methodologies, the classical active losses minimization problem is used along with the IEEE test systems up to 300-bus and two distribution networks called REAL-A and REAL-R.

It is important to clarify that the numerical experiments intend to show the convergence characteristics of the two proposed methods: Modified Byrd-Omojokun and Sℓ_1QP. As previously mentioned, both techniques have a reduced time complexity when compared to the Byrd-Omojokun method. Therefore, for one trust region (outer) iteration, these two methodologies present improved performances in terms of algorithm run time.

6.1 Test Systems and General Parameters

The algorithms were implemented using the following IP and trust region general parameters: $\mu_0 = 0.1$, $\gamma = 0.9995$, $\sigma = 0.2$, $\epsilon_1 = 10^{-4}$, $\Delta_0 = 2$, $\Delta_{\max} = 5$, $\xi = 0.8$ and $\eta_0 = 2$. The test systems and the active losses minimization problems dimensions are shown in Table 6.1. The active and reactive systems loads, the initial and minimum active losses and the percentage reduction are presented in Table 6.2.

Table 6.1: Test systems and active losses minimization problems dimensions.

System	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{G}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{T}\rvert$	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
REAL-A	47	1	13	12	120	94
REAL-R	42	1	11	5	101	84

Table 6.2: Active and reactive loads, initial and minimum active losses and percentage reduction.

System	P_L [MW]	Q_L [MW]	Loss_{ini} [MW]	$\text{Loss}_{\min}$ [MW]	$\downarrow$ [%]
IEEE-30	283.40	126.20	17.64	17.34	1.68
IEEE-57	1250.80	336.40	27.86	24.34	12.62
IEEE-118	3668.00	1438.00	132.86	118.03	11.16
IEEE-300	23247.00	7788.00	408.31	378.02	7.42
REAL-A	91.25	38.08	9.65	9.33	3.36
REAL-R	80.59	37.44	2.54	2.52	0.75

6.2 Primal-Dual IP Algorithms

The primal-dual methods described in Chapter 3 are locally convergent and, consequently, are expect to behave poorly when applied to a wide range of initial points. To elucidate the difference between local and global convergence characteristics of the used methods, the simulations were performed with four types of initialization: (I) power flow, (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) 100 random points within limits. The

random points were generated using the MATLAB's built-in function `rand()` according to a uniform distribution. The numerical results for the primal-dual (PD) and primal-dual predictor-corrector (PC) methods considering the first three initializations are shown in Table 6.3.

Table 6.3: Number of iterations for IP methods with three different initializations.

System	(I)		(II)		(III)	
	PD	PC	PD	PC	PD	PC
IEEE-30	12	8	13	10	13	9
IEEE-57	13	8	13	8	12	8
IEEE-118	15	9	19	16	13	8
IEEE-300	18	15	21	17	19	18
REAL-A	14	9	Fail	Fail	27	Fail
REAL-R	12	7	12	9	12	8

Tables 6.4 and 6.5 present a descriptive statistics for the PD and PC methods, respectively, including the number of converged cases (CC), the mode (Mo), the mean ($\overline{k}$) and the standard deviation (σ), along with the minimum ($k_{\min}$) and maximum ($k_{\max}$) number of outer iterations for 100 randomly generated starting points.

Table 6.4: Descriptive statistics for the PD method using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	14	13.56	0.50	13	14
IEEE-57	100	13	13.07	0.26	13	14
IEEE-118	100	15	14.97	0.44	14	16
IEEE-300	80	20	21.27	2.50	18	30
REAL-A	40	24	20.77	3.85	15	29
REAL-R	85	14	16.81	3.13	13	30

Table 6.5: Descriptive statistics for the PC method using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	9	9.68	0.99	8	15
IEEE-57	100	9	9.04	0.82	8	12
IEEE-118	100	10	10.36	0.85	9	14
IEEE-300	67	20	20.46	3.39	15	30
REAL-A	17	11	18.53	6.54	11	30
REAL-R	82	11	15.37	5.27	9	30

For most of the experiments, the PC method produces a reduction in the number

of iterations required to converge. In addition, the results show that both primal-dual techniques respond similarly for initialization (IV). It can be inferred from Tables 6.4 and 6.5 that the test systems IEEE-300, REAL-A and REAL-R are more difficult to solve than the other three IEEE networks. Specifically, the PD and the PC methods presented an inferior performance on the REAL-A test system. It is also important to observe from Tables 6.4 and 6.5 that, besides guaranteeing only local convergence, both primal-dual methods converged for all starting points on IEEE-30, IEEE-57 and IEEE-118 test systems.

The PD and PC methods were also submitted to simulations in which the loads of the test systems were increased in way to obtain feasible, but highly nonlinear, cases. Table 6.6 presents the numerical results.

Table 6.6: Number of iterations for IP methods for increased load cases and initializations (I), (II) and (III).

System	(I)		(II)		(III)		Loss [MW]		↓ [%]
	PD	PC	PD	PC	PD	PC	Ini.	Fin.	
IEEE-30	11	8	13	9	12	7	27.36	26.30	3.87
IEEE-57	14	9	14	9	13	8	60.43	55.36	8.39
IEEE-118	17	14	18	12	16	11	221.77	202.97	8.48
IEEE-300	17	13	21	16	21	12	444.93	409.46	7.97
REAL-A	14	10	Fail	Fail	27	Fail	9.96	9.57	3.92
REAL-R	Fail	Fail	Fail	Fail	Fail	Fail	2.57	2.54 [1]	1.17 [1]

6.3 Trust Region IP Algorithms

This section presents the main results obtained with the developed trust region IP methods. The generated QP problems were solved using both the PD and PC algorithms with equal primal and dual steps. Numerical results obtained with an implementation of the Byrd-Omojokun method derived from [53] are also provided with the objective of comparison with the devised trust region procedures. Besides, except where the contrary is expressly stated, the trust region subproblems are solved by the PD IP method.

[1] Results obtained from the trust region IP algorithms.

6.3.1 Trust Region Problems Solved by the PD Algorithm

Table 6.7 shows the number of outer iterations required until convergence by the Byrd-Omojokun (BO), the Modified Byrd-Omojokun (MBO) and the Sℓ_1QP methods for initializations (I), (II) and (III) using the PD algorithm. The number of the first outer iteration of the MBO algorithm, for which the vertical objective has become nearly zero, is provided within parentheses.

Table 6.7: Number of iterations for initializations (I), (II) and (III) with trust region problems solved by the PD method.

System	(I)			(II)			(III)		
	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP
IEEE-30	5	5(1)	6	4	4(1)	7	6	6(1)	6
IEEE-57	5	5(1)	6	7	7(1)	8	3	3(1)	3
IEEE-118	4	4(1)	4	6	6(2)	6	4	4(2)	6
IEEE-300	10	11(1)	11	11	11(2)	15	16	15(2)	17
REAL-A	4	4(1)	6	14	Fail	7	12	Fail	6
REAL-R	2	2(2)	3	4	4(3)	4	3	3(2)	3

General convergence characteristics of the considered trust region methods can be inferred. Firstly, with the exception of the IEEE-300, the number of trust region iterations does not seem to be influenced by the size of the test systems. In fact, depending on which method is used, the number of outer iterations required by the IEEE-118 can be smaller than the needed by the IEEE-30. Secondly, the BO method performed poorly on the REAL-A test system for initializations (II) and (III), which may explain the failure of convergence for the MBO method. Also, the performance of the Sℓ_1QP technique in terms of number of outer iterations has not significantly changed and the method converged for all three initializations.

Table 6.8 details the convergence process of the MBO algorithm for the IEEE-57 system using flat start. The vertical objective (4.28a) becomes smaller than the tolerance $\epsilon_v = 10^{-9}$ at iteration 1, which indicates that the solution of the first vertical subproblem is a feasible point to (4.27) and the routine switches to the MBO algorithm. Table 6.8 shows the value of the vertical objective evaluated at computed points $d = w_k$ of (4.27). The total algorithm run time was 3.0s instead of the 4.6s taken by the BO method, which represents a reduction of about 35%.

Tables 6.9, 6.10 and 6.11 present a descriptive statistics for the BO, the MBO and the Sℓ_1QP methods, respectively, for 100 randomly generated starting points.

Table 6.8: Convergence process of the MBO method for the IEEE-57 test system with (II).

k	Infeasibility Residuals Primal	Dual	Δ_k	ρ_k	$f(x)$	Vert. Obj.
0	-	-	2	-	0.26	-
1	2.75×10^{-1}	5.17×10^{-3}	2	0.95	24.84	6.38×10^{-13}
2	3.77×10^{-2}	8.33×10^{-4}	2	0.88	24.30	5.11×10^{-32}
3	9.61×10^{-3}	2.76×10^{-3}	2	0.87	24.28	5.14×10^{-34}
4	7.07×10^{-3}	2.41×10^{-3}	2	0.54	24.30	3.85×10^{-33}
5	2.92×10^{-3}	1.01×10^{-3}	2	0.63	24.32	1.48×10^{-33}
6	7.41×10^{-4}	9.69×10^{-5}	2	0.15	24.34	1.45×10^{-34}
7	4.76×10^{-7}	1.77×10^{-6}	2	1.00	24.34	3.06×10^{-37}

Considering the results for the BO technique shown in Table 6.9, almost all runs were well succeeded, which demonstrate the robustness of this methodology. The results for the $S\ell_1QP$ method are also comparable to that accomplished by the BO approach. It can be seen from Table 6.11 that the average number of trust region iterations for the $S\ell_1QP$ method is only slightly higher than that obtained for the BO technique for some test systems. Additionally, the modal values of these two techniques are very close, which indicates that a typical run of the $S\ell_1QP$ method is expected to be faster than the BO approach. Similarly, the MBO method performs as well as the BO method with the exception of the REAL-A test system, for which it converged for only 41 of the 100 starting points. The reasons that lead the MBO method to fail on the REAL-A are discussed later in this chapter with the help of primal-dual logarithmic indicators.

Table 6.9: Descriptive statistics for the BO method with PD algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	4	5.37	2.50	3	11
IEEE-57	100	6	6.56	1.04	4	10
IEEE-118	100	4	4.00	0.00	4	4
IEEE-300	93	15	14.76	2.11	11	24
REAL-A	85	9	11.31	3.38	7	24
REAL-R	100	4	4.07	0.41	3	5

The BO, the MBO and the Sℓ_1QP methods, with the QP problems solved by the PD IP algorithm, were also submitted to increased load simulations. Table 6.12 present the numerical results, in which "$(*)$" means that the MBO algorithm was

Table 6.10: Descriptive statistics for the MBO method with PD algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	4	5.42	2.54	3	11
IEEE-57	100	6	6.52	0.98	4	9
IEEE-118	100	4	4.00	0.00	4	4
IEEE-300	93	15	14.51	1.85	11	21
REAL-A	41	9	10.76	3.57	7	24
REAL-R	100	4	4.13	0.60	3	8

Table 6.11: Descriptive statistics for the $S\ell_1QP$ method with PD algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	4	4.90	1.64	3	9
IEEE-57	100	6	6.88	1.44	5	10
IEEE-118	100	4	4.00	0.00	4	4
IEEE-300	96	17	17.28	1.60	13	22
REAL-A	95	8	8.89	1.59	6	14
REAL-R	100	5	4.53	0.61	3	6

not activated during the iterative process. The number of outer iterations required to convergence has not significantly changed from those shown in Table 6.7, which indicates that the proposed Sℓ_1QP algorithm is not very sensitive to the complexity of the OPF problems. Additionally, despite the fails reported for the Sℓ_1QP the for the REAL-R test system in Table 6.12, a point with primal infeasibilities slightly greater than tolerance criterion $\epsilon_1 = 10^{-4}$ and with final active losses equal to 2.54 MW has been found for each one of the three initializations, which is sufficiently accurate for practical purposes.

Table 6.12: Number of outer iterations for the BO, the MBO and the Sℓ_1QP methods for increased load cases and initializations (I), (II) and (III).

System	(I)			(II)			(III)		
	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP
IEEE-30	4	4(1)	4	4	4(1)	4	4	4(1)	4
IEEE-57	4	4(2)	4	6	6(2)	5	4	4(3)	6
IEEE-118	6	6(3)	6	6	6(3)	8	5	5(3)	5
IEEE-300	10	11(2)	13	14	13(2)	12	13	13(2)	14
REAL-A	4	4(1)	6	14	Fail	7	12	Fail	5
REAL-R	3	3($*$)	Fail	4	4($*$)	Fail	3	3($*$)	Fail

Table 6.13 details the convergence process of the Sℓ_1QP approach for the IEEE-57 system with increased loads and power flow initialization. This case clearly illustrates the update process of the ℓ_1 penalty parameter. The initial value was made equal to $\eta_0 = 2$, slightly increases within the first outer iteration by using (5.30) and at the beginning of the second by using (5.29). Furthermore, from the beginning of the second outer iteration until the process converges, the ℓ_1 penalty parameter remained constant and equal to $\eta_0 = 6.0424$.

Table 6.13: Convergence process of the Sℓ_1QP method for IEEE-57 system with increased loads and initialization (I).

Iter.		Infeasibility Residuals		η_k	ϱ_k	Δ_k	$f(x_k)$
Out.	Inn.	Primal	Dual				
0		-	-	2.0000	-	2	26.25
	1	$4.28\times10^{+1}$	3.31×10^{-2}	2.0000			
	2	$9.66\times10^{+0}$	2.09×10^{-2}	2.0000			
	⋮	⋮	⋮	⋮			
	8	2.25×10^{-4}	3.90×10^{-6}	2.0120			
	9	2.47×10^{-2}	9.49×10^{-7}	2.0132			
	10	2.29×10^{-3}	4.79×10^{-8}	2.0141			
	⋮	⋮	⋮	⋮			
	17	6.66×10^{-9}	1.38×10^{-9}	2.0141			
1		1.72×10^{-1}	3.11×10^{-2}	2.0141	0.94	4	41.98
2		5.27×10^{-2}	1.54×10^{-2}	6.0424	0.83	4	54.46
3		2.08×10^{-3}	5.12×10^{-4}	6.0424	0.97	4	55.33
4		8.06×10^{-5}	1.68×10^{-5}	6.0424	0.98	4	55.36

Figure 6.1 illustrates the variation of the ℓ_1 penalty parameter η_k along with the primal and dual infeasibilities for the IEEE-118 with increased loads and power flow initialization. Similarly to the results obtained for the IEEE-57, the ℓ_1 penalty parameter increases within the first two outer iterations and then remains constant until the process converges.

6.3.2 Trust Region Problems Solved by the PC Algorithm

Table 6.14 presents the number of outer iterations required until convergence by the BO, the MBO and the Sℓ_1QP methods, with the trust region QP problems solved by the PC algorithm and for initializations (I), (II) and (III). The results, in terms of outer number of iterations, are very similar to those obtained by solving the QP subproblems with the PD algorithm. However, when using the PC algorithm, a

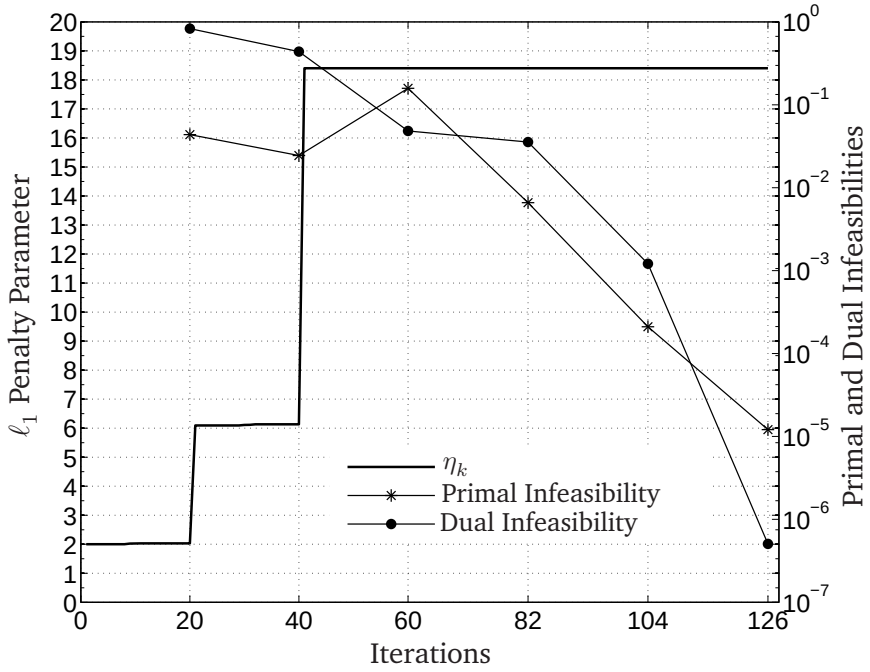

Figure 6.1: Variation of the ℓ_1 penalty parameter along with the primal and dual infeasibilities for IEEE-118 with increased loads.

larger number of failed processes were reported for the IEEE 300-bus. These failures are mostly related to not converged processes, in which the algorithm reaches the maximum number of outer iterations $k_{max} = 30$. For instance, considering the BO method and initialization (II), the iterative process shown in Table 6.15 matches the primal and dual feasibility conditions but does not match the tolerance criterion on the complementarity condition ($\epsilon_2 = 10^{-6}$).

Table 6.14: Number of iterations for initializations (I), (II) and (III) with trust region problems solved by the PC method.

System	(I)			(II)			(III)		
	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP	BO	MBO	Sℓ_1QP
IEEE-30	5	5(1)	5	6	6(1)	6	6	6(1)	4
IEEE-57	6	6(1)	6	7	7(1)	7	5	5(1)	3
IEEE-118	4	4(1)	4	6	6(2)	6	4	4(2)	6
IEEE-300	24	Fail	13	Fail	Fail	15	Fail	Fail	23
REAL-A	6	6(1)	6	18	18(16)	8	13	Fail	Fail
REAL-R	2	2(2)	3	4	4(3)	4	3	3(2)	3

Tables 6.16, 6.17 and 6.18 show a descriptive statistics for the BO, the MBO and the Sℓ_1QP methods, respectively, considering the QP trust region subproblems solved by the PC method and for 100 randomly generated starting points. Anal-

Table 6.15: Failed convergence process for the BO method and IEEE 300-bus system with PC algorithm and initialization (II).

k	Primal Infeas.	Dual Infeas.	Comp. Res.	Δ_k	ϱ_k	$f(x_k)$
0	$1.0000\times10^{+0}$	$1.0000\times10^{+0}$	$1.0000\times10^{+0}$	2.00	0.00	22.6433
1	$3.5119\times10^{+0}$	8.8837×10^{-3}	4.1619×10^{-3}	4.00	0.96	410.8330
2	8.0814×10^{-1}	1.2100×10^{-1}	1.1114×10^{-9}	4.00	0.81	380.6377
3	8.6895×10^{-2}	6.3626×10^{-3}	5.0082×10^{-9}	4.00	0.82	378.0239
4	1.0624×10^{-2}	3.2962×10^{-3}	7.2130×10^{-9}	4.00	0.85	377.9948
5	1.0432×10^{-2}	6.1121×10^{-4}	9.4314×10^{-9}	4.00	0.20	378.0175
6	1.0275×10^{-2}	4.4291×10^{-4}	1.5881×10^{-9}	4.00	0.26	378.0191
7	1.6055×10^{-3}	4.1095×10^{-4}	1.7293×10^{-5}	4.00	0.93	377.9720
8	$5.3474\times10^{+0}$	2.8130×10^{-2}	1.5294×10^{-5}	4.00	524.85	380.4170
9	1.2236×10^{-1}	6.5616×10^{-3}	1.5973×10^{-9}	4.00	0.99	377.7038
10	1.2258×10^{-3}	2.1672×10^{-3}	5.2221×10^{-9}	4.00	0.98	378.0208
11				1.00	-1.65	
12				0.25	-1.65	
13	$5.3515\times10^{+0}$	2.8211×10^{-2}	1.2026×10^{-5}	0.48	560.23	380.4197
14	1.2287×10^{-1}	6.7383×10^{-3}	1.0855×10^{-9}	0.65	0.99	377.7074
15	1.2539×10^{-3}	2.0191×10^{-3}	6.7900×10^{-9}	0.65	0.98	378.0204
16				0.16	-1.34	
17				0.04	-1.38	
18	1.1687×10^{-4}	2.2552×10^{-4}	3.9197×10^{-5}	0.08	0.91	378.0205
19				0.02	-12.17	
20	4.0353×10^{-5}	5.1631×10^{-5}	2.1684×10^{-5}	0.02	0.77	378.0200
21	5.7320×10^{-6}	8.0996×10^{-5}	1.4496×10^{-5}	0.03	0.82	378.0179
22				0.01	-5.11	
23	1.5209×10^{-5}	2.6662×10^{-5}	1.3025×10^{-5}	0.01	0.22	378.0152
24				0.00	0.03	
25	2.9696×10^{-7}	5.3899×10^{-6}	1.6215×10^{-5}	0.00	1.00	378.0190
26	3.5404×10^{-5}	7.9761×10^{-6}	1.3676×10^{-5}	0.01	3.62	378.0193
27	1.7797×10^{-5}	2.8586×10^{-5}	9.5544×10^{-6}	0.01	0.52	378.0143
28	7.3548×10^{-6}	2.2162×10^{-5}	1.0278×10^{-5}	0.01	0.76	378.0185
29				0.00	-1.25	
30	7.4700×10^{-7}	4.3537×10^{-6}	1.0556×10^{-5}	0.01	1.01	378.0192

ogously to the results obtained with the PD method, the modal values found by using the MBO and the Sℓ_1QP methods are quite similar to those ones of the BO technique.

Table 6.16: Descriptive statistics for the BO method with PC algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	4	5.10	1.88	3	11
IEEE-57	100	6	6.52	0.98	4	9
IEEE-118	100	4	4.00	0.00	4	4
IEEE-300	92	13	14.49	2.80	10	30
REAL-A	84	11	12.88	3.35	8	25
REAL-R	98	5	5.45	1.56	3	10

Table 6.17: Descriptive statistics for the MBO method with PC algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	4	5.10	1.88	3	11
IEEE-57	100	6	6.52	0.98	4	9
IEEE-118	100	4	4.00	0.00	4	4
IEEE-300	90	13	14.38	2.33	10	25
REAL-A	45	11	12.40	3.22	8	22
REAL-R	95	5	5.40	1.50	3	10

Table 6.18: Descriptive statistics for the $S\ell_1QP$ method with PC algorithm using initialization (IV).

System	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$
IEEE-30	100	6	5.34	1.02	3	8
IEEE-57	100	6	6.97	1.45	5	10
IEEE-118	100	4	4.01	0.10	4	5
IEEE-300	57	18	20.07	4.03	14	30
REAL-A	91	8	9.77	1.96	7	15
REAL-R	100	5	4.59	0.75	3	7

6.3.3 Performance For Different Sets of Parameters

As discussed in [57], the numerical efficiency of trust region algorithms can be further improved with a better parameter selection. Therefore, it is relevant to examine different sets of parameters for the used class of nonlinear OPF problems. In order to investigate impact of the trust region parameters on the performance of the proposed algorithms, the initial values of the trust region radius Δ_0 and the penalty parameter η_0 were picked from the grid formed by the sets $\{0.5, 1, 2, 4\}$ and $\{0.5, 2, 4, 8\}$, respectively. Additionally, for the BO and MBO algorithms, the value

of the reduction factor of the trust region ξ on the vertical subproblem was made equal to one value of the set $\{0.4, 0.6, 0.8\}$. For each set of parameters and for each test system, 30 random points within limits were used to evaluate the BO, MBO and Sℓ_1QP methods.

The main results for the parametric grid are detailed in Apendix A. Tables A.1, A.2, A.3, A.4, A.5 and A.6 show the main numerical experiments for the BO method. In the same way, Tables A.7, A.8, A.9, A.10, A.11 and A.12 show the main numerical results for the MBO method. Finally, Tables A.13, A.14, A.15, A.16, A.17 and A.18 show the results for the proposed Sℓ_1QP method.

General convergence characteristics of the used trust region IP methods can be gathered from the results of parametric grid. As can be clearly seen from Figure 6.2 and 6.3 for the IEEE-300 test system, the number of CC for the BO and the MBO methods is not very sensitive to changes in the initial trust region radius Δ_0. Both methods are likely to produce good results when Δ_0 values are chosen from $\{0.5, 1, 2\}$. However, for $\Delta_0 = 4$, the techniques may present a higher number of failed processes as the linear and quadratic approximations do not closely represent the related nonlinear functions. Additionally, regarding the initial value for the penalty parameter η_0 on the BO and MBO techniques, the IEEE-57 test system presented higher values for the maximum number of outer iterations for $\eta_0 = 0.5$.

On the other hand, a small Δ_0 appear not to be a good choice fo the Sℓ_1QP. For this method, intermediate values, such as $\Delta_0 = 1$ or $\Delta_0 = 2$, may produce better results in terms of number of CC. Furthermore, the initialization η_0 has no influence on the performance of the Sℓ_1QP method as the algorithm adaptively choose a proper value for the ℓ_1 penalty parameter. Besides, considering the IEEE test systems up to 118-bus, there is only a small influence of different sets of parameters on the performance of the used trust region IP algorithms.

6.3.4 Less Restrictive Tolerance Criteria on Inner Iterations

As previously discussed, another way to reduce the total computational effort of the BO method is to approximately solve each QP subproblem, reducing the overall number of inner iterations and, consequently, the number of matrix factorizations. Therefore, additional simulations were carried out by enlarging 10 times the tolerance criteria on the primal and dual infeasibilities of the QP subproblems, that is, from $\epsilon_1 = 10^{-4}$ to $\epsilon_1 = 10^{-3}$. It is important to clarify that only the tolerance cri-

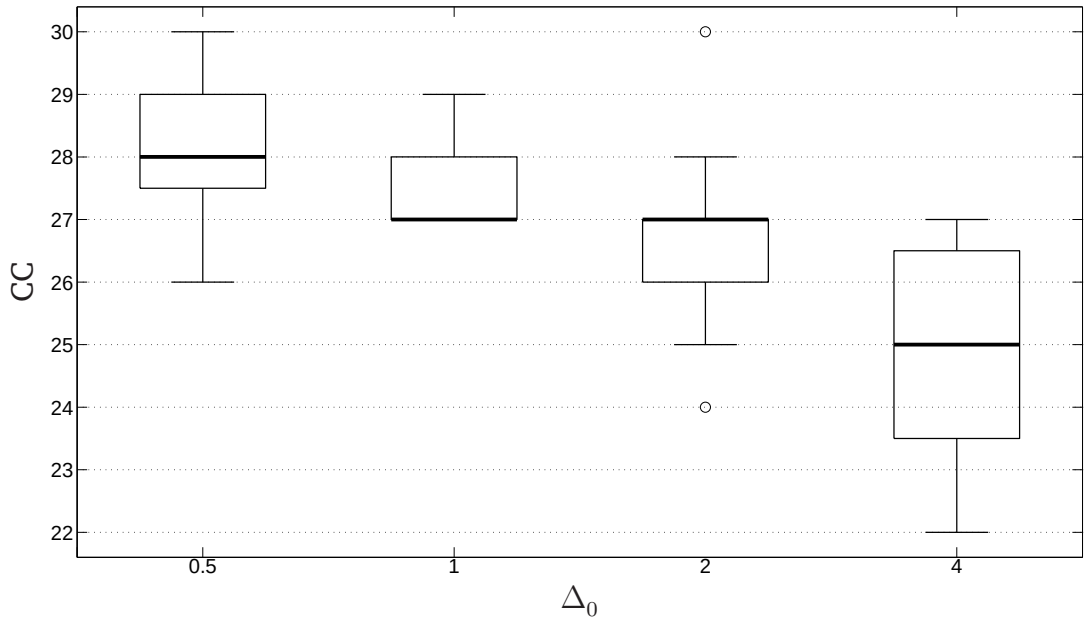

Figure 6.2: Boxplot for initial trust region radius Δ_0 and the number of CC, for BO method and IEEE 300-bus.

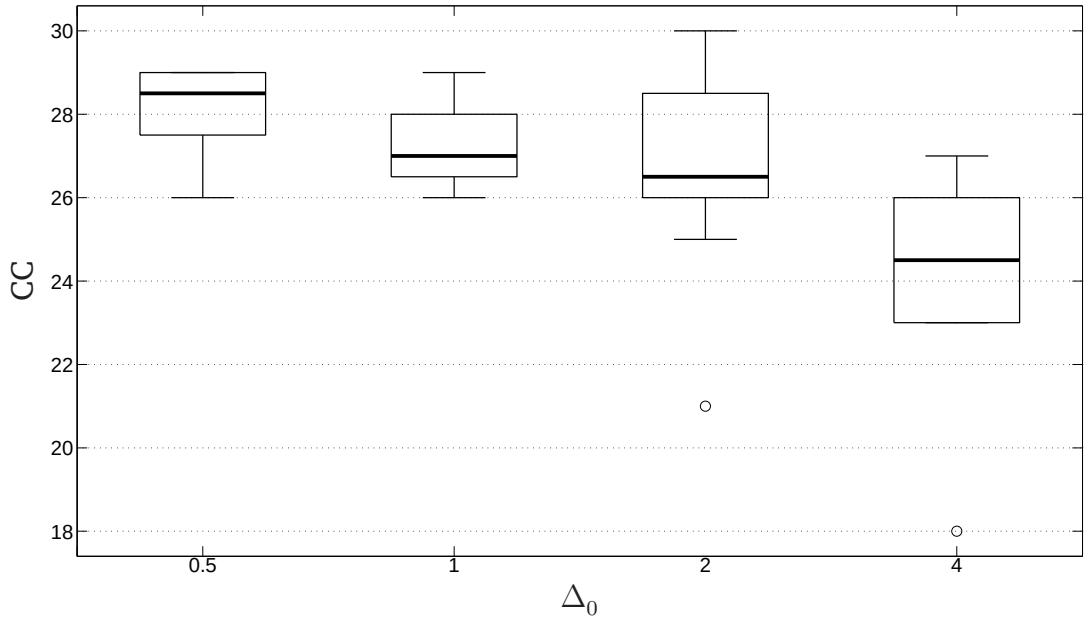

Figure 6.3: Boxplot for initial trust region radius Δ_0 and the number of CC, for MBO method and IEEE 300-bus.

teria on the trust region subproblems were relaxed, the tolerance criteria on outer iterations is still set to $\epsilon_1 = 10^{-4}$. The results for initializations (I), (II) and (III) are shown in Tables 6.19, 6.20 and 6.21, respectively.

Table 6.19: Less restrictive tolerance criterion on inner iterations for BO method and initialization (I).

System	ϵ_1	k	Vertical				Horizontal			
			$j_{\min}$	$\overline{j}$	$j_{\max}$	Total	$j_{\min}$	$\overline{j}$	$j_{\max}$	Total
IEEE-30	0.0001	5	12	12.00	12	60	16	16.60	17	83
	0.001	4	10	10.00	10	40	13	13.25	14	53
IEEE-57	0.0001	5	12	12.00	12	60	15	15.80	16	79
	0.001	4	11	11.00	11	55	13	14.00	15	70
IEEE-118	0.0001	4	13	13.00	13	52	17	17.50	19	70
	0.001	4	11	11.00	11	44	15	15.25	16	61
IEEE-300	0.0001	10	14	14.70	15	147	21	22.40	25	224
	0.001	26	12	13.12	14	341	19	22.04	25	573
REAL-A	0.0001	4	12	13.75	15	55	15	16.75	18	67
	0.001	Fail	-	-	-	-	-	-	-	-
REAL-R	0.0001	2	15	15.50	16	31	15	16.50	18	33
	0.001	Fail	-	-	-	-	-	-	-	-

Table 6.20: Less restrictive tolerance criterion on inner iterations for BO method and initialization (II).

System	ϵ_1	k	Vertical				Horizontal			
			$j_{\min}$	$\overline{j}$	$j_{\max}$	Total	$j_{\min}$	$\overline{j}$	$j_{\max}$	Total
IEEE-30	0.0001	4	11	11.75	12	47	12	15.75	18	63
	0.001	Fail	-	-	-	-	-	-	-	-
IEEE-57	0.0001	7	11	11.86	12	83	14	15.43	16	108
	0.001	Fail	-	-	-	-	-	-	-	-
IEEE-118	0.0001	6	13	14.50	22	87	17	19.00	23	114
	0.001	6	11	12.50	20	75	14	16.83	21	101
IEEE-300	0.0001	11	14	15.27	24	168	19	21.18	23	233
	0.001	Fail	-	-	-	-	-	-	-	-
REAL-A	0.0001	14	12	15.07	18	211	16	18.14	21	254
	0.001	Fail	-	-	-	-	-	-	-	-
REAL-R	0.0001	4	14	15.50	16	62	15	16.25	18	65
	0.001	Fail	-	-	-	-	-	-	-	-

Significant reductions in the number of inner iterations were achieved for IEEE-30, IEEE-57 and IEEE-118 test systems when using initializations (I) and (III). Furthermore, unexpected reductions in the number of outer iterations were also ob-

Table 6.21: Less restrictive tolerance criterion on inner iterations for BO method and initialization (III).

System	ϵ_1	k	Vertical				Horizontal			
			$j_{\min}$	$\overline{j}$	$j_{\max}$	Total	$j_{\min}$	$\overline{j}$	$j_{\max}$	Total
IEEE-30	0.0001	6	12	12.00	12	72	16	16.33	17	98
	0.001	4	10	10.00	10	40	13	13.00	13	52
IEEE-57	0.0001	3	12	12.00	12	36	16	16.00	16	48
	0.001	3	11	11.00	11	33	13	13.67	14	41
IEEE-118	0.0001	4	13	14.50	19	58	17	18.00	19	72
	0.001	4	11	12.25	16	49	15	15.25	16	61
IEEE-300	0.0001	16	14	15.38	26	246	17	23.19	30	371
	0.001	Fail	-	-	-	-	-	-	-	-
REAL-A	0.0001	12	13	14.58	17	175	16	18.50	21	222
	0.001	Fail	-	-	-	-	-	-	-	-
REAL-R	0.0001	3	16	16.00	16	48	15	15.67	17	47
	0.001	Fail	-	-	-	-	-	-	-	-

tained for the IEEE-30 and IEEE-57. Despite the fairly good results, the use of less restrictive tolerance criteria does not lead to convergence for all test systems and initializations. For instance, by considering the results for the IEEE-300 test system, a larger tolerance criteria caused an increase in the number of both inner and outer overall iterations. Similarly, the approximations generated in the QP subproblems for the REAL-A and REAL-R test systems were not sufficient to match the required tolerance $\epsilon_1 = 10^{-4}$ on outer iterations.

6.3.5 Monitoring by Primal-Dual Logarithmic Indicators

As shown before, the primal-dual logarithmic indicators can be used to monitor whether a problem is infeasible or not by gathering information from the slack variables and Lagrange multipliers associated with the simple bound constraints. This characteristic is very useful to investigate the failure of convergence of the proposed trust region IP algorithms as one can analyse the convergence process of the generated QP subproblems.

For instance, consider the solution of the REAL-A test system by the MBO algorithm and using flat start initialization shown in Table 6.7. This actual subtransmission system is mainly radial and it usually operates with low voltages on its ending branches, leading to OPF problems in which some lower bounds are almost con-

tinuously reached depending on the starting points. Furthermore, the assumptions regarding subsequent linearizations of the nonlinear constraints in the MBO procedure may not hold when the problem is highly nonlinear and the second order term in (5.4) is not neglectable, and the method may generate infeasible trust region subproblems. Figure 6.4 shows the logarithmic indicators profile for the second trust region subproblem solved by the MBO algorithm. The two sets of variables, corresponding to active and inactive simple bound constraints, can be clearly seen when the algorithm converges. Additionally, by analysing the way the indicators evolve during the iterative process, one can infer about the centrality of the steps generated by the IP method.

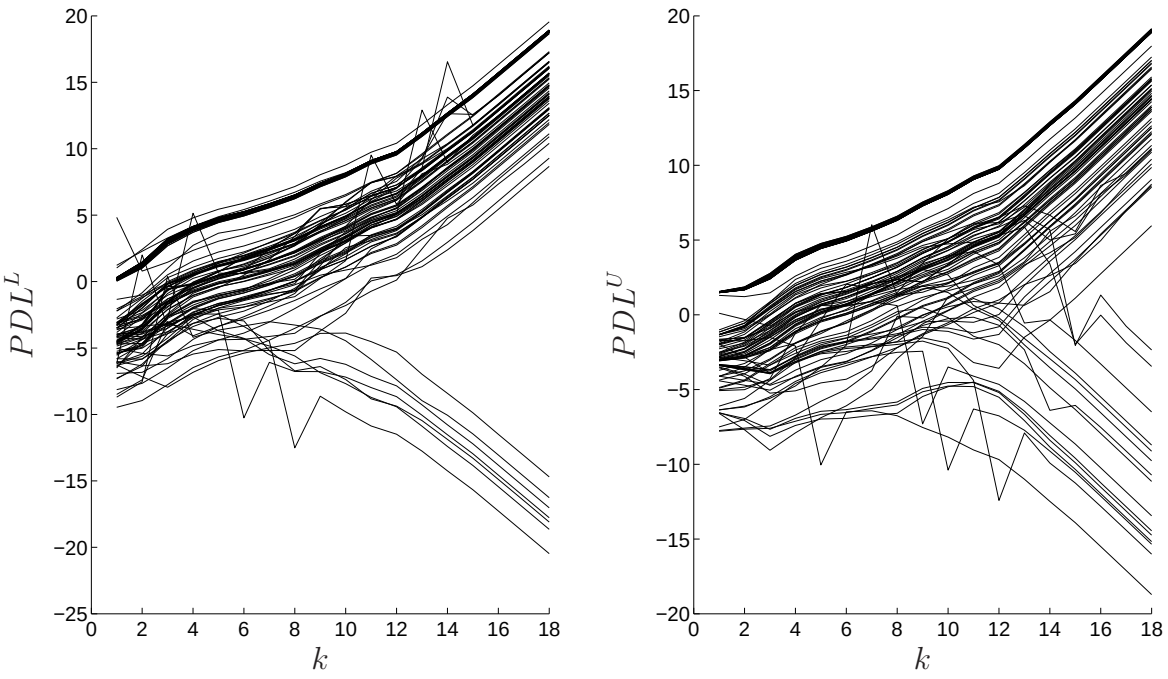

Figure 6.4: Logarithmic indicators profile for a feasible trust region subproblem.

On the other hand, Figure 6.5 presents the logarithmic indicators profile for the third trust region subproblem, for which the MBO method fails to converge. It can be inferred from the the lower and upper bound logarithmic indicators that the problem is infeasible or nearly infeasible. As discussed in Chapter 3, in the context of the solution by primal-dual IP methods, an infeasible simple bound can cause its respective slack variable to rapidly decrease to zero (limited by the logarithmic barrier) in an attempt to become negative. As a direct consequence of very small slack variables, the primal step calculated from (3.9) tends to zero and the algorithm fails

to converge. Meanwhile, the corresponding Lagrange multiplier increases very fast, providing the sensitivity that an active constraint has been found. It is importante to observe that Figure 6.5 also captures the effect of very small steps on variables that are strictly inside the feasible region, resulting in lines that are almost straight from the beginning of the iterative process until the end.

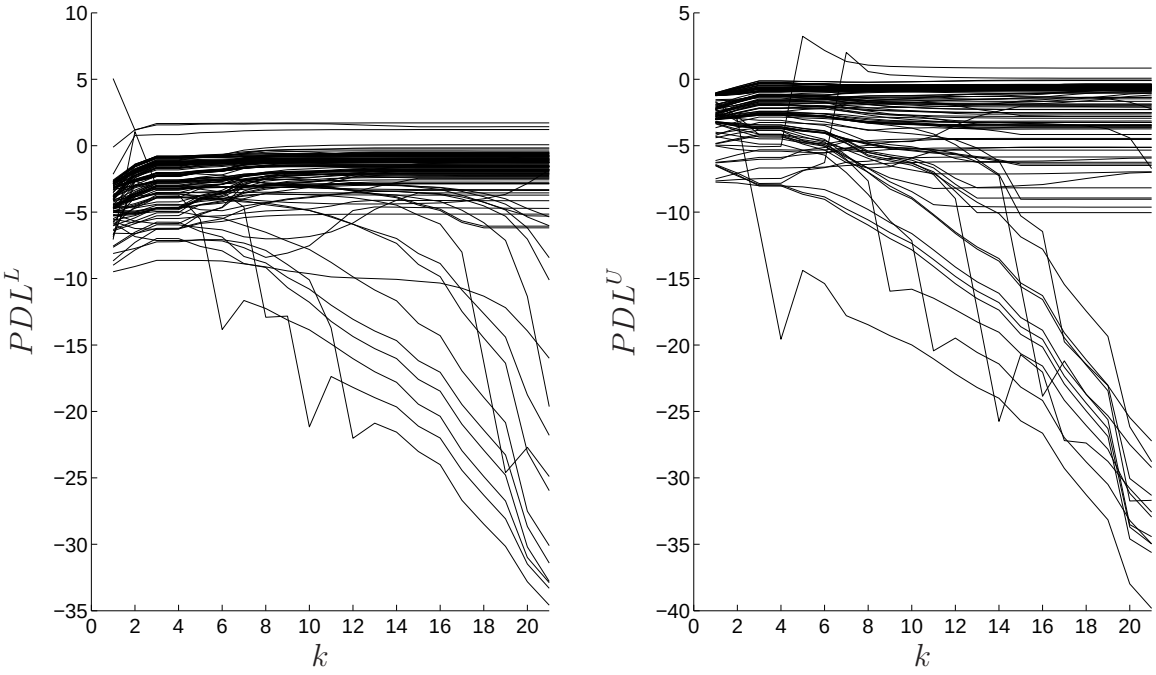

Figure 6.5: Logarithmic indicators profile for an infeasible trust region subproblem.

6.4 Final Remarks

This chapter presented the preliminary numerical experiments for the proposed trust region techniques. The results considering random initialization indicate that the MBO approach may experience some difficulties to handle highly nonlinear problems, for which the assumptions regarding the direct solution of the trust region problem may not hold. However, for almost all the tested OPF problems, the performance of the MBO method is similar to that accomplished by the BO method.

The simulations with the $S\ell_1QP$ method indicate that it is competitive when compared to the BO method. In each iteration, instead of dividing the solution of the trust region problem into two subproblems as the BO method does, the $S\ell_1QP$

technique solves just one slightly larger problem, which invariably takes less time. The simulations carried out with different types of initializations have shown that the proposed method is robust and can handle a wide range of starting points. Additionally, the computational experiments on the used class of OPF problems indicate that the procedure to update the ℓ_1 penalty parameter can properly estimate a value for the ℓ_1 penalty parameter during the iterative process.

Chapter 7

Conclusions

THERE ARE various reasons that can cause optimization methods fail to converge. For instance, locally convergent methods can fail to converge when the initial estimate is far from the solution. Line search and trust region are two important descent strategies for guaranteeing global convergence [1]. This work has focused on the application of trust region methods, which have been used to provide global convergence to a wide range of algorithms from unconstrained to constrained optimization. Particularly, trust region techniques can be combined with well-established IP methods to devise more robust algorithms, which can match the requirements of real-time power systems applications.

7.1 Summary and Contributions

Despite the recent advances regarding the application of the BO trust region method to the solution of OPF problems [53], the performance of this trust region algorithm can be further improved. In this way, this work presented the development of two trust region IP methods: the MBO and the Sℓ_1QP. The former is derived from the BO algorithm described in [53] and tries to improve on it. Under certain assumptions, the trust region problem is directly solved instead of the pair vertical and horizontal subproblems. By doing this, the total algorithm run time can be reduced by nearly half. Numerical experiments have shown a slightly inferior performance of the proposed MBO algorithm when compared to that discussed in [53]. It is important to observe that the modal values are very close to that accomplished by the BO approach, which indicates that the assumptions made to directly solve the trust region problem do not necessarily cause an increase in the number of outer iterations.

The Sℓ_1QP method uses the exact ℓ_1 penalty function to overcome possible inconsistencies among the linearized constraints. The present work described the development of this method along with the main steps of the solution of the derived trust region problems by primal-dual IP methods. Additionally, based on a simple simple heuristic, a practical procedure to update the ℓ_1 penalty parameter is proposed. Numerical experiments for OPF problems have demonstrated it is as robust as the BO strategy. The number of converged cases for randomly generated starting points are comparable to that achieved by the BO method. Additionally, the number of linear equations solved in each trust region outer iteration is smaller than the solved by the BO approach. Therefore, for the tested OPF problems, the Sℓ_1QP method has been proved to be faster than the BO method.

Another relevant aspect when solving an optimization problem is to efficiently detect and handle infeasibility. Infeasible simple bounds can be identified by using logarithmic primal-dual indicators [68]. These indicators are functions of the slack variables and the Lagrange multipliers, which are readily available in IP routines. When using these indicators, it is of great importance to observe that the values assumed for weakly active constraints during well-centered iterates are much different from that found for infeasible bounds, which should be interpreted as strong active constraints.

The MBO and the Sℓ_1QP methods were tested with different sets of trust region

parameters, including those ones given by the literature. Additionally, computational experiments were carried out with a looser tolerance criteria on the vertical and horizontal subproblems so as to reduce the computational effort of the proposed methods. Despite the fairly good results, the use of a looser tolerance criteria on inner iterations has not produced the expected reduction on the algorithm run time for all test systems.

The difficulties found during the development of the Thesis can be sumarized as follows:

- The MBO approach can only be applied under certain assumptions concerning the identification of a feasible point for the original nonlinar problem. These assumptions may not be valid for higly nonlinear systems, for which the second order term in the Taylor expansion is not negligible.
- The value of the ℓ_1 penalty parameter is calculated by a simple heuristic approach that uses the current estimate of the Lagrange multipliers. This parameter has a great influence on the performance of the Sℓ_1QP method, so a more sophisticated update methodology could be used to obtain better results.

7.2 Perspectives for Future Research

During the development of the Thesis some ideas to further improve the performance of the related trust region IP methods came up. We consider that the most exciting and prominent research in this field can be summarized as follows:

- To fully link the infeasibility treatment and monitoring routines to the trust region QP subproblems. Instead of ending a process that is likely to fail, these routines would naturally allow and handle inconsistent constraints in the trust region subproblems in the early stage of convergence.
- To consider scaling matrices and other norms on the used trust region IP methods.
- To use quasi-Newton approximations to the Hessian H_k in the used trust region IP methods in order to improve their computational performance.

- To apply the gradient projection method to the solution of the vertical subproblem in the BO method.
- To develop algorithms to define and efficiently implement the optimal sequence of controls provided by an OPF solution.

References

[1] NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. [S.l.]: Springer, 2006.

[2] BYRD, R. H.; NOCEDAL, J.; WALTZ, R. A. *Steering Exact Penalty Methods for Nonlinear Programming*. [S.l.], 2007.

[3] DOMMEL, H. W.; TINNEY, W. F. Optimal power flow solution. *AIEEE*, PAS-87, p. 1866–1876, October 1968.

[4] ALSAC, O.; STOTT, B. Optimal load flow with steady-state security. *Power Apparatus and Systems, IEEE Transactions on*, PAS-93, n. 3, p. 745–751, May 1974.

[5] SUN, D. I.; ASHLEY, B.; BREWER, B.; HUGHES, A.; TINNEY, W. F. Optimal power flow by newton approach. *Power Apparatus and Systems, IEEE Transactions on*, PAS-103, n. 10, p. 2864–2880, October 1984.

[6] STOTT, B.; ALSAC, O.; MONTICELLI, A. J. Security analysis and optimization. *Proceedings of the IEEE*, v. 75, n. 12, p. 1623–1644, December 1987.

[7] TINNEY, W. F.; BRIGHT, J. M.; DEMAREE, K. D.; HUGHES, B. A. Some deficiencies in optimal power flow. *IEEE Transactions on Power Systems*, v. 3, n. 2, p. 676–683, May 1988.

[8] GRANVILLE, S. Optimal reactive dispatch through interior point methods. *IEEE Transactions on Power Systems*, v. 9, n. 1, p. 136–146, February 1994.

[9] CAPITANESCU, F. *et al.* State-of-the-art, challenges, and future trends in security constrained optimal power flow. *Electric Power Systems Research*, Elsevier Science, v. 81, n. 8, p. 1731–1741, 2011.

[10] CARPENTIER, J. Optimal power flows. *IPC Business Press*, v. 1, n. 1, p. 1–13, April 1979.

[11] CARPENTIER, J. Contribuition à l'étude du dispaching économique. *Bulletin de la Societé Françoice des Electriciens,SER-8*, v. 3, p. 432–447, August 1962.

[12] AOKI, K.; FAN, M.; NISHIKORI, A. Optimal var planning by approximation method for recursive mixed-integer linear programming. *IEEE Transactions on Power Systems*, v. 3, n. 4, p. 1741–1747, November 1988.

[13] PAPALEXOPOULOS, A. D.; IMPARATO, C. F.; WU, F. F. Large-scale optimal power flow: Effects of initialization, decoupling & discretization. *IEEE Transactions on Power Systems*, v. 4, n. 2, p. 748–759, May 1989.

[14] LIU, W.-H. E.; PAPALEXOPOULOS, A. D.; TINNEY, W. F. Discrete shunt controls in a Newton optimal power flow. *IEEE Transactions on Power Systems*, v. 7, n. 4, p. 1509–1518, November 1992.

[15] LIU, W.-H. E.; PAPALEXOPOULOS, A. D.; BRIGHT, J. M. Discrete shunt device based voltage control in an adjusted power flow solution. *Proceedings of 11th PSCC*, August 1993.

[16] DING, Q.; LI, N.; WANG, X. Implementation of interior point method based voltage/reactive power optimization. *Proc. of the IEEE PES General Meeting*, June 1996.

[17] SHARIF, S. S.; TAYLOR, J. H. Minlp formulation of optimal reactive power flow. In: *Proceedings of the America Control Conference*. [S.l.: s.n.], 1997. v. 3, p. 1974–1978.

[18] LIU, M.; TSO, S. K.; CHENG, Y. An extended nonlinear primal-dual interior-point algorithm for reactive-power optimization of large-scale power systems with discrete control variables. *IEEE Transactions on Power Systems*, v. 17, n. 4, p. 982–991, November 2002.

[19] YAN, W.; LIU, F.; CHUNG, C. Y.; WONG, K. P. A hybrid genetic algorithm-interior point method for optimal reactive power flow. *IEEE Transactions on Power Systems*, v. 21, n. 3, p. 1163–1169, August 2006.

[20] XIAOYING, D.; XIFAN, W.; LIN, L. Interior point cutting plane method for discrete decoupled optimal power flow. *IAENG International Journal of Applied Mathematics*, August 2007.

[21] SIMONI, V. R. *Tratamento de Requisitos para Uso de Fluxo de Potência Ótimo em Tempo Real*. Dissertao (Master Thesis) — Universidade Federal de Pernambuco, Brasil, 2010.

[22] CAPITANESCU, F.; WEHENKEL, L. Sensitivity-based approaches for handling discrete variables in optimal power flow computations. *IEEE Transactions on Power Systems*, v. 25, n. 4, p. 1780 –1789, November 2010.

[23] CHANG, S.-K.; MARKS, G. E.; KATO, K. Optimal real-time voltage control. *IEEE Transactions on Power Systems*, v. 5, n. 3, p. 750–758, August 1990.

[24] SOMAN, S.; PARTHASARATHY, K.; THUKARAM, D. Curtailed number and reduced controller movement optimization algorithms for real time voltage/reactive power control. *IEEE Transactions on Power Systems*, v. 9, n. 4, p. 2035 –2041, November 1994.

[25] TORRES, G. L.; CAÑIZARES, C. A.; SIMONI, V. R. Supressão de ajustes ineficazes na solução de fluxo de potência ótimo. In: *Anais do III Simpósio Brasileiro de Sistemas Elétricos*. Belém, PA, Brasil: [s.n.], 2010. v. 1, n. ISSN 2177-6164, p. 1–7.

[26] CAPITANESCU, F.; WEHENKEL, L. Optimal power flow computations with a limited number of controls allowed to move. *IEEE Transactions on Power Systems*, v. 25, n. 1, p. 586 –587, February 2010.

[27] CAPITANESCU, F.; WEHENKEL, L. Redispatching active and reactive powers using a limited number of control actions. *IEEE Transactions on Power Systems*, v. 26, n. 3, p. 1221 –1230, August 2011.

[28] MONTICELLI, A. J.; LIU, W. E. Adaptive movement penalty method for the Newton optimal power flow. *IEEE Transactions on Power Systems*, v. 7, n. 1, p. 334–342, February 1992.

[29] SIMONI, V. R.; TORRES, G. L.; CARVALHO, M. A. de; FREITAS, F. E. F.; FONTE, L. A. M. da. Modeling and analysis of naturally saturated reactors in optimal power flow studies. In: *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*. [S.l.: s.n.], 2011. p. 1–8.

[30] MONTICELLI, A. J. *Fluxo de carga em redes de energia elétrica*. [S.l.]: São Paulo, Edgard Blücher, 1983.

[31] TORRES, G. L.; QUINTANA, V. H. A jacobian smoothing nonlinear complementarity method for solving nonlinear optimal power flows. In: *Proc. of the 14th PSCC*. Sevilla, Spain: [s.n.], 2002.

[32] BORGES, S. S.; FERNANDES, T. S. P.; ALMEIDA, K. C. d. Pré-despacho hidrotérmico de potência ativa e reativa via método dos pontos interiores e coordenadas retangulares. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, v. 22, p. 479 – 494, Outubro 2011.

[33] WRIGHT, S. J. *Primal-Dual Interior-Point Methods*. [S.l.]: SIAM, 1997.

[34] MEHROTRA, S. On the implementation of a primal-dual interior point method. *SIAM J. on Optimization*, n. 2, p. 575–601, 1992.

[35] GONDZIO, J. Multiple centrality correntions in a primal-dual method for liner programming. *Computacional Optimization and Applications*, v. 6, p. 137–156, 1996.

[36] TORRES, G. L.; QUINTANA, V. H. On a nonlinear multiple-centrality-corrections interior-point method for optimal power flow. *IEEE Transactions on Power Systems*, v. 16, n. 2, p. 222–228, May 2001.

[37] ALMEIDA, K. C.; GALIANA, F. D. Critical cases in the optimal power flow. *IEEE Transactions on Power Systems*, v. 11, n. 3, p. 1509–1518, August 1996.

[38] DENNIS, J. E.; SCHNABEL, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. [S.l.]: Society for Industrial and Applied Mathematics, 1996.

[39] CONN, A. R.; GOULD, N. I. M.; TOINT, P. L. *Trust Region Methods*. [S.l.]: Society for Industrial and Applied Mathematics, 2000.

[40] BYRD, R. H.; GILBERT, J. C.; NOCEDAL, J. *A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming*. Optimization Technology Center, Evanston, USA, May 1996. v. 1, n. 2896, 38 p.

[41] PLANTENGA, T. D. *Large-scale nonlinear constrained optimization using trust regions*. Tese (PhD Thesis) — Northwestern University, Evanston, Illinois, United States of America, 1994.

[42] PAJIĆ, S.; CLEMENTS, K. Globally convergent state estimation via the trust region method. In: *Power Tech Conference Proceedings, 2003 IEEE Bologna*. [S.l.: s.n.], 2003. v. 1, p. 6.

[43] PAJIĆ, S. *Power System State Estimation and Contingency Constrained Optimal Power Flow - A Numerically Robust Implementation*. Tese (PhD Thesis) — Worcester Polytechnic Institute, United States of America, 2007.

[44] HASSAÏNE, Y.; DELOURME, B.; HAUSHEER, P.; PANCIATICI, P.; WALTER, E. M-arctan estimator based on the trust-region method. In: *Power Systems Computation Conference*. Liège, Belgium: [s.n.], 2005. v. 1, p. 1 – 7.

[45] COSTA, A.; SALGADO, R.; HAAS, P. Globally convergent state estimation based on givens rotations. In: *Bulk Power System Dynamics and Control - VII. Revitalizing Operational Reliability, 2007 iREP Symposium*. [S.l.: s.n.], 2007. p. 1 –9.

[46] MIN, W.; SHENGSONG, L. A trust region interior point algorithm for optimal power flow problems. *International Journal of Electrical Power & Energy Systems*, v. 27, n. 4, p. 293–300, May 2005.

[47] WANG, H.; MURILLO-SANCHEZ, C.; ZIMMERMAN, R.; THOMAS, R. On computational issues of market-based optimal power flow. *Power Systems, IEEE Transactions on*, v. 22, n. 3, p. 1185 –1193, August 2007.

[48] GIACOMONI, A.; WOLLENBERG, B. Linear programming optimal power flow utilizing a trust region method. In: *North American Power Symposium (NAPS), 2010*. [S.l.: s.n.], 2010. p. 1 –6.

[49] EL-SOBKY, B.; ABO-ELNAGA, Y.; ZAHED, H. Utilization of trust region algorithm in solving reactive power compensation problem. *Applied Mathematical Sciences*, v. 6, n. 54, p. 2649 – 2667, 2012.

[50] BISHEH, H. M.; KIAN, A. R.; ESFAHANI, M. M. S. Solving environmental/economic power dispatch problem by a trust region based augmented lagrangian method. *Iranian Journal of Electrical & Electronic Engineering*, v. 8, n. 2, 2012.

[51] BIEHL, S. V. *Uma Nova Abordagem para Resolução de Problemas de Fluxo de Carga com Variáveis Discretas*. Tese (Doutorado) — Universidade de São Paulo, Brasil, 2012.

[52] TORRES, G. L. Globally convergent optimal power flow using complementarity functions and trust region methods. In: *Power Systems Computation Conference*. Stockholm, Sweden: [s.n.], 2011. v. 1, p. 1 – 8.

[53] SOUZA, A. A.; TORRES, G. L.; CAÑIZARES, C. A. Robust optimal power flow solution using trust region and interior-point methods. *Power Systems, IEEE Transactions on*, v. 26, n. 2, p. 487 –499, May 2011.

[54] FLETCHER, R. *Practical Methods of Optimization*. [S.l.]: JWS, 1987.

[55] FLETCHER, R.; DEMYANOV, V. F.; BOMZE, I. M.; TERLAKY, T. The sequential quadratic programming method. In: *Nonlinear Optimization*. [S.l.]: Springer Berlin / Heidelberg, 2010, (Lecture Notes in Mathematics, v. 1989). p. 165–214.

[56] GOULD, N. I. M.; ORBAN, D.; TOINT, P. L. *An interior-point ℓ_1-penalty method for nonlinear optimization*. [S.l.], 2003.

[57] GOULD, N. I. M.; ORBAN, D.; SARTENAER, A.; TOINT, P. L. Sensitivity of trust-region algorithms to their parameters. *4OR*, v. 3, n. 3, p. 227–241, 2005.

[58] ZHANG, Y. *Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment*. [S.l.], 1996.

[59] TORRES, G. L. *Nonlinear Optimal Power Flow by Interior-Point and Non-Interior-Point Methods*. Tese (PhD Thesis) — University of Waterloo, Canada, 1998.

[60] FRISCH, K. R. *The Logarithmic Potential Method of Convex Programming*. Oslo, Norway, 1955.

[61] LUENBERGER, D. G. *Linear and Nonlinear Programming*. [S.l.]: Addison-Wesley Inc., 1984.

[62] HEI, L.; NOCEDAL, J.; WALTZ, R. A numerical study of active-set and interior-point methods for bound constrained optimization. In: BOCK, H.; KOSTINA, E.; PHU, H.; RANNACHER, R. (Ed.). *Modeling, Simulation and Optimization of Complex Processes*. [S.l.]: Springer Berlin Heidelberg, 2008. p. 273–292.

[63] FIACCO, A. V.; MCCORMICK, G. P. *Nonlinear Programming*: Sequential unconstrained minimization techniques. [S.l.]: JWS, 1968.

[64] DANTZIG, G. B. *Linear Programming and Extensions*. [S.l.]: Princeton University Press, 1963.

[65] KARMARKAR, N. A new polynomial-time algorithm for linear programming. *Combinatorica*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 4, n. 4, p. 373–395, 1984.

[66] SIMONI, V. R.; TORRES, G. L. Tratamento de inviabilidades no fluxo de potência ótimo por métodos de pontos interiores. In: *Anais do XVIII Congresso Brasileiro de Automática*. Bonito, MS, Brasil: [s.n.], 2010. v. 1, n. 66879, p. 4585–4592.

[67] SIMONI, V. R.; TORRES, G. L. Indicador logarítmico para identificação de inviabilidades no fluxo de potência ótimo. In: *IV Simpósio Brasileiro de Sistemas Elétricos*. [S.l.: s.n.], 2012. v. 1.

[68] EL-BAKRY, A. S.; TAPIA, R. A.; ZHANG, Y. *Logamithmic Indicators and the Identification of Subgroups of Variables in Interior Point Methods*. Houston, 1993.

[69] ANDERSON, E. *et al. LAPACK Users' Guide*. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN 0-89871-447-8 (paperback).

[70] DAVIS, T. A. *UMFPACK Version 4.6 User Guide*. Univ. of Florida, Gainesville, FL: Dept. of Computer and Information Science and Engineering, 2002.

[71] CELIS, M. R. *A Trust Region Strategy for Nonlinear Equality Constrained Optimization*. Tese (PhD Thesis) — Rice University, Houston, Texas, United States of America, 1985.

[72] MARQUARDT, D. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, v. 11, n. 2, p. 431–441, 1963.

[73] MORÉ, J. J. The Levenberg-Marquardt algorithm: Implementation and theory. In: WATSON, G. (Ed.). *Numerical Analysis*. [S.l.]: Springer Berlin / Heidelberg, 1978, (Lecture Notes in Mathematics, v. 630). p. 105–116.

[74] POWELL, M. J. D. A hybrid method for nonlinear equations. In: RABINOWITZ, P. (Ed.). *Numerical Methods for Nonlinear Algebraic Equations*. London: Gordon & Breach, 1970. p. 87–114.

[75] SOUZA, A. A. *Fluxo de Potência Ótimo Globalmente Convergente Utilizando Métodos de Pontos Interiores com Estratégias de Região de Confiaça*. Tese (Doutorado) — Universidade Federal de Pernambuco, Brasil, 2008.

[76] ERWAY, J. B.; GILL, P. E.; GRIFFIN, J. D. *Iterative Methods for Fiding a Trust Region Step*. [S.l.], 2007.

[77] ERWAY, J. B.; MARCIA, R. F. *MSS: MATLAB Software For L-BFGS Trust-Region Subproblems For Large-Scale Optimization*. [S.l.], 2012.

[78] GILBERT, J. R.; MOLER, C.; SCHREIBER, R. Sparse matrices in matlab: Design and implementation. *SIAM J. Matrix Anal. Appl*, v. 13, p. 333–356, 1992.

Appendix A

Parametric Grid

This appendix presents the numerical results for a parametric analysis in which the initial values of the trust region radius Δ_0 and the penalty parameter η_0 were picked from the grid formed by the sets $\{0.5, 1, 2, 4\}$ and $\{0.5, 2, 4, 8\}$, respectively. Additionally, for the BO and MBO algorithms, the value of the reduction factor of the trust region ξ on the vertical subproblem was made equal to one value of the set $\{0.4, 0.6, 0.8\}$.

Tables A.1, A.2, A.3, A.4, A.5 and A.6 show the main numerical experiments for the BO method. Additionally, Tables A.7, A.8, A.9, A.10, A.11 and A.12 show the main numerical results for the MBO method. Tables A.13, A.14, A.15, A.16, A.17 and A.18 show the results for the proposed Sℓ_1QP method.

Table A.1: Parametric grid for the IEEE 30-bus solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	4.83	1.62	3	9	12	12.19	17	13	16.14	19
0.5	0.5	0.6	30	4	4.93	1.57	3	8	12	12.09	16	14	16.23	19
0.5	0.5	0.8	30	4	5.07	1.80	3	9	11	12.05	16	13	16.21	19
0.5	2	0.4	30	4	4.80	1.63	3	9	12	12.18	17	13	16.08	19
0.5	2	0.6	30	4	4.80	1.54	3	9	12	12.09	16	14	16.15	19
0.5	2	0.8	30	4	5.07	1.95	3	9	11	12.05	16	13	16.11	19
0.5	4	0.4	30	4	4.73	1.48	3	9	12	12.18	17	13	16.10	19
0.5	4	0.6	30	4	4.73	1.39	3	9	12	12.09	16	14	16.17	19
0.5	4	0.8	30	4	5.00	1.84	3	9	11	12.05	16	13	16.12	19
0.5	8	0.4	30	4	4.73	1.48	3	9	12	12.18	17	13	16.10	19
0.5	8	0.6	30	4	4.73	1.39	3	9	12	12.09	16	14	16.17	19
0.5	8	0.8	30	4	5.00	1.84	3	9	11	12.05	16	13	16.13	19
1	0.5	0.4	30	4	5.00	1.74	3	8	12	12.08	16	14	16.03	19
1	0.5	0.6	30	4	5.00	1.74	3	8	11	11.95	14	14	16.04	19
1	0.5	0.8	30	4	5.00	1.74	3	8	11	11.94	12	14	16.04	19
1	2	0.4	30	4	4.83	1.58	3	9	12	12.08	16	14	16.01	19
1	2	0.6	30	4	4.97	1.90	3	10	11	11.95	14	14	16.00	19
1	2	0.8	30	4	4.97	1.90	3	10	11	11.94	12	14	16.00	19
1	4	0.4	30	4	4.93	1.86	3	10	12	12.08	16	14	15.99	19
1	4	0.6	30	4	4.93	1.86	3	10	11	11.95	14	14	16.00	19
1	4	0.8	30	4	4.93	1.86	3	10	11	11.94	12	14	16.00	19
1	8	0.4	30	4	4.93	1.86	3	10	12	12.08	16	14	15.99	19
1	8	0.6	30	4	4.93	1.86	3	10	11	11.95	14	14	16.00	19
1	8	0.8	30	4	4.93	1.86	3	10	11	11.94	12	14	16.00	19
2	0.5	0.4	30	4	4.97	1.83	3	8	11	11.94	12	14	15.89	19
2	0.5	0.6	30	4	4.97	1.83	3	8	11	11.88	12	14	15.89	19
2	0.5	0.8	30	4	4.97	1.83	3	8	11	11.83	12	14	15.89	19
2	2	0.4	30	4	5.47	2.78	3	12	11	11.97	13	14	15.82	19
2	2	0.6	30	4	5.43	2.70	3	11	11	11.89	12	14	15.82	19
2	2	0.8	30	4	5.43	2.70	3	11	11	11.83	12	14	15.82	19
2	4	0.4	30	4	5.40	2.63	3	10	11	11.95	13	14	15.83	19
2	4	0.6	30	4	5.40	2.63	3	10	11	11.89	12	14	15.83	19
2	4	0.8	30	4	5.40	2.63	3	10	11	11.83	12	14	15.83	19
2	8	0.4	30	4	5.40	2.63	3	10	11	11.95	13	14	15.83	19
2	8	0.6	30	4	5.40	2.63	3	10	11	11.89	12	14	15.83	19
2	8	0.8	30	4	5.40	2.63	3	10	11	11.83	12	14	15.83	19
4	0.5	0.4	30	4	4.83	1.62	3	8	11	11.83	12	14	15.83	19
4	0.5	0.6	30	4	4.83	1.62	3	8	11	11.82	12	14	15.83	19
4	0.5	0.8	30	4	4.83	1.62	3	8	11	11.82	12	14	15.83	19
4	2	0.4	30	4	4.87	1.94	3	10	11	11.83	12	14	15.79	19
4	2	0.6	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19
4	2	0.8	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19
4	4	0.4	30	4	4.87	1.94	3	10	11	11.83	12	14	15.79	19
4	4	0.6	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19
4	4	0.8	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19
4	8	0.4	30	4	4.87	1.94	3	10	11	11.83	12	14	15.79	19
4	8	0.6	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19
4	8	0.8	30	4	4.87	1.94	3	10	11	11.82	12	14	15.79	19

Table A.2: Parametric grid for the IEEE 57-bus solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	6	8.23	3.64	5	16	12	12.15	15	13	15.47	19
0.5	0.5	0.6	30	6	7.67	2.41	6	12	12	12.00	12	14	15.56	19
0.5	0.5	0.8	30	6	7.67	2.41	6	12	12	12.00	12	14	15.56	19
0.5	2	0.4	30	6	6.53	0.90	5	8	12	12.10	15	14	15.61	19
0.5	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
0.5	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
0.5	4	0.4	30	6	6.53	0.90	5	8	12	12.10	15	14	15.61	19
0.5	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
0.5	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
0.5	8	0.4	30	6	6.53	0.90	5	8	12	12.10	15	14	15.61	19
0.5	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
0.5	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	14	15.62	19
1	0.5	0.4	30	6	7.37	2.66	6	16	12	12.04	14	13	15.52	19
1	0.5	0.6	30	6	7.03	1.65	6	11	12	12.00	12	14	15.56	19
1	0.5	0.8	30	6	7.03	1.65	6	11	12	12.00	12	14	15.57	19
1	2	0.4	30	6	6.57	0.86	6	8	12	12.01	14	15	15.59	19
1	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
1	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
1	4	0.4	30	6	6.57	0.86	6	8	12	12.01	14	15	15.59	19
1	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
1	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
1	8	0.4	30	6	6.57	0.86	6	8	12	12.01	14	15	15.59	19
1	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
1	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	0.5	0.4	30	6	8.80	4.83	6	18	12	12.08	14	13	15.38	19
2	0.5	0.6	30	6	8.80	4.83	6	18	12	12.08	14	13	15.36	19
2	0.5	0.8	30	6	7.80	2.91	6	13	12	12.00	12	13	15.52	19
2	2	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	4	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	8	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
2	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	0.5	0.4	30	6	8.80	4.31	6	17	12	12.07	14	13	15.41	19
4	0.5	0.6	30	6	7.97	2.83	6	12	12	12.00	12	14	15.53	19
4	0.5	0.8	30	6	7.97	2.83	6	12	12	12.00	12	14	15.54	19
4	2	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	4	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	8	0.4	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19
4	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	15	15.59	19

Table A.3: Parametric grid for the IEEE 118-bus solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	4.33	0.55	4	6	13	14.84	22	15	16.94	19
0.5	0.5	0.6	30	4	4.17	0.38	4	5	13	14.04	22	15	16.93	19
0.5	0.5	0.8	30	4	4.20	0.41	4	5	13	13.50	22	14	17.01	19
0.5	2	0.4	30	4	4.23	0.43	4	5	13	14.80	22	15	16.87	19
0.5	2	0.6	30	4	4.20	0.41	4	5	13	14.03	22	15	16.93	19
0.5	2	0.8	30	4	4.20	0.41	4	5	13	13.50	22	14	17.02	19
0.5	4	0.4	30	4	4.23	0.43	4	5	13	14.80	22	15	16.87	19
0.5	4	0.6	30	4	4.17	0.38	4	5	13	14.04	22	15	16.93	19
0.5	4	0.8	30	4	4.17	0.38	4	5	13	13.51	22	14	17.02	19
0.5	8	0.4	30	4	4.23	0.43	4	5	13	14.80	22	15	16.86	19
0.5	8	0.6	30	4	4.17	0.38	4	5	13	14.04	22	15	16.92	19
0.5	8	0.8	30	4	4.17	0.38	4	5	13	13.51	22	14	17.02	19
1	0.5	0.4	30	4	4.07	0.37	4	6	13	13.59	22	15	16.96	19
1	0.5	0.6	30	4	4.17	0.65	4	7	13	13.18	20	15	16.99	19
1	0.5	0.8	30	4	4.13	0.51	4	6	13	13.14	20	15	16.94	19
1	2	0.4	30	4	4.00	0.00	4	4	13	13.55	22	15	16.94	19
1	2	0.6	30	4	4.00	0.00	4	4	13	13.10	15	15	16.97	19
1	2	0.8	30	4	4.00	0.00	4	4	13	13.07	15	15	16.91	19
1	4	0.4	30	4	4.00	0.00	4	4	13	13.55	22	15	16.94	19
1	4	0.6	30	4	4.00	0.00	4	4	13	13.10	15	15	16.97	19
1	4	0.8	30	4	4.00	0.00	4	4	13	13.07	15	15	16.91	19
1	8	0.4	30	4	4.00	0.00	4	4	13	13.55	22	15	16.94	19
1	8	0.6	30	4	4.00	0.00	4	4	13	13.10	15	15	16.97	19
1	8	0.8	30	4	4.00	0.00	4	4	13	13.07	15	15	16.91	19
2	0.5	0.4	30	4	4.07	0.37	4	6	13	13.11	19	15	16.99	19
2	0.5	0.6	30	4	4.00	0.00	4	4	13	13.00	13	15	16.96	19
2	0.5	0.8	30	4	4.00	0.00	4	4	12	12.98	15	15	16.98	19
2	2	0.4	30	4	4.00	0.00	4	4	13	13.07	15	15	16.95	19
2	2	0.6	30	4	4.00	0.00	4	4	13	13.00	13	15	16.95	19
2	2	0.8	30	4	4.00	0.00	4	4	12	12.98	15	15	16.97	19
2	4	0.4	30	4	4.00	0.00	4	4	13	13.07	15	15	16.95	19
2	4	0.6	30	4	4.00	0.00	4	4	13	13.00	13	15	16.95	19
2	4	0.8	30	4	4.00	0.00	4	4	12	12.98	15	15	16.97	19
2	8	0.4	30	4	4.00	0.00	4	4	13	13.07	15	15	16.95	19
2	8	0.6	30	4	4.00	0.00	4	4	13	13.00	13	15	16.95	19
2	8	0.8	30	4	4.00	0.00	4	4	12	12.98	15	15	16.97	19
4	0.5	0.4	30	4	4.03	0.18	4	5	12	13.00	16	15	17.00	19
4	0.5	0.6	30	4	4.10	0.55	4	7	12	12.98	19	15	17.02	19
4	0.5	0.8	30	4	4.10	0.55	4	7	13	13.02	18	15	17.02	20
4	2	0.4	30	4	4.00	0.00	4	4	12	12.98	15	15	16.98	19
4	2	0.6	30	4	4.00	0.00	4	4	12	12.94	13	15	16.98	19
4	2	0.8	30	4	4.00	0.00	4	4	13	13.00	13	15	16.98	19
4	4	0.4	30	4	4.00	0.00	4	4	12	12.98	15	15	16.98	19
4	4	0.6	30	4	4.00	0.00	4	4	12	12.94	13	15	16.98	19
4	4	0.8	30	4	4.00	0.00	4	4	13	13.00	13	15	16.98	19
4	8	0.4	30	4	4.00	0.00	4	4	12	12.98	15	15	16.98	19
4	8	0.6	30	4	4.00	0.00	4	4	12	12.94	13	15	16.98	19
4	8	0.8	30	4	4.00	0.00	4	4	13	13.00	13	15	16.98	19

Table A.4: Parametric grid for the IEEE 300-bus solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	28	23	19.32	3.91	12	28	14	18.07	30	18	22.32	30
0.5	0.5	0.6	30	17	18.10	2.83	12	24	14	17.95	30	18	22.63	30
0.5	0.5	0.8	28	17	16.86	2.52	13	23	14	17.66	30	15	22.67	30
0.5	2	0.4	28	17	17.79	4.57	13	30	14	17.99	30	18	22.08	30
0.5	2	0.6	27	17	16.74	3.12	13	29	14	17.86	30	18	22.49	30
0.5	2	0.8	29	15	15.72	2.22	12	22	14	17.88	30	18	22.61	30
0.5	4	0.4	26	17	16.96	3.35	13	27	14	18.03	30	18	22.07	30
0.5	4	0.6	28	15	16.64	2.70	13	25	14	17.74	30	18	22.45	30
0.5	4	0.8	30	17	16.60	3.23	13	27	14	17.59	30	18	22.65	30
0.5	8	0.4	28	17	16.57	2.99	13	25	14	18.17	30	18	22.03	30
0.5	8	0.6	26	15	16.81	3.39	13	29	14	17.72	30	18	22.39	30
0.5	8	0.8	29	17	16.69	3.25	13	27	14	17.62	30	18	22.63	30
1	0.5	0.4	27	15	17.85	3.08	13	24	14	17.64	30	17	22.39	30
1	0.5	0.6	28	15	16.96	3.53	14	30	14	17.64	30	18	22.57	30
1	0.5	0.8	27	15	15.70	2.27	12	24	14	17.17	30	18	22.80	30
1	2	0.4	29	15	16.62	3.62	12	30	14	17.78	30	17	22.32	30
1	2	0.6	28	15	16.29	2.80	12	27	14	17.56	30	18	22.63	30
1	2	0.8	27	15	15.48	2.23	11	21	14	17.32	30	18	22.72	30
1	4	0.4	27	15	16.41	3.53	12	27	14	17.67	30	17	22.35	30
1	4	0.6	27	15	15.81	2.00	12	23	14	17.58	30	18	22.59	30
1	4	0.8	27	15	15.56	1.93	12	20	14	17.09	30	13	22.71	30
1	8	0.4	27	16	15.67	2.22	12	22	14	17.74	30	18	22.37	30
1	8	0.6	28	15	15.75	1.76	12	21	14	17.62	30	17	22.61	30
1	8	0.8	27	15	15.78	2.41	12	22	14	17.07	30	13	22.72	30
2	0.5	0.4	27	18	18.26	3.53	12	26	14	17.28	30	11	22.48	30
2	0.5	0.6	25	15	15.84	2.98	11	26	14	17.12	30	14	22.46	30
2	0.5	0.8	24	14	16.04	3.42	13	27	14	16.79	30	17	22.61	30
2	2	0.4	30	15	16.63	3.03	11	25	14	17.22	30	11	22.35	30
2	2	0.6	27	14	15.22	2.62	12	26	14	16.96	30	16	22.43	30
2	2	0.8	26	14	14.42	1.33	11	17	14	16.56	30	17	22.47	30
2	4	0.4	27	15	16.11	2.90	11	26	14	17.06	30	11	22.29	30
2	4	0.6	27	15	14.85	1.96	12	22	14	17.00	30	16	22.31	30
2	4	0.8	26	14	14.19	1.30	11	17	14	16.57	30	17	22.45	30
2	8	0.4	28	15	16.00	3.65	11	30	14	17.09	30	11	22.32	30
2	8	0.6	27	15	14.85	1.83	12	22	14	17.01	30	16	22.37	30
2	8	0.8	26	14	14.19	1.30	11	17	14	16.56	30	17	22.45	30
4	0.5	0.4	25	15	16.44	2.00	13	21	14	16.83	30	16	22.46	30
4	0.5	0.6	25	15	15.32	2.67	11	22	14	16.69	30	16	22.74	30
4	0.5	0.8	23	15	17.70	3.70	13	26	14	16.44	30	16	22.89	30
4	2	0.4	25	15	15.56	2.52	11	25	14	16.74	30	16	22.41	30
4	2	0.6	27	15	14.70	2.03	11	18	14	16.60	30	16	22.60	30
4	2	0.8	24	15	15.12	3.21	12	27	14	16.07	30	9	22.71	30
4	4	0.4	26	15	15.19	1.81	11	20	14	16.84	30	16	22.37	30
4	4	0.6	26	15	14.96	2.52	11	22	14	16.48	30	16	22.60	30
4	4	0.8	22	15	14.05	1.40	12	16	14	16.07	30	16	22.71	30
4	8	0.4	27	15	15.07	1.54	11	18	14	16.87	30	16	22.38	30
4	8	0.6	27	15	14.85	2.25	11	22	14	16.56	30	16	22.64	30
4	8	0.8	22	15	14.05	1.40	12	16	14	16.07	30	16	22.73	30

Table A.5: Parametric grid for the REAL-A solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	25	8	10.04	1.95	7	14	12	14.70	18	11	18.17	30
0.5	0.5	0.6	28	8	10.14	2.48	6	16	12	14.56	19	13	18.29	30
0.5	0.5	0.8	29	8	10.97	3.39	7	18	12	14.62	19	14	18.10	30
0.5	2	0.4	21	8	12.38	4.96	7	26	12	14.91	18	11	18.01	30
0.5	2	0.6	26	8	10.92	2.30	7	14	12	14.84	19	14	18.69	30
0.5	2	0.8	25	8	10.88	3.14	7	17	12	14.49	19	15	18.33	30
0.5	4	0.4	21	8	13.71	6.13	7	27	12	14.91	18	11	17.89	30
0.5	4	0.6	26	8	12.12	4.21	7	25	12	14.83	19	14	18.35	30
0.5	4	0.8	25	8	12.56	5.79	8	30	12	14.61	19	15	18.22	30
0.5	8	0.4	22	8	15.27	8.22	7	28	12	14.85	18	11	17.61	30
0.5	8	0.6	25	8	12.36	4.81	7	21	12	14.71	19	14	18.14	30
0.5	8	0.8	25	8	13.00	5.77	8	26	12	14.46	18	15	18.11	30
1	0.5	0.4	23	8	11.43	2.81	8	16	12	14.63	18	11	18.20	30
1	0.5	0.6	27	8	11.44	2.44	8	15	12	14.61	21	11	18.03	30
1	0.5	0.8	27	13	10.59	2.52	7	17	12	14.42	19	11	18.21	30
1	2	0.4	23	8	12.13	4.74	7	25	12	14.76	19	14	18.15	30
1	2	0.6	24	12	12.33	2.63	8	17	12	14.74	20	11	18.10	30
1	2	0.8	28	9	10.64	2.72	7	17	12	14.50	19	10	18.46	30
1	4	0.4	22	8	13.82	6.77	7	28	12	14.79	18	14	18.03	30
1	4	0.6	22	8	14.41	6.12	8	28	12	14.80	20	14	18.06	30
1	4	0.8	27	8	11.44	5.32	7	30	12	14.42	19	10	18.45	30
1	8	0.4	21	8	13.57	7.69	7	30	12	14.66	18	12	18.09	30
1	8	0.6	23	8	16.65	7.47	8	30	12	14.82	19	11	17.68	30
1	8	0.8	24	8	13.25	7.44	7	30	12	14.30	19	14	18.08	30
2	0.5	0.4	20	8	11.80	3.27	8	16	12	14.43	19	14	17.69	30
2	0.5	0.6	28	8	11.39	2.78	8	18	12	14.59	19	11	18.19	30
2	0.5	0.8	26	8	10.50	2.42	8	18	12	14.38	18	12	18.33	30
2	2	0.4	22	9	12.77	4.20	8	22	12	14.70	19	14	17.78	30
2	2	0.6	25	9	11.68	2.82	8	18	12	14.62	19	10	18.13	30
2	2	0.8	26	9	10.73	2.16	8	17	12	14.36	18	12	18.39	30
2	4	0.4	22	9	13.18	5.10	8	24	12	14.73	19	14	17.82	30
2	4	0.6	28	9	12.39	4.48	8	27	12	14.61	19	10	18.13	30
2	4	0.8	24	9	11.33	3.89	8	23	12	14.40	18	12	18.32	30
2	8	0.4	18	8	15.22	8.16	8	29	12	14.63	19	14	17.59	30
2	8	0.6	26	9	14.12	6.54	8	29	12	14.53	19	10	18.01	30
2	8	0.8	27	9	12.74	5.57	8	26	12	14.38	18	9	18.11	30
4	0.5	0.4	23	15	12.96	3.66	8	18	12	14.58	20	12	17.79	30
4	0.5	0.6	26	8	11.73	2.88	8	16	12	14.42	19	13	18.04	30
4	0.5	0.8	27	8	11.48	3.98	8	28	12	14.32	19	7	18.17	30
4	2	0.4	23	10	13.43	4.94	8	24	12	14.71	20	12	18.08	30
4	2	0.6	26	14	13.31	3.99	8	27	12	14.62	19	7	18.03	30
4	2	0.8	27	9	12.30	3.57	8	21	12	14.45	18	11	18.35	30
4	4	0.4	25	8	13.56	5.43	8	23	12	14.65	20	12	18.09	30
4	4	0.6	23	9	14.39	5.16	8	26	12	14.62	19	8	17.80	30
4	4	0.8	26	9	12.12	4.44	8	26	12	14.45	18	7	18.34	30
4	8	0.4	22	10	13.64	7.18	8	29	12	14.49	19	12	18.09	30
4	8	0.6	24	9	16.29	7.28	8	30	12	14.54	19	8	17.59	30
4	8	0.8	26	9	13.58	6.57	8	30	12	14.39	18	7	18.14	30

Table A.6: Parametric grid for the IEEE REAL-R solved by the BO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Horizontal		
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	3.93	0.45	3	5	13	15.44	17	14	16.52	20
0.5	0.5	0.6	30	4	3.93	0.45	3	5	13	15.43	17	14	16.52	20
0.5	0.5	0.8	30	4	3.93	0.45	3	5	13	15.43	17	14	16.52	20
0.5	2	0.4	30	4	4.13	0.63	3	7	13	15.45	17	14	16.53	20
0.5	2	0.6	30	4	4.07	0.37	3	5	13	15.45	17	14	16.51	20
0.5	2	0.8	30	4	4.07	0.37	3	5	13	15.44	17	14	16.51	20
0.5	4	0.4	30	4	4.03	0.32	3	5	13	15.45	17	14	16.50	20
0.5	4	0.6	30	4	4.07	0.37	3	5	13	15.45	17	14	16.51	20
0.5	4	0.8	30	4	4.07	0.37	3	5	13	15.44	17	14	16.51	20
0.5	8	0.4	30	4	4.03	0.32	3	5	13	15.45	17	14	16.50	20
0.5	8	0.6	30	4	4.07	0.37	3	5	13	15.45	17	14	16.51	20
0.5	8	0.8	30	4	4.07	0.37	3	5	13	15.44	17	14	16.51	20
1	0.5	0.4	30	4	3.93	0.45	3	5	13	15.43	17	14	16.51	20
1	0.5	0.6	30	4	3.93	0.45	3	5	13	15.41	17	14	16.49	20
1	0.5	0.8	30	4	3.93	0.45	3	5	13	15.41	17	14	16.50	20
1	2	0.4	30	4	4.07	0.37	3	5	13	15.44	17	14	16.50	20
1	2	0.6	30	4	4.07	0.37	3	5	13	15.42	17	14	16.49	20
1	2	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.50	20
1	4	0.4	30	4	4.07	0.37	3	5	13	15.44	17	14	16.50	20
1	4	0.6	30	4	4.07	0.37	3	5	13	15.42	17	14	16.49	20
1	4	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.50	20
1	8	0.4	30	4	4.07	0.37	3	5	13	15.44	17	14	16.50	20
1	8	0.6	30	4	4.07	0.37	3	5	13	15.42	17	14	16.49	20
1	8	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.50	20
2	0.5	0.4	30	4	3.93	0.45	3	5	13	15.41	17	14	16.48	20
2	0.5	0.6	30	4	3.93	0.45	3	5	13	15.40	17	14	16.46	20
2	0.5	0.8	30	4	3.93	0.45	3	5	13	15.40	17	14	16.48	20
2	2	0.4	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
2	2	0.6	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
2	2	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
2	4	0.4	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
2	4	0.6	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
2	4	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
2	8	0.4	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
2	8	0.6	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
2	8	0.8	30	4	4.07	0.37	3	5	13	15.42	17	14	16.48	20
4	0.5	0.4	30	4	3.90	0.40	3	5	13	15.40	17	14	16.46	20
4	0.5	0.6	30	4	3.90	0.40	3	5	13	15.39	17	14	16.48	20
4	0.5	0.8	30	4	3.90	0.40	3	5	13	15.38	17	14	16.50	20
4	2	0.4	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
4	2	0.6	30	4	4.03	0.32	3	5	13	15.41	17	14	16.47	20
4	2	0.8	30	4	4.03	0.32	3	5	13	15.40	17	14	16.50	20
4	4	0.4	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
4	4	0.6	30	4	4.03	0.32	3	5	13	15.41	17	14	16.47	20
4	4	0.8	30	4	4.03	0.32	3	5	13	15.40	17	14	16.50	20
4	8	0.4	30	4	4.03	0.32	3	5	13	15.42	17	14	16.46	20
4	8	0.6	30	4	4.03	0.32	3	5	13	15.41	17	14	16.47	20
4	8	0.8	30	4	4.03	0.32	3	5	13	15.40	17	14	16.50	20

Table A.7: Parametric grid for the IEEE 30-bus solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	4.83	1.62	3	9	12	12.57	17	1.17	14	16.22	19
0.5	0.5	0.6	30	4	4.93	1.57	3	8	12	12.23	16	1.17	14	16.27	19
0.5	0.5	0.8	30	4	5.07	1.80	3	9	11	12.23	16	1.10	14	16.24	19
0.5	2	0.4	30	4	4.80	1.63	3	9	12	12.57	17	1.17	14	16.16	19
0.5	2	0.6	30	4	4.80	1.54	3	9	12	12.23	16	1.17	14	16.19	19
0.5	2	0.8	30	4	5.07	1.95	3	9	11	12.23	16	1.10	14	16.15	19
0.5	4	0.4	30	4	4.73	1.48	3	9	12	12.57	17	1.17	14	16.19	19
0.5	4	0.6	30	4	4.73	1.39	3	9	12	12.23	16	1.17	14	16.22	19
0.5	4	0.8	30	4	5.00	1.84	3	9	11	12.23	16	1.10	14	16.15	19
0.5	8	0.4	30	4	4.73	1.48	3	9	12	12.57	17	1.17	14	16.19	19
0.5	8	0.6	30	4	4.73	1.39	3	9	12	12.23	16	1.17	14	16.22	19
0.5	8	0.8	30	4	5.00	1.84	3	9	11	12.23	16	1.10	14	16.16	19
1	0.5	0.4	30	4	5.00	1.74	3	8	12	12.27	16	1.10	14	16.05	19
1	0.5	0.6	30	4	5.00	1.74	3	8	11	11.80	14	1.00	14	16.04	19
1	0.5	0.8	30	4	5.00	1.74	3	8	11	11.73	12	1.00	14	16.04	19
1	2	0.4	30	4	4.83	1.58	3	9	12	12.27	16	1.10	14	16.03	19
1	2	0.6	30	4	4.97	1.90	3	10	11	11.80	14	1.00	14	16.00	19
1	2	0.8	30	4	4.97	1.90	3	10	11	11.73	12	1.00	14	16.00	19
1	4	0.4	30	4	4.93	1.86	3	10	12	12.27	16	1.10	14	16.01	19
1	4	0.6	30	4	4.93	1.86	3	10	11	11.80	14	1.00	14	16.00	19
1	4	0.8	30	4	4.93	1.86	3	10	11	11.73	12	1.00	14	16.00	19
1	8	0.4	30	4	4.93	1.86	3	10	12	12.27	16	1.10	14	16.01	19
1	8	0.6	30	4	4.93	1.86	3	10	11	11.80	14	1.00	14	16.00	19
1	8	0.8	30	4	4.93	1.86	3	10	11	11.73	12	1.00	14	16.00	19
2	0.5	0.4	30	4	4.97	1.83	3	8	11	11.73	12	1.00	14	15.89	19
2	0.5	0.6	30	4	4.97	1.83	3	8	11	11.47	12	1.00	14	15.89	19
2	0.5	0.8	30	4	4.97	1.83	3	8	11	11.23	12	1.00	14	15.89	19
2	2	0.4	30	4	5.43	2.70	3	11	11	11.73	12	1.00	14	15.82	19
2	2	0.6	30	4	5.43	2.70	3	11	11	11.47	12	1.00	14	15.82	19
2	2	0.8	30	4	5.43	2.70	3	11	11	11.23	12	1.00	14	15.82	19
2	4	0.4	30	4	5.40	2.63	3	10	11	11.73	12	1.00	14	15.83	19
2	4	0.6	30	4	5.40	2.63	3	10	11	11.47	12	1.00	14	15.83	19
2	4	0.8	30	4	5.40	2.63	3	10	11	11.23	12	1.00	14	15.83	19
2	8	0.4	30	4	5.40	2.63	3	10	11	11.73	12	1.00	14	15.83	19
2	8	0.6	30	4	5.40	2.63	3	10	11	11.47	12	1.00	14	15.83	19
2	8	0.8	30	4	5.40	2.63	3	10	11	11.23	12	1.00	14	15.83	19
4	0.5	0.4	30	4	4.83	1.62	3	8	11	11.23	12	1.00	14	15.83	19
4	0.5	0.6	30	4	4.83	1.62	3	8	11	11.20	12	1.00	14	15.83	19
4	0.5	0.8	30	4	4.83	1.62	3	8	11	11.20	12	1.00	14	15.83	19
4	2	0.4	30	4	4.87	1.94	3	10	11	11.23	12	1.00	14	15.79	19
4	2	0.6	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19
4	2	0.8	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19
4	4	0.4	30	4	4.87	1.94	3	10	11	11.23	12	1.00	14	15.79	19
4	4	0.6	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19
4	4	0.8	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19
4	8	0.4	30	4	4.87	1.94	3	10	11	11.23	12	1.00	14	15.79	19
4	8	0.6	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19
4	8	0.8	30	4	4.87	1.94	3	10	11	11.20	12	1.00	14	15.79	19

Table A.8: Parametric grid for the IEEE 57-bus solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	6	7.67	2.41	6	12	12	12.27	15	1.00	14	15.57	19
0.5	0.5	0.6	30	6	7.67	2.41	6	12	12	12.00	12	1.00	14	15.57	19
0.5	0.5	0.8	30	6	7.67	2.41	6	12	12	12.00	12	1.00	14	15.57	19
0.5	2	0.4	30	6	6.57	0.86	6	8	12	12.27	15	1.00	14	15.62	19
0.5	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
0.5	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
0.5	4	0.4	30	6	6.57	0.86	6	8	12	12.27	15	1.00	14	15.62	19
0.5	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
0.5	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
0.5	8	0.4	30	6	6.57	0.86	6	8	12	12.27	15	1.00	14	15.62	19
0.5	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
0.5	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	14	15.62	19
1	0.5	0.4	30	6	7.03	1.65	6	11	12	12.00	12	1.00	14	15.58	19
1	0.5	0.6	30	6	7.03	1.65	6	11	12	12.00	12	1.00	14	15.58	19
1	0.5	0.8	30	6	7.03	1.65	6	11	12	12.00	12	1.00	14	15.58	19
1	2	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	4	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	8	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
1	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	0.5	0.4	30	6	7.63	2.74	6	13	12	12.00	12	1.00	13	15.53	19
2	0.5	0.6	30	6	7.63	2.74	6	13	12	12.00	12	1.00	13	15.53	19
2	0.5	0.8	30	6	7.63	2.74	6	13	12	12.00	12	1.00	13	15.53	19
2	2	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	4	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	8	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
2	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	0.5	0.4	30	6	7.97	2.83	6	12	12	12.00	12	1.00	14	15.55	19
4	0.5	0.6	30	6	7.97	2.83	6	12	12	12.00	12	1.00	14	15.55	19
4	0.5	0.8	30	6	7.97	2.83	6	12	12	12.00	12	1.00	14	15.55	19
4	2	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	2	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	2	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	4	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	4	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	4	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	8	0.4	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	8	0.6	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19
4	8	0.8	30	6	6.57	0.86	6	8	12	12.00	12	1.00	15	15.59	19

Table A.9: Parametric grid for the IEEE 118-bus solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	4.33	0.55	4	6	13	15.85	22	2.77	16	16.93	18
0.5	0.5	0.6	30	4	4.17	0.38	4	5	13	14.92	22	2.13	16	17.02	18
0.5	0.5	0.8	30	4	4.20	0.41	4	5	13	14.01	22	1.77	15	17.00	19
0.5	2	0.4	30	4	4.23	0.43	4	5	13	15.81	22	2.70	16	16.84	18
0.5	2	0.6	30	4	4.20	0.41	4	5	13	14.92	22	2.13	16	16.99	18
0.5	2	0.8	30	4	4.20	0.41	4	5	13	14.01	22	1.77	15	17.01	19
0.5	4	0.4	30	4	4.23	0.43	4	5	13	15.81	22	2.70	16	16.84	18
0.5	4	0.6	30	4	4.17	0.38	4	5	13	14.92	22	2.13	16	16.99	18
0.5	4	0.8	30	4	4.17	0.38	4	5	13	14.01	22	1.77	15	17.01	19
0.5	8	0.4	30	4	4.23	0.43	4	5	13	15.81	22	2.70	16	16.82	18
0.5	8	0.6	30	4	4.17	0.38	4	5	13	14.92	22	2.13	16	16.98	18
0.5	8	0.8	30	4	4.17	0.38	4	5	13	14.01	22	1.77	15	17.01	19
1	0.5	0.4	29	4	4.00	0.00	4	4	13	14.16	22	1.59	15	16.99	19
1	0.5	0.6	28	4	4.00	0.00	4	4	13	13.23	15	1.36	15	17.02	19
1	0.5	0.8	28	4	4.00	0.00	4	4	13	13.18	15	1.32	15	16.94	19
1	2	0.4	30	4	4.00	0.00	4	4	13	14.17	22	1.60	15	16.98	19
1	2	0.6	30	4	4.00	0.00	4	4	13	13.27	15	1.37	15	17.02	19
1	2	0.8	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.93	19
1	4	0.4	30	4	4.00	0.00	4	4	13	14.17	22	1.60	15	16.98	19
1	4	0.6	30	4	4.00	0.00	4	4	13	13.27	15	1.37	15	17.02	19
1	4	0.8	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.93	19
1	8	0.4	30	4	4.00	0.00	4	4	13	14.17	22	1.60	15	16.98	19
1	8	0.6	30	4	4.00	0.00	4	4	13	13.27	15	1.37	15	17.02	19
1	8	0.8	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.93	19
2	0.5	0.4	29	4	4.00	0.00	4	4	13	13.19	15	1.31	15	16.98	19
2	0.5	0.6	30	4	4.00	0.00	4	4	13	13.00	13	1.23	15	16.96	19
2	0.5	0.8	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.98	19
2	2	0.4	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.96	19
2	2	0.6	30	4	4.00	0.00	4	4	13	13.00	13	1.23	15	16.96	19
2	2	0.8	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.97	19
2	4	0.4	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.96	19
2	4	0.6	30	4	4.00	0.00	4	4	13	13.00	13	1.23	15	16.96	19
2	4	0.8	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.97	19
2	8	0.4	30	4	4.00	0.00	4	4	13	13.18	15	1.33	15	16.96	19
2	8	0.6	30	4	4.00	0.00	4	4	13	13.00	13	1.23	15	16.96	19
2	8	0.8	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.97	19
4	0.5	0.4	29	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.99	19
4	0.5	0.6	29	4	4.00	0.00	4	4	12	12.76	13	1.00	15	16.99	19
4	0.5	0.8	29	4	4.00	0.00	4	4	13	13.00	13	1.00	15	16.99	19
4	2	0.4	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.98	19
4	2	0.6	30	4	4.00	0.00	4	4	12	12.77	13	1.00	15	16.98	19
4	2	0.8	30	4	4.00	0.00	4	4	13	13.00	13	1.00	15	16.98	19
4	4	0.4	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.98	19
4	4	0.6	30	4	4.00	0.00	4	4	12	12.77	13	1.00	15	16.98	19
4	4	0.8	30	4	4.00	0.00	4	4	13	13.00	13	1.00	15	16.98	19
4	8	0.4	30	4	4.00	0.00	4	4	12	12.93	15	1.00	15	16.98	19
4	8	0.6	30	4	4.00	0.00	4	4	12	12.77	13	1.00	15	16.98	19
4	8	0.8	30	4	4.00	0.00	4	4	13	13.00	13	1.00	15	16.98	19

Table A.10: Parametric grid for the IEEE 300-bus solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	27	19	18.52	3.76	12	28	14	24.52	30	6.67	18	22.82	30
0.5	0.5	0.6	28	16	17.57	2.67	14	25	14	24.63	30	5.71	18	23.09	30
0.5	0.5	0.8	29	16	16.93	3.37	13	29	14	24.43	30	5.24	18	22.81	30
0.5	2	0.4	29	13	17.38	3.55	13	29	14	24.38	30	6.00	18	22.55	30
0.5	2	0.6	27	16	16.07	1.80	12	20	14	24.30	30	5.37	18	23.02	30
0.5	2	0.8	29	18	15.41	2.10	11	18	14	24.39	30	5.03	18	22.82	30
0.5	4	0.4	29	13	17.76	4.18	13	30	14	24.38	30	6.00	18	22.62	30
0.5	4	0.6	28	16	16.61	2.82	12	27	14	24.23	30	5.18	18	22.92	30
0.5	4	0.8	29	17	16.69	3.31	13	30	14	24.16	30	4.93	18	22.94	30
0.5	8	0.4	29	18	17.21	3.54	13	28	14	24.39	30	5.97	19	22.55	30
0.5	8	0.6	26	16	16.12	2.07	12	22	14	24.13	30	5.12	18	22.98	30
0.5	8	0.8	28	17	16.14	2.07	13	23	14	24.21	30	4.96	19	22.92	30
1	0.5	0.4	26	16	17.65	3.68	13	29	14	24.26	30	5.54	19	22.86	30
1	0.5	0.6	29	15	17.07	3.52	14	30	14	24.33	30	5.17	18	22.95	27
1	0.5	0.8	27	15	15.67	2.40	12	24	14	23.69	30	4.30	18	23.04	30
1	2	0.4	28	15	15.96	2.66	12	24	14	23.94	30	5.14	10	22.82	30
1	2	0.6	27	15	15.63	1.67	13	21	14	24.49	30	4.56	19	22.97	28
1	2	0.8	28	15	15.43	2.18	12	23	14	23.84	30	4.32	13	22.94	30
1	4	0.4	26	15	15.96	3.13	12	28	14	23.94	30	5.12	19	22.89	30
1	4	0.6	27	15	15.70	1.75	12	21	14	24.52	30	4.52	19	22.93	28
1	4	0.8	27	15	15.30	1.68	12	18	14	23.72	30	4.04	13	23.01	30
1	8	0.4	26	15	15.35	1.79	12	18	14	23.97	30	5.04	19	22.93	28
1	8	0.6	28	15	15.86	1.96	12	21	14	24.55	30	4.54	19	22.92	28
1	8	0.8	28	15	15.61	2.48	12	25	14	23.76	30	4.04	13	23.07	30
2	0.5	0.4	27	15	17.41	3.99	12	26	14	23.95	30	5.22	11	22.77	29
2	0.5	0.6	25	15	16.20	3.55	11	26	14	23.44	30	4.20	14	22.81	28
2	0.5	0.8	21	14	15.00	1.82	12	18	14	23.15	30	3.43	19	22.91	30
2	2	0.4	29	15	16.21	3.31	11	25	14	23.73	30	4.59	11	22.86	30
2	2	0.6	28	15	16.43	4.45	12	30	14	23.17	30	3.71	17	22.71	30
2	2	0.8	26	14	14.23	1.48	11	17	14	22.79	30	3.23	19	22.79	30
2	4	0.4	30	15	16.90	4.96	11	30	14	23.80	30	4.80	11	22.88	30
2	4	0.6	27	15	15.93	4.00	12	29	14	23.29	30	3.74	17	22.62	28
2	4	0.8	26	14	14.23	1.39	11	17	14	22.80	30	3.19	19	22.74	30
2	8	0.4	30	15	16.83	4.56	11	30	14	23.81	30	4.80	11	22.84	30
2	8	0.6	26	15	14.96	1.84	12	21	14	23.43	30	3.81	17	22.77	28
2	8	0.8	26	14	14.23	1.39	11	17	14	22.80	30	3.19	19	22.74	30
4	0.5	0.4	24	15	16.08	1.91	12	20	14	23.47	30	4.12	18	22.85	30
4	0.5	0.6	24	15	15.46	2.36	11	21	14	22.79	30	3.33	19	22.99	27
4	0.5	0.8	18	15	16.72	2.93	13	24	14	22.25	30	3.78	17	23.09	30
4	2	0.4	25	15	15.44	2.83	10	24	14	23.06	30	3.84	18	22.89	30
4	2	0.6	26	15	14.38	1.77	11	18	14	22.39	30	3.12	19	22.81	27
4	2	0.8	23	15	14.43	1.50	12	18	14	20.87	30	2.70	16	22.85	30
4	4	0.4	27	15	15.52	2.85	10	25	14	23.23	30	3.85	18	22.85	30
4	4	0.6	26	15	14.38	1.65	11	18	14	22.39	30	3.12	19	22.86	27
4	4	0.8	23	15	14.48	1.62	12	19	14	20.87	30	2.70	16	22.87	30
4	8	0.4	27	15	15.04	1.83	10	18	14	23.23	30	3.85	18	22.87	30
4	8	0.6	26	15	14.38	1.65	11	18	14	22.39	30	3.12	19	22.86	27
4	8	0.8	23	15	14.70	1.99	12	20	14	20.87	30	2.70	16	22.86	30

Table A.11: Parametric grid for the REAL-A solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	16	8	9.06	1.53	7	12	12	13.82	18	3.75	15	17.19	30
0.5	0.5	0.6	15	8	9.53	2.75	6	17	12	13.80	18	2.87	15	17.36	30
0.5	0.5	0.8	13	8	9.08	1.55	8	12	12	13.71	18	3.46	14	16.91	20
0.5	2	0.4	13	8	9.08	2.14	7	15	12	13.71	18	3.69	15	17.33	30
0.5	2	0.6	11	8	8.45	1.37	7	12	12	13.26	18	2.36	14	17.78	30
0.5	2	0.8	9	8	8.44	0.73	8	10	12	13.37	17	1.89	15	17.38	30
0.5	4	0.4	13	8	9.62	3.33	7	17	12	13.74	18	4.31	15	17.36	30
0.5	4	0.6	10	8	8.20	0.63	7	9	12	13.00	17	1.60	14	17.83	30
0.5	4	0.8	9	8	8.89	1.36	8	12	12	13.37	17	1.89	15	17.27	30
0.5	8	0.4	14	8	10.43	6.00	7	28	12	13.76	18	5.36	15	17.18	30
0.5	8	0.6	10	8	8.30	0.82	7	10	12	13.00	17	1.60	14	17.84	30
0.5	8	0.8	9	8	8.89	1.36	8	12	12	13.37	17	1.89	15	17.27	30
1	0.5	0.4	16	8	10.12	2.90	7	16	12	13.81	18	4.44	15	17.14	30
1	0.5	0.6	12	8	10.17	2.48	8	15	12	13.50	19	4.17	15	17.38	30
1	0.5	0.8	12	8	10.00	2.63	8	17	12	13.35	17	3.58	14	17.49	30
1	2	0.4	13	8	8.77	1.83	7	14	12	13.54	17	2.77	14	17.66	30
1	2	0.6	11	8	9.64	2.42	8	16	12	13.46	18	3.27	15	17.46	30
1	2	0.8	11	8	9.82	2.64	8	15	12	13.47	19	3.73	15	17.61	30
1	4	0.4	15	8	9.33	3.37	7	21	12	13.53	18	3.27	14	17.53	30
1	4	0.6	10	8	8.60	0.70	8	10	12	13.04	17	1.90	15	17.84	30
1	4	0.8	14	8	9.29	1.82	8	15	12	13.54	19	3.14	15	17.70	30
1	8	0.4	16	8	9.50	4.24	7	25	12	13.61	18	3.50	14	17.60	30
1	8	0.6	11	8	9.18	1.83	8	14	12	13.33	18	2.73	15	17.72	30
1	8	0.8	10	8	8.60	0.70	8	10	12	13.00	17	1.80	15	17.41	30
2	0.5	0.4	9	8	8.89	1.05	8	11	12	12.83	16	1.56	15	17.06	30
2	0.5	0.6	14	8	10.93	3.36	8	18	12	13.98	19	5.29	14	17.24	30
2	0.5	0.8	11	8	9.73	2.97	8	18	12	13.32	17	3.18	15	17.33	30
2	2	0.4	7	8	9.14	1.77	8	13	12	13.00	18	2.14	15	17.27	30
2	2	0.6	10	8	11.30	3.43	8	17	12	13.90	19	5.20	15	17.20	30
2	2	0.8	9	8	10.00	2.87	8	17	12	13.31	17	3.11	15	17.29	30
2	4	0.4	8	8	10.00	2.93	8	16	12	13.06	18	2.12	13	17.26	30
2	4	0.6	13	8	11.23	3.00	8	16	12	13.92	19	4.62	13	17.50	30
2	4	0.8	10	8	10.60	3.31	8	17	12	13.32	17	3.00	13	17.27	30
2	8	0.4	6	8	8.67	0.82	8	10	12	12.50	15	1.33	15	17.16	30
2	8	0.6	10	8	10.70	3.33	8	17	12	13.63	19	4.90	15	17.44	30
2	8	0.8	10	8	10.00	2.71	8	17	12	13.53	18	3.60	15	17.33	30
4	0.5	0.4	11	8	9.91	2.91	8	18	12	13.25	18	3.82	14	17.31	30
4	0.5	0.6	11	8	9.09	1.87	8	14	12	13.12	18	2.73	15	17.51	30
4	0.5	0.8	12	8	11.00	5.69	8	28	12	13.41	19	4.83	14	17.21	30
4	2	0.4	10	8	9.40	2.50	8	16	12	13.16	17	2.10	15	17.86	30
4	2	0.6	13	9	12.31	5.41	8	27	12	13.86	19	5.62	15	17.45	30
4	2	0.8	11	9	10.18	2.93	8	17	12	13.39	18	3.18	6	17.89	30
4	4	0.4	12	8	9.08	1.31	8	12	12	13.22	17	2.08	14	17.99	30
4	4	0.6	12	9	10.00	2.59	8	16	12	13.27	18	2.83	14	17.78	30
4	4	0.8	13	9	10.00	3.46	8	21	12	13.49	18	3.62	14	17.83	30
4	8	0.4	12	8	9.08	1.31	8	12	12	13.26	17	2.25	15	17.61	30
4	8	0.6	11	9	10.09	2.70	8	16	12	13.25	17	2.82	15	17.39	30
4	8	0.8	13	9	10.15	3.46	8	21	12	13.56	18	3.85	6	17.70	30

Table A.12: Parametric grid for the REAL-R solved by the MBO Method.

Δ_0	η_0	ξ	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Vertical			Trust Region Problem			
									$j_{\min}$	$\overline{j}$	$j_{\max}$	$\overline{k}_{first}$	$j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	0.4	30	4	3.93	0.45	3	5	13	15.19	17	2.67	14	15.65	17
0.5	0.5	0.6	30	4	3.93	0.45	3	5	13	15.18	17	2.67	14	15.63	17
0.5	0.5	0.8	30	4	3.93	0.45	3	5	13	15.17	17	2.67	14	15.63	17
0.5	2	0.4	29	4	4.03	0.33	3	5	13	15.19	17	2.66	14	15.67	17
0.5	2	0.6	30	4	4.07	0.37	3	5	13	15.18	17	2.67	14	15.65	17
0.5	2	0.8	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
0.5	4	0.4	30	4	4.03	0.32	3	5	13	15.19	17	2.67	14	15.66	17
0.5	4	0.6	30	4	4.07	0.37	3	5	13	15.18	17	2.67	14	15.65	17
0.5	4	0.8	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
0.5	8	0.4	30	4	4.03	0.32	3	5	13	15.19	17	2.67	14	15.66	17
0.5	8	0.6	30	4	4.07	0.37	3	5	13	15.18	17	2.67	14	15.65	17
0.5	8	0.8	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
1	0.5	0.4	30	4	3.93	0.45	3	5	13	15.17	17	2.67	14	15.63	17
1	0.5	0.6	30	4	3.93	0.45	3	5	13	15.15	17	2.67	14	15.65	17
1	0.5	0.8	30	4	3.93	0.45	3	5	13	15.15	17	2.67	14	15.68	17
1	2	0.4	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
1	2	0.6	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.68	17
1	2	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
1	4	0.4	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
1	4	0.6	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.68	17
1	4	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
1	8	0.4	30	4	4.07	0.37	3	5	13	15.17	17	2.67	14	15.66	17
1	8	0.6	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.68	17
1	8	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
2	0.5	0.4	30	4	3.93	0.45	3	5	13	15.15	17	2.67	14	15.68	17
2	0.5	0.6	30	4	3.93	0.45	3	5	13	15.15	17	2.67	14	15.67	17
2	0.5	0.8	30	4	3.93	0.45	3	5	13	15.15	17	2.67	14	15.66	17
2	2	0.4	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
2	2	0.6	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.68	17
2	2	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.67	17
2	4	0.4	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
2	4	0.6	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.68	17
2	4	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.67	17
2	8	0.4	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.69	17
2	8	0.6	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.68	17
2	8	0.8	30	4	4.07	0.37	3	5	13	15.15	17	2.67	14	15.67	17
4	0.5	0.4	30	4	3.90	0.40	3	5	13	15.15	17	2.67	14	15.71	17
4	0.5	0.6	30	4	3.90	0.40	3	5	13	15.13	17	2.67	14	15.70	17
4	0.5	0.8	30	4	3.90	0.40	3	5	13	15.12	17	2.67	14	15.70	17
4	2	0.4	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.72	17
4	2	0.6	30	4	4.03	0.32	3	5	13	15.13	17	2.67	14	15.72	17
4	2	0.8	30	4	4.03	0.32	3	5	13	15.12	17	2.67	14	15.72	17
4	4	0.4	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.72	17
4	4	0.6	30	4	4.03	0.32	3	5	13	15.13	17	2.67	14	15.72	17
4	4	0.8	30	4	4.03	0.32	3	5	13	15.12	17	2.67	14	15.72	17
4	8	0.4	30	4	4.03	0.32	3	5	13	15.15	17	2.67	14	15.72	17
4	8	0.6	30	4	4.03	0.32	3	5	13	15.13	17	2.67	14	15.72	17
4	8	0.8	30	4	4.03	0.32	3	5	13	15.12	17	2.67	14	15.72	17

Table A.13: Parametric grid for the IEEE 30-bus solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	30	4	4.77	1.61	3	9	14	15.10	17
0.5	2	30	4	4.77	1.61	3	9	14	15.10	17
0.5	4	30	4	4.77	1.61	3	9	14	15.10	17
0.5	8	30	4	4.77	1.61	3	9	14	15.10	17
1	0.5	30	4	4.90	1.65	3	9	15	15.44	16
1	2	30	4	4.90	1.65	3	9	15	15.44	16
1	4	30	4	4.90	1.65	3	9	15	15.44	16
1	8	30	4	4.90	1.65	3	9	15	15.44	16
2	0.5	30	4	4.87	1.83	3	9	15	15.89	17
2	2	30	4	4.87	1.83	3	9	15	15.89	17
2	4	30	4	4.87	1.83	3	9	15	15.89	17
2	8	30	4	4.87	1.83	3	9	15	15.89	17
4	0.5	30	4	4.70	1.60	3	10	15	16.05	17
4	2	30	4	4.70	1.60	3	10	15	16.05	17
4	4	30	4	4.70	1.60	3	10	15	16.05	17
4	8	30	4	4.70	1.60	3	10	15	16.05	17

Table A.14: Parametric grid for the IEEE 57-bus solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	30	6	6.97	1.63	5	12	15	16.13	18
0.5	2	30	6	6.97	1.63	5	12	15	16.13	18
0.5	4	30	6	6.97	1.63	5	12	15	16.13	18
0.5	8	30	6	6.97	1.63	5	12	15	16.13	18
1	0.5	30	6	7.03	1.79	5	12	15	16.53	18
1	2	30	6	7.03	1.79	5	12	15	16.53	18
1	4	30	6	7.03	1.79	5	12	15	16.53	18
1	8	30	6	7.03	1.79	5	12	15	16.53	18
2	0.5	30	6	6.90	1.45	5	10	15	17.00	19
2	2	30	6	6.90	1.45	5	10	15	17.00	19
2	4	30	6	6.90	1.45	5	10	15	17.00	19
2	8	30	6	6.90	1.45	5	10	15	17.00	19
4	0.5	30	6	7.10	1.99	5	13	15	17.15	20
4	2	30	6	7.10	1.99	5	13	15	17.15	20
4	4	30	6	7.10	1.99	5	13	15	17.15	20
4	8	30	6	7.10	1.99	5	13	15	17.15	20

Table A.15: Parametric grid for the IEEE 118-bus solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	30	5	4.77	1.14	4	10	15	17.99	30
0.5	2	30	5	4.77	1.14	4	10	15	17.99	30
0.5	4	30	5	4.77	1.14	4	10	15	17.99	30
0.5	8	30	5	4.77	1.14	4	10	15	17.99	30
1	0.5	30	4	4.30	0.47	4	5	17	18.20	20
1	2	30	4	4.30	0.47	4	5	17	18.20	20
1	4	30	4	4.30	0.47	4	5	17	18.20	20
1	8	30	4	4.30	0.47	4	5	17	18.20	20
2	0.5	30	4	4.00	0.00	4	4	18	18.51	20
2	2	30	4	4.00	0.00	4	4	18	18.51	20
2	4	30	4	4.00	0.00	4	4	18	18.51	20
2	8	30	4	4.00	0.00	4	4	18	18.51	20
4	0.5	30	4	4.00	0.00	4	4	18	18.73	21
4	2	30	4	4.00	0.00	4	4	18	18.73	21
4	4	30	4	4.00	0.00	4	4	18	18.73	21
4	8	30	4	4.00	0.00	4	4	18	18.73	21

Table A.16: Parametric grid for the IEEE 300-bus solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	15	18	19.13	2.39	15	23	18	23.54	30
0.5	2	15	18	19.13	2.39	15	23	18	23.54	30
0.5	4	15	18	19.13	2.39	15	23	18	23.54	30
0.5	8	15	18	19.13	2.39	15	23	18	23.54	30
1	0.5	29	18	18.90	2.18	15	23	19	23.41	30
1	2	29	18	18.90	2.18	15	23	19	23.41	30
1	4	29	18	18.90	2.18	15	23	19	23.41	30
1	8	29	18	18.90	2.18	15	23	19	23.41	30
2	0.5	29	17	17.14	1.46	15	20	19	23.74	30
2	2	29	17	17.14	1.46	15	20	19	23.74	30
2	4	29	17	17.14	1.46	15	20	19	23.74	30
2	8	29	17	17.14	1.46	15	20	19	23.74	30
4	0.5	27	17	16.93	1.49	14	21	17	24.17	30
4	2	27	17	16.93	1.49	14	21	17	24.17	30
4	4	27	17	16.93	1.49	14	21	17	24.17	30
4	8	27	17	16.93	1.49	14	21	17	24.17	30

Table A.17: Parametric grid for the REAL-A solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	26	8	8.77	1.31	7	12	14	18.16	30
0.5	2	26	8	8.77	1.31	7	12	14	18.16	30
0.5	4	26	8	8.77	1.31	7	12	14	18.16	30
0.5	8	26	8	8.77	1.31	7	12	14	18.16	30
1	0.5	30	8	9.13	2.26	7	18	14	18.57	30
1	2	30	8	9.13	2.26	7	18	14	18.57	30
1	4	30	8	9.13	2.26	7	18	14	18.57	30
1	8	30	8	9.13	2.26	7	18	14	18.57	30
2	0.5	28	8	8.89	1.23	8	13	14	19.20	30
2	2	28	8	8.89	1.23	8	13	14	19.20	30
2	4	28	8	8.89	1.23	8	13	14	19.20	30
2	8	28	8	8.89	1.23	8	13	14	19.20	30
4	0.5	30	8	9.43	2.49	7	19	15	19.63	30
4	2	30	8	9.43	2.49	7	19	15	19.63	30
4	4	30	8	9.43	2.49	7	19	15	19.63	30
4	8	30	8	9.43	2.49	7	19	15	19.63	30

Table A.18: Parametric grid for the REAL-R solved by the Sℓ_1QP Method.

Δ_0	η_0	CC	Mo	$\overline{k}$	σ	$k_{\min}$	$k_{\max}$	Sℓ_1QP $j_{\min}$	$\overline{j}$	$j_{\max}$
0.5	0.5	30	5	4.57	0.63	3	6	15	17.25	30
0.5	2	30	5	4.57	0.63	3	6	15	17.25	30
0.5	4	30	5	4.57	0.63	3	6	15	17.25	30
0.5	8	30	5	4.57	0.63	3	6	15	17.25	30
1	0.5	30	5	4.57	0.63	3	6	16	17.83	30
1	2	30	5	4.57	0.63	3	6	16	17.83	30
1	4	30	5	4.57	0.63	3	6	16	17.83	30
1	8	30	5	4.57	0.63	3	6	16	17.83	30
2	0.5	30	5	4.57	0.63	3	6	17	18.72	30
2	2	30	5	4.57	0.63	3	6	17	18.72	30
2	4	30	5	4.57	0.63	3	6	17	18.72	30
2	8	30	5	4.57	0.63	3	6	17	18.72	30
4	0.5	30	5	4.57	0.63	3	6	17	19.15	30
4	2	30	5	4.57	0.63	3	6	17	19.15	30
4	4	30	5	4.57	0.63	3	6	17	19.15	30
4	8	30	5	4.57	0.63	3	6	17	19.15	30