Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Matias Lunde Ellingsen

Security-Constrained Optimal Power Flow in Meshed Distribution Grids

Master's thesis in Energy and Environmental Engineering
Supervisor: Olav Bjarte Fosso
June 2020

NTNU
Norwegian University of Science and Technology

Matias Lunde Ellingsen

Security-Constrained Optimal Power Flow in Meshed Distribution Grids

Master's thesis in Energy and Environmental Engineering
Supervisor: Olav Bjarte Fosso
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

With the modernization of the power grid, smart solutions for effective and secure system planning and operation are essential. Traditional power system optimization techniques are being revisited and becoming relevant for new applications. Security-Constrained Optimal Power Flow (SCOPF) is an extension of the Optimal Power Flow (OPF) problem. The objective of SCOPF is usually to find the lowest cost dispatch of power that satisfy all system constraints, now, and during a defined set of likely contingencies, such as line- or generator outages. The majority existing SCOPF software either rely on a DC approximation, which is unsuitable especially for distribution grids, is not freely available, or don't consider post-contingency rescheduling. The objective of this thesis is to identify a solution method for the nonlinear SCOPF problem and make a prototype implementation in the open-source programming language Python.

Based on a selection of available literature, the basics of the SCOPF problem are laid out and a solution approach is identified and presented. The OPF is solved by Sequential Linear Programming, utilizing a Trust Region Method. With this approach, the original nonlinear problem is iteratively linearized around the current solution and solved by Linear Programming. The Trust Region method adjusts the "window" for which the linearizations are assumed to be valid based on the accuracy of the previous iteration. To solve the SCOPF problem, where contingency constraints are included, Benders Decomposition is employed. That involves dividing the problem into a base-case master problem and a set of slave subproblems for each considered contingency. These are also solved iteratively, with infeasible subproblems generating a linear constraint for the master problem, known as a Bender's Cut. The algorithm is designed to consider both *preventive security* and *corrective security*.

The program is tested and demonstrated with two illustrative systems with some numerical examples. The proposed OPF algorithm performs well on the 6-bus example system, where convergence is achieved in 9 iterations with a cost reduction of 5.45 % compared to the initial "guess". The SCOPF algorithm is demonstrated on an example 12-bus system. Preventive and corrective security result in a cost increase of 1.25 % and 0.41 %, respectively, compared to the basic OPF solution. Tests of the considered contingencies reveal that the solutions hold, but that the corrective solution can be inaccurate for larger allowed corrective rescheduling. It's concluded that the methods employed here for solving SCOPF problems show great promise, and with further work can become useful tools in the planning and operation of future distribution grids.

Sammendrag

Når kraftnettet nå moderniseres, trengs det smarte løsninger for effektiv og sikker planlegging og drift av systemene. Tradisjonelle optimeringsteknikker får fornyet oppmerksomhet siden de nå blir relevante i nye anvendelser. "Security-Constrained Optimal Power Flow" (SCOPF) er en utvidelse av optimal lastflyt (OPF). Hensikten med SCOPF er vanligvis å finne den mest kostnadseffektive fordelingen av kraftproduksjon i et system, samtidig som alle begrensninger overholdes, både nå og under en rekke eventualiteter, som utfall av enkelte linjer eller generatorer. Brorparten av eksisterende SCOPF programvare lener seg enten på DC tilnærminger, som ikke egner seg for distribusjonsnett, er ikke åpent tilgjengelig, eller tar ikke hensyn til muligheten for korrigerende handlinger ved inntrufne eventualiteter.

Med utgangspunkt i et utvalg tilgjengelig litteratur, et grunnlag for forståelse og løsning av SCOPF problemet presenteres, og en spesifikk løsningsstrategi identifiseres og legges fram. Optimal lastflyt-problemet løses ved sekvensiell lineær programmering, inkludert en tillitsregion-metode. Med denne tilnærmingen lineariseres problemet rundt en nåværende løsning og løses iterativt med lineær programmering. Tillitsregion-metoden justerer området hvor lineariseringen er antatt å være gyldig med utgangspunkt i nøyaktigheten av forrige iterasjons løsning. For å løse SCOPF-problemet, hvor begrensninger som følge av eventualiteter også må tas med, brukes Benders dekomposisjon. Det involverer å dekomponere problemet til et masterproblem og en rekke underproblemer for hver eventualitet. Disse løses også iterativt, og fra underproblemer hvor det oppdages overskridelser genereres det lineære begrensinger som legges til masterproblemet. Den presenterte algoritmen er designet til å kunne vurdere både *forebyggende sikkerhet* og *korrigerende sikkerhet*.

Programmet testes og demonstreres på to illustrerende systemer med noen numeriske eksempler. Den foreslåtte optimal lastflyt-algoritmen presterer bra på eksempelsystemet med 6 samleskinner. Her kreves 9 iterasjoner, og løsningen gir en kostnadsreduksjon på 5,45 % sammenlignet med utgangspunktet. SCOPF-algoritmen demonstreres på et eksempelsystem med 12 samleskinner. Forebyggende og korrigerende sikkerhet resulterer i en kostnadsøkning på henholdsvis 1,25 % og 0.41 % sammenlignet med standard optimal lastflytløsning. Tester av de vurderte eventualitetene viser at løsningene holder, men at løsningen ved korrigerende sikkerhet kan være unøyaktig hvis større korrigerende handlinger tillates. Det konkluderes med at de brukte metodene virker lovende, og med videre arbeid kan bli nyttige verktøy i planlegging og drift av fremtidens distribusjonsnett.

Preface

This master thesis concludes my studies at the Department of Electric Power Engineering at the Norwegian University of Science and Technology. I'm grateful for the opportunity to conduct my education here. The past five years have been both challenging and rewarding, and I'm proud of my achievements. It's been a special semester to complete my time as a student. With the corona-virus restrictions denying access to campus, it's been a quite different semester than we all had expected. This period has highlighted the value of having a vibrant environment of curious and like-minded fellow students to discuss and share ideas with, as this is what I have missed the most.

The work on this thesis has involved a great deal of programming, and the code is handed in separately to my supervisor.

I would like to express my sincerest gratitude to my supervisor Professor Olav B. Fosso for his competent guidance and for sharing his knowledge and expertise.

Finally, I would like to thank my family for their unwavering support in my endeavours.

Trondheim, June 2020

Matias Lunde Ellingsen

Table of Contents

List of Tables

List of Figures

Abbreviations

DG	=	Distributed Generation
RES	=	Renewable Energy Sources
SG	=	Smart Grid
LP	=	Linear Programming
QP	=	Quadratic Programming
NLP	=	Nonlinear Programming
AC	=	Alternating Current
DC	=	Direct Current
LF	=	Load Flow
PF	=	Power Flow
ED	=	Economic Dispatch
OPF	=	Optimal Power Flow
SCOPF	=	Security Constrained Optimal Power Flow
PSCOPF	=	Preventive Security Constrained Optimal Power Flow
CSCOPF	=	Corrective Security Constrained Optimal Power Flow
BD	=	Benders Decomposition
BC	=	Benders Cut
TR	=	Trust Region

Chapter 1

Introduction

1.1 Background and Motivation

Modern society relies on secure access to electrical energy and power, and this dependency is increasing as we move towards a Low Carbon Society. Rapid changes in how we produce and consume electricity set new demands for the power grid. Distributed Generation (DG) from Renewable Energy Sources (RES) is increasing, and the electrification of sectors such as transport is changing the consumption patterns. To meet these demands, large investments and incorporation of new technology are needed. Estimates suggest that investments in the Norwegian power grid will amount to 135 billion NOK in the period 2018 to 2027 [1]. It's therefore paramount to secure efficiency in every layer of power system planning and operation while maintaining high system security.

In the last decades, the world has seen tremendous technological advancements driven by the expanding use of information and communication technology. The power grid, however, has been lagging behind. This is however about to change, as measures are being taken to move in the direction of Smart Grids (SG). The term refers to the modernisation of the power grid, including the integration of new technologies such as dispersed generation, dispatchable loads, communication systems and energy storage to efficiently deliver sustainable, economic and secure electricity [2]. A part of this is the division of distribution grids into smaller controllable entities known as microgrids [3]. In light of this development, traditional optimisation techniques have recently received increased attention in the research community. They must evolve in order to meet the requirements of new demands.

Security-Constrained Optimal Power Flow (SCOPF) is an extension of the Optimal Power Flow (OPF) problem [4], which is one of the most well-researched and

practically important sub-fields of constrained nonlinear optimisation [5]. A solution to the OPF problem finds the optimal operating state of a power system given a chosen objective while meeting all current system constraints. It's most defining feature is the inclusion of the Power Flow (PF) equations as equality constraints [5]. The problem was first introduced by Carpentier in 1962 [6] and has since been extensively covered in the literature. About a decade later, the extension known as SCOPF was presented [7]. This formulation also includes constraints on the pre-contingency operation, considering system operating feasibility during a set of postulated contingencies [4]. The goal is normally to find the lowest cost operating state of the system while maintaining an n-1 security level. In 1987, the authors of [8] described a mathematical framework for also taking into account the system corrective capabilities.

While there exist several strategies for solving the SCOPF problem, it's generally considered a non-linear, non-convex, large-scale, optimisation problem, with continuous and discrete variables, placing it in the category of mixed-integer nonlinear programming (MINLP) [4]. The main challenge in an optimization perspective is the handling of the constraints. In practical applications, such as electricity pricing in some markets, an approximation known as DC SCOPF are used [4]. These methods usually make use of Linear Programming (LP) or Quadratic Programming (QP), which great advantage is the easy handling of constraints and wide availability. The complexity of the original problem has given such approaches great appeal, and it remains the standard form usually found in textbook examples such as in [9]. However, for the SCOPF solution to be reliable, especially in distribution grids, it's insufficient to use a purely linear system model. Over the years, a great number of nonlinear optimization methods have been proposed and used for solving OPF and SCOPF problems, including gradient methods [10], [7], newton type methods [11], sequential linear programming [12], [13], interior point methods[14], [15], and more recently, convex relaxation type methods [16], [17]. The immensity of the OPF literature and different approaches has also led to several state of the art surveys being published. The current state of the art SCOPF formulations and challenges are for instance reviewed in [4] and [18].

With the development of SG's, including the introduction of microgrids, it's necessary to develop smart algorithms for optimal and secure operation of the power system also in lower levels of the transmission and distribution network hierarchy. Traditional uses of SCOPF is typically related to higher-level transmission networks and large power plants. For those wanting to implement SCOPF in their system, it's a challenge that the available software is often commercial and difficult to customise. Much is Matlab-based, and as several companies, including Transmission- and Distribution Grid Operators (TSO and DSO), are transitioning towards more python-based software, developing a SCOPF solver in this environ-

ment is desirable. This is an area where a lot of work is required, but a good starting point is to demonstrate how a basic non-linear SCOPF algorithm can be formulated and implemented in Python, only using freely available software.

1.2 Objective

The objective of this thesis is to identify a solution strategy to the non-linear SCOPF problem that can apply to distribution grids, make a prototype implementation in Python, and demonstrate it on an example system. The program should handle multiple contingency constraints, and should consider both ”preventive security” and ”corrective security”. It should take an object-oriented design, and be general in the sense that it should handle different, though simplified, system topologies. Success factors and possible challenges in a generalized implementation ought to be discussed. The work should be a building block in the toolbox for planning and operation of microgrids.

1.3 Structure of Thesis

Chapter 1 - *Introduction*, provides the reader with the work's motivational background and places it in a scientific context. The objective of the thesis is also clearly stated here.

Chapter 2 - *Basic Theory*, gives a presentation of some fundamental concepts that are needed for analysing and solving the problem.

Chapter 3 - *Problem Formulation and Solution Method*, formulates the problem in mathematical terms and explains the solution approach.

Chapter 4 - *Numerical Examples*, presents two numerical examples that test and demonstrate the developed program.

Chapter 5 - *Discussion*, contains a general discussion on the chosen methods and the performance of the program, focusing on success factors and possible challenges.

Chapter 6 - *Conclusion and Further Work*, concludes the thesis and provides suggestions for further work.

Chapter 2

Basic Theory

This chapter explains some important concepts for the solution of the SCOPF problem. It should serve as an orientation and lay the necessary foundation for the solution method presented in chapter 3.

2.1 Power Flow Analysis

In this section, some important concepts of Power Flow (PF) Analysis, or Load Flow (LF), is presented. A PF study is essentially the calculation of the voltage magnitudes and angles of all buses in a power system under balanced, three-phase, steady-state conditions, and it's performed for various reasons. The fact that it's a steady-state method means that the solution describes the system in a single instance, and therefore only the static conditions are important. As its input, it takes the load and generation data, and the system topology in the form of a single line diagram, including the line impedances. As a by-product of the established voltage magnitudes and angles, real and reactive power quantities, like branch flows or net bus injections, can be obtained. [19]

2.1.1 Power System Representation

An electrical power system may be modelled as a network consisting of buses (nodes), interconnected by branches, representing for instance transmission lines, cables or transformers. The buses, which represent physical points of interconnection, is referenced by a node index $i \in \boldsymbol{N}$. Branches, on the other hand, are referenced as arcs between two nodes $(i, j) \in \boldsymbol{L}$, where $i, j \in \boldsymbol{L}$. The system size can be described by the number of buses $N = |\boldsymbol{N}|$, and the number of lines $L = |\boldsymbol{L}|$. Each bus i has an associated complex voltage $|V_i|\angle\theta_i$. Branches can be

described by their admittance y_{ij}. A neat way to represent the connectivity of the network, including the admittances, is through the bus-admittance matrix. [5]

2.1.2 Bus-admittance matrix

The network topology, assuming a pi-model of the transmission lines, can be represented mathematically by the bus-admittance matrix[5]:

$$\boldsymbol{Y_{bus}} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \tag{2.1}$$

The diagonal elements of $\boldsymbol{Y_{bus}}$, the self admittances, are calculated as[20]:

$$Y_{ii} = y_{i0} + \sum_{j=1, j\neq i}^{n} y_{ij}, \tag{2.2}$$

and the off-diagonal elements as[20]:

$$Y_{ij} = -y_{ij}, \tag{2.3}$$

where y_{ij} is the admittance of the line between bus i and bus k, and y_{i0} is the shunt admittance on bus i. The elements of the Y_{bus}-matrix are complex numbers, which in rectangular coordinates can be expressed as:

$$Y_{ij} = G_{ij} + jB_{ij} \tag{2.4}$$

The $\boldsymbol{Y_{bus}}$-matrix can therefore be split into the two matrices $\boldsymbol{G_{bus}}$ and $\boldsymbol{B_{bus}}$, representing the real and imaginary part of $\boldsymbol{Y_{bus}}$, respectively. For further details of the construction of $\boldsymbol{Y_{bus}}$ see for instance [20] or [5].

2.1.3 Bus-classification

Each bus i in the system is characterised by the four variables; active power injection P_i, reactive power injection Q_i, voltage magnitude $|V_i|$ and voltage angle θ_i. For a given system, two of these variables are specified, and two must be calculated. Which two variables that are specified, determine the classification, and the three main classes are[20]:

- **Load bus (PQ):** A bus where only loads are connected, and hence the active- and reactive power injection (demand) P_i and Q_i are known. For this reason, it's also called a PQ-bus. The voltage magnitude and angle, $|V_i|$ and θ_i, must be calculated.

- **Generator bus (PV):** A bus that has a generator connected to it. A synchronous generator can vary the active power output P_i by changing the prime mover input, as well as controlling the voltage magnitude $|V_i|$ by changing the excitation current in the field winding. For this reason, it's also called a voltage-controlled- or PV-bus. The voltage angle θ_i and reactive power Q_i are unknown and must be calculated.

- **Slack bus:** A bus with a generator must always be assigned as the slack bus. It has both a physical and a mathematical purpose. Before the PF is computed, the losses in the network are unknown. The generator on the slack bus is assigned to compensate for these losses, and hence the active- and reactive power, P_i and Q_i, are unknown. It also serves as a reference for the voltages in the system, with the voltage angle θ_i specified to be zero. Hence it's also called the "reference bus". The known variables are therefore $|V_i|$ and θ_i, while P_i and Q_i must be calculated.

2.1.4 The Power Flow Equations

The AC PF equations are a set of real-valued simultaneous equations that is applicable for mathematical programming formulations. [5] The two main ways of representing the network are the bus injection (nodal) model and the branch flow model. While the branch flow model has gained recent popularity for its applicability to convex relaxation[21], the bus injection model is the most compact and commonly used. [5] In this thesis the bus-injection model will be used.

Applying Kirchhoff's current law and using Y_{bus}, the nodal equations for a power system network can be written as:

$$\boldsymbol{I} = \boldsymbol{Y_{bus}}\boldsymbol{V}, \tag{2.5}$$

where $\mathbf{I}$ is a vector of the current injected at each node, and $\mathbf{V}$ is a vector of all the bus voltages. For bus i, the nodal equation is then:

$$I_i = \sum_{j=1}^{n} Y_{ij}V_j, \tag{2.6}$$

where n is the total number of buses. The complex power injected at bus i is

$$S_i = P_i + jQ_i = V_i I_i^* \tag{2.7}$$

[19] Writing the voltages in polar coordinates as $|V|\angle\theta$ and the admittance in rectangular coordinates as in (2.4), inserting (2.6) into (2.7), and separating into

real and imaginary part, the PF equations are obtained as:

$$P_i(V,\theta) = |V_i| \sum_{j=1}^{n} |V_j|[G_{ij}cos(\theta_i - \theta_j) + B_{ij}sin(\theta_i - \theta_j)] \tag{2.8}$$

$$Q_i(V,\theta) = |V_i| \sum_{j=1}^{n} |V_j|[G_{ij}sin(\theta_i - \theta_j) - B_{ij}cos(\theta_i - \theta_j)] \tag{2.9}$$

In a PF solution, these equations are simultaneously solved for all the buses in the system, except the reference bus. That constitutes $2(n-1)$ nonlinear equations, with variables P_i, Q_i, $|V_i|$ and $\angle\theta_i$. These must be solved numerically by an iterative approach. [20]

While Equations (2.8) and (2.9) describe the net injected power on a specific bus i, the active and reactive power flow on a specific branch ij can be expressed as:

$$P_{ij} = |V_i|^2 G_{ij} - |V_i||V_j|[G_{ij}cos(\theta_i - \theta_j) + B_{ij}sin(\theta_i - \theta_j)], \tag{2.10}$$

$$Q_{ij} = -|V_i|^2 G_{ij} - |V_i||V_j|[G_{ij}sin(\theta_i - \theta_j) - B_{ij}cos(\theta_i - \theta_j)]. \tag{2.11}$$

For ease of notation, the magnitude sign for the voltage is skipped from this point on, meaning that V will always mean voltage magnitude, while the angle is treated separately as θ unless otherwise specified.

2.1.5 The Newton-Raphson Method

There are several numerical methods for solving the PF problem, with the Gauss-Seidel and the Newton-Raphson (NR) methods being the most common. While the Gauss-Seidel method can be relatively fast, especially for smaller systems, the reliable NR method is the primary engine in PF analysis. Independent of the system size, it converges in 2-5 iterations from a flat start ($|V| = 1.0p.u., \theta = 0$). [20]

The NR method is well covered in the power systems literature [9][19][20][22], and a summary is presented here. The PF equations (2.8) and (2.9), can be approximated by their first order Taylor series around the current estimate of V and θ as:

$$\begin{bmatrix} \boldsymbol{\Delta P} \\ \boldsymbol{\Delta Q} \end{bmatrix} \approx [\boldsymbol{J}] \begin{bmatrix} \boldsymbol{\Delta \theta} \\ \boldsymbol{\Delta V} \end{bmatrix}, \tag{2.12}$$

where $\boldsymbol{J}$ is the jacobian matrix. At every iteration, the mismatches ΔP and ΔQ are calculated as:

$$\Delta P_i = (P_i^G - P_i^L) - P_i(V,\theta), \tag{2.13}$$

$$\Delta Q_i = (Q_i^G - Q_i^L) - Q_i(V,\theta), \tag{2.14}$$

with L and G meaning generated power and load, respectively. [5] The correction vector for voltage and angle can then be calculated according to equation (2.12). As matrix inversion is very inefficient [20], a lower-upper(LU)-decomposition and forward and backward substitution algorithm is normally used [23]. The voltage magnitudes and angles are then updated by their corresponding corrections, before a new iteration is started. The algorithm is stopped when the calculated mismatch falls under a predefined threshold [22].

Including the PF equations of all the buses in the system would lead to a trivial set of voltage equations. The $\boldsymbol{\theta}$- and $\boldsymbol{V}$-vectors are simplified to only include the unknown values as:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \tag{2.15}$$

$$\boldsymbol{V} = \begin{bmatrix} V_{i_1} \\ \vdots \\ V_{i_m} \end{bmatrix}, \tag{2.16}$$

where n is the total number of buses and m is the number of load buses, and assuming the slack bus is bus number one. The set of load flow equations to solve then becomes [22]:

$$P_i^s = P_i(\theta, V); i = 1, ..., n, \tag{2.17}$$

$$Q_{i_l}^s = Q_{i_l}(\theta, V); l = 1, ..., m, \tag{2.18}$$

with the superscript s referring to *specified* values. The Jacobian matrix can then be expressed as:

$$\boldsymbol{J}(\boldsymbol{\theta}, \boldsymbol{V}) = \begin{bmatrix} \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} \\ \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} \end{bmatrix}, \tag{2.19}$$

where:

$$\boldsymbol{P} = \begin{bmatrix} P_2 \\ \vdots \\ P_n \end{bmatrix}, \tag{2.20}$$

$$\boldsymbol{Q} = \begin{bmatrix} Q_{i_1} \\ \vdots \\ Q_{i_m} \end{bmatrix}. \tag{2.21}$$

If one introduces the following composite terms:

$$T_{ij} \equiv G_{ij}cos(\theta_i - \theta_j) + B_{ij}sin(\theta_i - \theta_j), \tag{2.22}$$

$$U_{ij} \equiv G_{ij}sin(\theta_i - \theta_j) - B_{ij}cos(\theta_i - \theta_j), \tag{2.23}$$

one may express the different terms of the jacobian matrix directly as:

$$\frac{\partial P_i}{\partial \theta_i} = V_i \sum_{j \neq i} V_j U_{ij} \tag{2.24}$$

$$\frac{\partial P_i}{\partial \theta_j} = -V_i V_j U_{ij} \tag{2.25}$$

$$\frac{\partial P_i}{\partial V_i} = 2V_i G_{ii} - \sum_{j \neq i} V_j T_{ij} \tag{2.26}$$

$$\frac{\partial P_i}{\partial V_j} = -V_i T_{ij} \tag{2.27}$$

$$\frac{\partial Q_i}{\partial \theta_i} = -V_i \sum_{j \neq i} V_j T_{ij} \tag{2.28}$$

$$\frac{\partial Q_i}{\partial \theta_j} = V_i V_j T_{ij} \tag{2.29}$$

$$\frac{\partial Q_i}{\partial V_i} = -2V_i B_{ii} - \sum_{j \neq i} V_j U_{ij} \tag{2.30}$$

$$\frac{\partial Q_i}{\partial V_j} = -V_i U_{ij} \tag{2.31}$$

[22]

2.1.6 Simplified PF Methods

As the full AC PF can be quite computationally heavy and often inconvenient to solve, measures to simplify the problem for various application have been made.

Decoupled PF

A class of such methods is the decoupled PF methods, which rely on the following observations:

- For most transmission networks line and transformer reactances are much larger than the corresponding resistances, meaning:

$$|G_{ij}| \ll |B_{ij}|, \ i, j = 1, ..., n \tag{2.32}$$

- The difference in angle across a line or transformer, $(\theta_i - \theta_j)$, is quite small, and rarely exceeding $30°$. Then it follows that:

$$cos(\theta_i - \theta_j) \approx 1.0, \tag{2.33}$$

$$|sin(\theta_i - \theta_j)| \approx small. \tag{2.34}$$

As a result of these observations, it can be concluded that the terms $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}}$ and $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\theta}}$ are small compared to $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}}$.[22] Taking advantage of this by setting the smaller matrix blocks to equal zero, one may separate Newton updates for θ and V with correspondingly smaller matrices, saving computation time. The decoupled PF is locally convergent to the exact solution because the exact power mismatches $\boldsymbol{\Delta P}$ and $\boldsymbol{\Delta Q}$ are calculated using Equations (2.8) and (2.9) and both $\boldsymbol{\theta}$ and $\boldsymbol{V}$ are updated at each iteration, but more iterations are required. [5] A specific decoupled method is the Fast Decoupled Power Flow, which is a much used algorithm for when an approximate solution is needed. The approximation is completely independent of V and θ, but performs surprisingly well despite apparently drastic simplifications. [24][25]

DC Power Flow

An even further simplification of the PF problem, is the DC PF. Despite it's name, it has nothing to do with "Direct Current", but is just a linearization of the full AC PF equations. As these equations are nonlinear, iterative numerical methods are required. This demand is eliminated with a DC PF representation of the system. [5] Such a model has great appeal and is employed in numerous applications, including contingency screening, medium-to-long term transmission planning, and Locational Marginal Pricing-based market applications. [26] Following similar observations as described for the decoupled PF, the following assumptions are made for DC PF:

1. All branch resistances are assumed to be zero. That means the transmission is lossless, and all $G_{ij} = 0$.

2. The angular difference between buses are small, resulting in $sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$ and $cos(\theta_i - \theta_j) \approx 1$.

3. All voltage magnitudes are assumed equal to 1.0 p.u.

4. Reactive PF is neglected.

[5] The DC PF equation is obtained by applying these assumptions to equation (2.8), resulting in:

$$P_i(\theta) \approx \sum_{j=1}^{n} B_{ij}(\theta_i - \theta_j). \tag{2.35}$$

Similarly, the active power flow on a specific branch is approximated by:

$$P_{ij} \approx -B_{ij}(\theta_i - \theta_j). \tag{2.36}$$

[5]

While the DC PF model, under some circumstances, can give reasonably good estimates for the active PF in the system, it gives no information on the voltage magnitudes and reactive PF and should be used with care. A clear drawback is that the model produces significant errors for stressed systems, which is often when a PF solution is most needed. [26]

2.2 Optimal Power Flow

Optimal Power Flow (OPF) is a set of optimisation problems in electrical engineering first introduced by Carpentier [6] in 1962 as an extension to the Economic Dispatch (ED) problem [5]. An ED calculation finds the lowest-cost generation dispatch for a set of generators, only limited by the generator capabilities, and results in a total generation equal to the total load and losses. OPF takes this calculation a step further by also taking into account the effect the generation has on the transmission network, as well as the limitations caused by it. OPF is, therefore, a coupling between an ED and a PF calculation, solving both simultaneously. The losses are accounted for by the PF calculation, and the ED can be constrained by transmission limits such as MW or MVA flow limits on lines and voltage limits on buses. [9]

OPF is usually formulated as a minimisation problem, where the goal is to minimise some objective function $f(\boldsymbol{x}, \boldsymbol{u})$, subject to a set of equality constraints $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u})$, and inequality constraints $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u})$. In general terms, the problem can be formulated as:

$$\begin{aligned} min \quad & f(\boldsymbol{x}, \boldsymbol{u}), \\ s.t \quad & \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) = 0, \\ & \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) \leq 0, \end{aligned} \tag{2.37}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are vectors of state- and control variables respectively. [27]

2.2.1 Objective Function

As the name suggests, OPF aims to find the most optimal operating state of the power system. To find an optimal operating state, one must first decide on what should be optimised, which depends on the purpose of the OPF. Some example objectives are mentioned next.

- **Minimisation of cost:** The most common use of OPF is to minimise the cost of producing electricity. The objective function is then set to minimise

the total cost of generating active power [28], and can be formulated as:

$$min \sum_{i=1}^{N_{gen}} F_i(P_{Gi}), \tag{2.38}$$

where $F_i(P_{Gi})$ is the cost function of the generator at bus i. [9]

- **Minimisation of active and reactive power losses:** For a system operator, the minimisation of power losses is arguably one of the most important considerations, after ensuring system security. To facilitate this, the objective function can be formulated as:

$$min \sum_{i,j \in L} S_{ij} + S_{ji}. \tag{2.39}$$

[28]

- **Maintaining a constant voltage profile:** Sometimes, it might be important for a system operator to maintain a constant voltage profile, for instance to avoid voltage instability issues. An OPF can then identify the preferable set of actions to achieve this. The objective function could for instance be formulated as:

$$min \sum_{i=1}^{n} (V_i - V_{setpoint,i})^2 \tag{2.40}$$

[28]

2.2.2 Variables

As described in section 2.1, a PF calculation seeks to find the unknown voltage magnitudes and angles, for then to uniquely describes the system state. An OPF calculation, on the other hand, seeks to find the most optimal system state within a feasible region. As a consequence, some variables that in a PF calculation are held constant may be permitted to change in an OPF calculation. A general classification of the variables in an OPF calculation may, therefore, be as follows:

- **Control Variables** are variables that can be actively controlled by the system operator to achieve a desired operating state. In other words, they are the independent variables of the optimization problem. These are connected to the known variables in a PF calculation. Typical examples of such variables are the active and reactive power output of generators.

- **State variables** are the variables that together uniquely describe the system state, but cannot be directly controlled; hence can't be defined as control variables. These are influenced by the changes in the control variables and are hence the dependent variables of the optimization. The state variables are typically the voltage angles of all non-slack buses and voltage magnitudes at load buses.

- **Parameter values** are values that should be kept fixed. These are the known variables in a PF calculation that are not allowed to change in the OPF calculation either. This is usually always the voltage angle at the slack bus, which should be kept as a reference at zero, and typically also the active- and reactive power at load buses.

2.2.3 Constraints

The constraints are an essential part of OPF and are what defines the feasible region of the optimisation problem. As already mentioned, the constraints are typically divided into two categories: equality- and inequality constraints.

Equality Constraints

The equality constraints of an OPF are fulfilment of the PF Equations (2.8) and (2.9). These must be met for all buses for any operating point determined by the optimisation to be a true operating point of the power system. [28] The constraints can be written as:

$$P_i(V,\theta) = P_i^G - P_i^L \quad \forall\, i \in \boldsymbol{N} \tag{2.41}$$

$$Q_i(V,\theta) = Q_i^G - Q_i^L \quad \forall\, i \in \boldsymbol{N} \tag{2.42}$$

Inequality Constraints

The inequality constraints consist of all the operational limits of the system. These may vary between different formulations, but typical constraints to include are:

- **Active power constraints:** The active power produced at each generator must stay within their respective limits, typically specified as a minimum and a maximum production. If a bus has no generator, i.e. is a load bus, these limits are zero. Generally:

$$P_i^{G,min} \leq P_i^G \leq P_i^{G,max} \quad \forall\, i \in \boldsymbol{G} \tag{2.43}$$

- **Reactive power constraints:** Just as for the active power limits, the reactive power at a bus need to stay within specified limits. These could be decided by the reactive power capabilities of a connected generator or other controllable reactive power components. Different from the active power limits is that reactive power lower limit can be negative. Generally:

$$Q_i^{G,min} \leq Q_i^G \leq Q_i^{G,max} \quad \forall\, i \in \boldsymbol{G} \tag{2.44}$$

- **Voltage magnitude constraints:** The voltage at a specific bus is not allowed to exceed a certain limit around the nominal value. This limit may vary by country and voltage level, but is typically in the range +/- 5-10 %. Generally:

$$V_i^{min} \leq V_i \leq V_i^{max} \quad \forall\, i \in \boldsymbol{N} \tag{2.45}$$

- **Voltage angle constraints:** While too large shifts in voltage angle between buses in a realistic system shouldn't be an issue, several nonlinear solvers need to limit the voltage angle to avoid obtaining the same PF solution at intervals of $360°$. Assuming radians are used, this can be done by limiting θ to stay within the interval $[-\pi,\ \pi]$. [28] Generally:

$$\theta_i^{min} \leq \theta_i \leq \theta_i^{max} \quad \forall\, i \in \boldsymbol{N} \tag{2.46}$$

- **Branch flow constraints:** An important group of constraints in a power system is the power transfer ability of the branches. These limit the amount of power that can flow between two nodes in the network. There are different possibilities for which variables to use for constraining the branch flows, and the choice usually depends on what is the limiting factor. Shorter transmission lines are usually limited by their thermal limits, expressed by the maximum allowed current. Many OPF formulations, however, uses apparent power flow limits instead, as this can be more convenient and are closely correlated to the current. For transformers, it's usually also practical to use apparent power, as that's how they are generally rated. A third option is to limit the branch flow using active power. This is preferable for very long lines when the limiting factor becomes the steady-state stability limit of the line. [28] The branch flow constraints could, therefore, be formulated as one of the following:

$$I_{ij}^{min} \leq I_{ij} \leq I_{ij}^{max} \quad \forall\, (i,j) \in \boldsymbol{L} \tag{2.47}$$

$$S_{ij}^{min} \leq S_{ij} \leq S_{ij}^{max} \quad \forall\, (i,j) \in \boldsymbol{L} \tag{2.48}$$

$$P_{ij}^{min} \leq P_{ij} \leq P_{ij}^{max} \quad \forall\, (i,j) \in \boldsymbol{L} \tag{2.49}$$

2.2.4 DCOPF

The OPF problem, as it's been described thus far, is often called ACOPF. This comes from the fact that the full AC PF equations are considered, and that there exists a much-used alternative algorithm which uses the DC PF formulation described in section 2.1.6, called DCOPF. A great advantage of DCOPF is that the constraints become linear and the problem is convex [28]. It may be solved using relatively simple optimization techniques such as QP or LP [9].

The DCOPF formulation, however, comes with the same limitations as the DC PF itself; it's an approximation, and the accuracy of the result depends on the validity of the assumptions. For distribution grids, in particular, these assumptions are usually quite inaccurate, as the line resistances are relatively large and voltage variations are common [28]. As mentioned in section 2.1.6, stressed systems are also poorly reassembled by the DC PF approximation.

This thesis will focus on the ACOPF problem, and the DCOPF algorithm is not covered in depth. It is, however, important to be aware of its existence, usage and limitations when considering OPF, because of its wide use and central position it the overall OPF literature. It's also a good starting point for those new to OPF, as it is quite easily solved. The interested reader may see [9], or several other sources, for further details on the formulation and solution of the DCOPF problem.

2.3 Security-Constrained Optimal Power Flow

In many parts of the power system, it's required to operate the system under a so-called n-1 security level. That means the system should continue to operate safely during a predefined set of likely contingencies, not occurring simultaneously. A contingency is an event that removes a line or generator from the system, leaving the rest of the network in a more stressed condition [5]. This requirement has led to the development of SCOPF, as the solution to the original OPF problem cannot guarantee the feasibility of the operating state during contingencies. The inclusion of contingency constraints increases the size of the problem significantly and makes it much harder to solve. The solution will under no circumstances be associated with a lower cost than the solution to the OPF problem without contingency constraints, meaning the increase in security comes at a cost.

The general formulation of the OPF problem, Equation (2.37), can for the

SCOPF case be extended as:

$$\begin{aligned} min \;\; & f(\boldsymbol{x_0}, \boldsymbol{u}), \\ s.t \;\; & \boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{u}) = 0, \\ & \boldsymbol{h}(\boldsymbol{x_0}, \boldsymbol{u}) \leq 0, \\ & \boldsymbol{g}(\boldsymbol{x_c}, \boldsymbol{u}) = 0 \quad \forall\, c \in \boldsymbol{C}, \\ & \boldsymbol{h}(\boldsymbol{x_c}, \boldsymbol{u}) \leq 0 \quad \forall\, c \in \boldsymbol{C}, \end{aligned} \tag{2.50}$$

where $\boldsymbol{C} = \{1, ..., N_c\}$ is the set of considered contingencies. [5] For the solution of the SCOPF problem to be feasible, the same equality and inequality constraints as in the original OPF problem must hold true under all the contingencies in $\boldsymbol{C}$.

2.3.1 Post-Contingency Corrective Rescheduling

Solutions to the SCOPF problem as defined in (2.50) would lead to implementation of preventive control actions. That's because it's demanded that the system is to remain in a feasible state during contingency, without doing any corrective actions. If, for instance, it's possible to ramp up or down a generator's power output by a certain amount before an overload becomes critical, it would be possible to achieve the same level of security, but at a lower operating cost. This fact is pointed out in [8], and a solution strategy to the SCOPF problem with post-contingency rescheduling is presented. The range of rescheduling actions are represented by coupling constraints of the type $|u_0 - u_c| \leq \Delta_c$. The problem would then be formulated as:

$$\begin{aligned} min \;\; & f(\boldsymbol{x_0}, \boldsymbol{u_0}), \\ s.t \;\; & \boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{u_0}) = 0, \\ & \boldsymbol{h}(\boldsymbol{x_0}, \boldsymbol{u_0}) \leq 0, \\ & \boldsymbol{g}(\boldsymbol{x_c}, \boldsymbol{u_c}) = 0 \quad \forall\, c \in \boldsymbol{C}, \\ & \boldsymbol{h}(\boldsymbol{x_c}, \boldsymbol{u_c}) \leq 0 \quad \forall\, c \in \boldsymbol{C}, \\ & |\boldsymbol{u_0} - \boldsymbol{u_c}| \leq \boldsymbol{\Delta_c} \quad \forall\, c \in \boldsymbol{C}, \end{aligned} \tag{2.51}$$

where $\boldsymbol{u_0}$ are the control variables during base case, $\boldsymbol{u_c}$ are the control variables during contingency c, and $\boldsymbol{\Delta_c}$ are the allowed changes in control variables from the base case to the contingency case. Requiring that $\boldsymbol{\Delta} = 0$, would then be the same as requiring preventive security. Henceforth, preventive SCOPF will be denoted PSCOPF and SCOPF with post-contingency corrective rescheduling as CSCOPF. In [8], the CSCOPF problem is proposed solved using a Bender Decomposition (BD) approach.

2.3.2 Benders Decomposition

BD is a method that has been widely used for solving various SCOPF problems since it was generally formulated in [29]. The BD approach consists of decomposing the problem into a master problem and several subproblems, interacting iteratively. The possibility of keeping both the master- and subproblem very tractable makes it especially appealing. BD also makes it possible to distribute the computation across several processors. [4] A drawback, however, is that convergence of the BD algorithm only can be guaranteed if the convexity of the feasible region is assumed. This is not necessarily true in an AC SCOPF, so BD should be used with care. [29]

The methodology for solving the CSCOPF problem, as it's described in [8], and using the notation of Equation (2.51), consists of a two stage process:

- Find an operating point $(\boldsymbol{x_0}, \boldsymbol{u_0})$ for the "base case" problem (2.37).
- Given the the operating point $(\boldsymbol{x_0}, \boldsymbol{u_0})$, find new operating points $(\boldsymbol{x_c}, \boldsymbol{u_c})$ that meets the constraints $\boldsymbol{g}(\boldsymbol{x_c}, \boldsymbol{u_c}) = 0$ and $\boldsymbol{h}(\boldsymbol{x_c}, \boldsymbol{u_c}) \leq 0$ and the coupling constraints $|\boldsymbol{u_0} - \boldsymbol{u_c}| \leq \boldsymbol{\Delta_c}$ for all contingencies in $\boldsymbol{C}$.

The goal is to minimise the operation cost while making sure the problems in the second stage are feasible. This is done by separately considering the "base-case" operation and the N_c post-contingency operating states. If the post-contingency state is feasible, no change to the "base-case" is needed. On the other hand, if the post-contingency subproblem leads to infeasibilities, constraints on the "base-case" operation must be added in order to secure feasibility of the subproblem.

The subproblem can be formulated as:

$$\begin{aligned} w(\boldsymbol{x_0}, \boldsymbol{u_0}) = \ & min \ \ \boldsymbol{d^r} \cdot \boldsymbol{r} + \boldsymbol{d^s} \cdot \boldsymbol{s} \\ & s.t \ \ \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{r} = 0, \\ & \quad \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{s} \leq 0, \\ & \quad |\boldsymbol{u_0} - \boldsymbol{u}| - \boldsymbol{s} \leq \boldsymbol{\Delta}, \end{aligned} \tag{2.52}$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are positive vectors of penalty variables for the operating- and coupling constraints, respectively, and $\boldsymbol{d^r}$ and $\boldsymbol{d^s}$ are positive cost vectors. The value of the objective function, w, can be seen, and are here written, as a function of the "base-case" operating point. It can be concluded from problem (2.52) that:

$$\begin{aligned} w = 0 \quad &\Leftrightarrow \quad \text{The subproblem is feasible} \\ w > 0 \quad &\Leftrightarrow \quad \text{The subproblem is not feasible} \end{aligned}$$

Requiring that $w_c(\boldsymbol{x_0}, \boldsymbol{u_0}) \leq 0$ for all contingencies c in $\boldsymbol{C}$, would then be equivalent to requiring the feasibility of the post-contingency subproblems. [8] The

CSCOPF problem (2.51) can then be rewritten as:

$$\begin{aligned} min \;\; & f(\boldsymbol{x_0}, \boldsymbol{u_0}), \\ s.t \;\; & \boldsymbol{g}(\boldsymbol{x_0}, \boldsymbol{u_0}) = 0, \\ & \boldsymbol{h}(\boldsymbol{x_0}, \boldsymbol{u_0}) \leq 0, \\ & w_c(\boldsymbol{x_0}, \boldsymbol{u_0}) \leq 0 \quad \forall\, c \in \boldsymbol{C}, \end{aligned} \tag{2.53}$$

BD is a technique for approximating $w_c(\boldsymbol{x_0}, \boldsymbol{u_0})$. The approximation is improved by iteratively solving the "base-case" problem and the N_c operating subproblems. Associated with each subproblem solution, there is a set of Lagrange multipliers that tell how marginal changes in the base-case operating point $(\boldsymbol{x_0}, \boldsymbol{u_0})$ affect the infeasibility of the subproblem. These multipliers are, together with the value of w, used to form a linear constraint from a particular infeasible subproblem, which is fed to the base-case problem. This constraint is known as a Benders Cut (BC), and is written only in terms of the base-case variables $(\boldsymbol{x_0}, \boldsymbol{u_0})$. Modifying the approximation of $w_c(\boldsymbol{x_0}, \boldsymbol{u_0})$ corresponds to adding the BC to the base-case problem. [8]

2.4 Optimization

As already mentioned in the introduction, the SCOPF problem is a large-scale, nonlinear, non-convex optimization problem, making it quite difficult to solve. Some different nonlinear optimization methods that are being used to solve it were also mentioned. A thorough breakdown of the different methods, with associated pros and cons, will not be covered, as it's not the aim of this theses. The interested reader may see [4], [18] or [30] for an overview of different methods. Here, the focus will be on the Sequential Linear Programming (SLP) approach.

2.4.1 Linear Programming

LP is the most widely used of all optimization tools. An LP problem is characterized by having a linear objective function and linear constraints. Even nonlinear situations are often formulated as linear models, so that they can be solved by LP. Solution by LP is very appealing because of the advanced state and wide availability of the software, guaranteed convergence to the global optimum, and because of uncertainty in nonlinear models. LP's are usually expressed and analyzed in the standard form:

$$\begin{aligned} min \;\; & \boldsymbol{c^T x}, \\ s.t. \;\; & \boldsymbol{Ax} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq 0, \end{aligned} \tag{2.54}$$

where $\boldsymbol{c}$ and $\boldsymbol{x}$ are vectors in $\mathbb{R}^n$, $\boldsymbol{b}$ is a vector in $\mathbb{R}^m$ and $\boldsymbol{A}$ is an $m \times n$ matrix. Inequality constraints can also be transformed to the standard form by introducing a vector of slack variables, and potential negative variables can be made non-negative by splitting them into their non-negative and non-positive part. [31]

The Simplex Method

The simplex method was developed by George Dantzig in the late 1940s, and has since then been continually improved and refined, and remains one of the most popular algorithms for LP. There are several different variants, but perhaps the most common is the revised simplex method. [31] For a detailed description of the method, see chapter 13.3 of [31]. The simplex method is well established and is found in both commercial and freely available LP solvers. Although there are several competing algorithms, it holds sway as the most used algorithm in optimization software.

2.4.2 Sequential Linear Programming

Sequential LP, also known as Successive LP or Iterative LP, are algorithms that solve nonlinear optimization problems by a series of LP's. They are especially attractive for large, sparse nonlinear programs (NLP's). [32] The method consists of linearizing the nonlinear objective function and/or constraints around the current operating point, solve the optimization problem by LP, update the solution, and repeat iteratively until the optimal solution of the nonlinear problem is reached. Because the linearization is only valid in a smaller area around the linearization point, the change in the variables from one iteration to the next has to be restricted.

Consider the NLP:

$$\begin{aligned} min \;\; & f(\boldsymbol{x}), \\ s.t \;\; & \boldsymbol{g}(\boldsymbol{x}) = 0, \\ & \boldsymbol{h}(\boldsymbol{x}) \leq 0, \\ & \boldsymbol{x}^{min} \leq \boldsymbol{x} \leq \boldsymbol{x}^{max}, \end{aligned} \tag{2.55}$$

where $\boldsymbol{x}$ are a vector of optimization variables, and $\boldsymbol{x^{min}}$ and $\boldsymbol{x^{max}}$ are vectors of the variables lower and upper bounds, respectively. At every iteration k, the objective function f and constraints $\boldsymbol{g}$ and $\boldsymbol{h}$ are approximated by their first order

Taylor series:

$$m_k(\boldsymbol{p}) = f(\boldsymbol{x_k}) + \nabla f(\boldsymbol{x_k})^T \boldsymbol{p}, \tag{2.56}$$

$$\boldsymbol{g}(\boldsymbol{x_k}) + \nabla \boldsymbol{g}(\boldsymbol{x_k})^T \boldsymbol{p} = 0, \tag{2.57}$$

$$\boldsymbol{h}(\boldsymbol{x_k}) + \nabla \boldsymbol{h}(\boldsymbol{x_k})^T \boldsymbol{p} \leq 0, \tag{2.58}$$

where $\boldsymbol{p} = \boldsymbol{x} - \boldsymbol{x_k}$. In order to obtain a globally convergent method, a Trust Region (TR) method should be applied. This includes limiting the step-length by a value δ_k, called the TR radius. The optimization problem to solve at each iteration is then:

$$\begin{aligned} min \;\; & m_k(\boldsymbol{p}), \\ s.t. \;\; & \boldsymbol{g}(\boldsymbol{x_k}) + \nabla \boldsymbol{g}(\boldsymbol{x_k})^T \boldsymbol{p} = 0, \\ & \boldsymbol{h}(\boldsymbol{x_k}) + \nabla \boldsymbol{h}(\boldsymbol{x_k})^T \boldsymbol{p} \leq 0, \\ & max((\boldsymbol{x^{min}} - \boldsymbol{x_k}), -\delta_k) \;\leq\; \boldsymbol{p} \;\leq\; min((\boldsymbol{x^{max}} - \boldsymbol{x_k}), \delta_k). \end{aligned} \tag{2.59}$$

After the the solution $\boldsymbol{p_k}$ of (2.59) is computed, it's used as a step to define the new solution approximation: $\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \boldsymbol{p_k}$. A challenge, however, is that it may be impossible to find a solution for which the linearized constraints hold, or the found solution $\boldsymbol{p_k}$ could be so that the new approximation $\boldsymbol{x_{k+1}}$ doesn't satisfy the non-linear constraints of the original problem. To deal with the problem of infeasibility, some, or all, of the constraints may be added to the objective function as penalty parameters. Violation of these constraints would then be associated with a high positive cost, instead of rendering the problem unsolvable. [33]

Because the linearized problem could be a good or bad representation of the original nonlinear problem within the current TR radius, it's necessary to evaluate if the step $\boldsymbol{p_k}$ should be accepted, and if the TR radius should be adjusted for the next iteration. It's common to base these decisions on how well the *predicted reduction* of the objective function matches the *actual reduction*. [33] This can be described by the ratio between the actual- and predicted cost reduction:

$$\sigma_k = \frac{f(\boldsymbol{x_k}) - f(\boldsymbol{x_k} + \boldsymbol{p_k})}{m_k(0) - m_k(\boldsymbol{p_k})} \tag{2.60}$$

[34]If σ_k is above a certain threshold σ_{bad}, typically $\in (0, \frac{1}{4})$, the step is accepted. There is typically also defined different threshold for σ_k, for whether to decrease, increase or keep the TR radius. If $\sigma_k < \sigma_{bad}$, the step is rejected and the TR radius reduced. [33]

Chapter 3

Problem Formulation and Solution Method

This chapter endeavour to explain the problem and chosen solution strategy. First, relevant assumptions are stated before the problem is defined in mathematical terms. Then it's explained in two parts how the SCOPF problem is solved. First, the solution to the base-case ACOPF problem using SLP and a TR method is explained. Next, it's explained how the algorithm is extended to solve both the PSCOPF and CSCOPF problem using BD. Finally, some notes on the implementation are included.

3.1 Problem Formulation

3.1.1 Assumptions

To keep the problem manageable under the scope of the thesis work, some assumptions or simplifications have been made.

- The power system is in steady-state and under balanced conditions.
- Only one time-step is considered.
- Power demand at load buses are assumed to be known and are fixed.
- Only active power production has associated cost. All other quantities are allowed to take any value within their specified limits.

- Active and reactive power output of generators are the only considered control actions. Other control actions, such as transformer tap position and reactive power compensation, are hence not included.

- Line constraints are imposed using active power limits.

- No coupling constraints imposed on reactive power rescheduling. Reactive power is assumed to be freely re-dispatched during a contingency.

- No discrete variables, such as the start-up of generators and network switching, are considered.

3.1.2 Notation

Vectors, matrices and sets are written with bold type letters.

Indices

i, j	Bus indices
k	Iteration index
c	Contingency index

Sets

$\boldsymbol{N}$	Buses
$\boldsymbol{L}$	Branches (Lines)
$\boldsymbol{G}$	Generators
$\boldsymbol{M}$	Loads
$\boldsymbol{C}$	Contingency cases (line outages)

Parameters

$P_i^{G,min}, P_i^{G,max}$	Minimum and maximum active power generation at bus i
$Q_i^{G,min}, Q_i^{G,max}$	Minimum and maximum reactive power generation at bus i
V_i^{min}, V_i^{max}	Minimum and maximum allowed voltage magnitude at bus i
$\theta_i^{min}, \theta_i^{max}$	Minimum and maximum allowed voltage angle at bus i
$P_{i,j}^{max}$	Maximum active power flow on the branch from bus i to j
P_i^L, Q_i^L	Active and reactive power load at bus i

Variables

P_i^G, Q_i^G	Active and reactive power production at bus i
V_i, θ_i	Voltage magnitude and angle at bus i
ΔP_i, ΔQ_i	Change in active and reactive power at bus i
$\Delta\theta_i$, ΔV_i	Change in voltage angle and magnitude at bus i
z	Penalty variable

TR parameters

δ_k	TR radius at iteration k
δ_{max}	Maximum TR radius
σ_k	Ratio of actual- over predicted reduction in cost at iteration k
τ	Threshold for σ_k for accepting solution step
η	Chosen parameter for defining if approximation is good or bad.
γ	Factor by which to multiply δ_k by if approximation is bad.
ϵ	Convergence criterion

LP inputs

$\boldsymbol{v}$	Vector of linear cost coefficients
$\boldsymbol{A}$	Matrix of constraint coefficients
$\boldsymbol{b}$	Vector of r.h.s of constraints
$\boldsymbol{x}$	Vector of LP variables
$\boldsymbol{e}$	Vector with info on the ”type” of constraint
$\boldsymbol{vlb}$	Vector of LP variables lower bounds
$\boldsymbol{vub}$	Vector of LP variables upper bounds

Subproblem variables and parameters

$\Delta P_i \uparrow, \Delta P_i \downarrow$	Up- and down-regulation of base-case active power at bus i
$\Delta Q_i \uparrow, \Delta Q_i \downarrow$	Up- and down-regulation of base-case reactive power at bus i
s_i	Active power post-contingency rescheduling at bus i
μ	Fraction of generator maximum active power that can be post-contingency rescheduled.
Δ_i	Allowed active power post-contingency rescheduling at bus i.

3.1.3 Problem Statement

The objective is to minimize the total active power generation cost, while meeting all the system constraint and contingency constraints. The overall problem is stated

as follows:

$$
\begin{aligned}
min \quad & \sum_{i=1}^{N_{gen}} F_i(P_i^G), && && \text{(3.1a)}\\
s.t. \quad & P_i(V,\theta) = P_i^G - P_i^L && \forall\, i \in \boldsymbol{N}, && \text{(3.1b)}\\
& Q_i(V,\theta) = Q_i^G - Q_i^L && \forall\, i \in \boldsymbol{N}, && \text{(3.1c)}\\
& P_i^{G,min} \leq P_i^G \leq P_i^{G,max} && \forall\, i \in \boldsymbol{G}, && \text{(3.1d)}\\
& Q_i^{G,min} \leq Q_i^G \leq Q_i^{G,max} && \forall\, i \in \boldsymbol{G}, && \text{(3.1e)}\\
& V_i^{min} \leq V_i \leq V_i^{max} && \forall\, i \in \boldsymbol{N}, && \text{(3.1f)}\\
& \theta_i^{min} \leq \theta_i \leq \theta_i^{max} && \forall\, i \in \boldsymbol{N}, && \text{(3.1g)}\\
& |P_{ij}| \leq P_{ij}^{max} && \forall\, (i,j) \in \boldsymbol{L}, && \text{(3.1h)}\\
& w_c(\boldsymbol{P^G}) \leq 0 && \forall\, c \in \boldsymbol{C}, && \text{(3.1i)}
\end{aligned}
$$

where F_i is the cost function of the generator at bus i, which in this case is assumed to be a quadratic polynomial function:

$$F_i(P_i^G) = a_i + b_i P_i^G + c_i (P_i^G)^2. \tag{3.2}$$

Constraints (3.1b) through (3.1h) are the base-case operating constraints, while constraints (3.1i) represent the contingency constraints as described in subsection 2.3.2. Constraints (3.1i) require that constraints (3.1b) through (3.1h) also holds under all defined contingencies in $\boldsymbol{C}$, after any potentially allowed post-contingency rescheduling have been performed.

3.2 Solution Strategy

Problem (3.1) is proposed solved by an SLP algorithm using a TR method, and BD for handling the contingency constraints.

3.2.1 ACOPF Algorithm

First, the optimization problem (3.1a) subject to constraints (3.1b) through (3.1h) are considered. That is a standard ACOPF problem and is proposed solved using SLP, similar to that presented in chapter 8.10 of [9]. The algorithm follows the methodology outlined in subsection 2.4.2, but is adapted to fit the OPF problem. A TR method for adjusting the TR radius, or "window size", following that outlined in [34] is employed. The flowchart in Figure 3.1 gives an overview of the overall iterative ACOPF algorithm. The different steps will be discussed in more detail next.

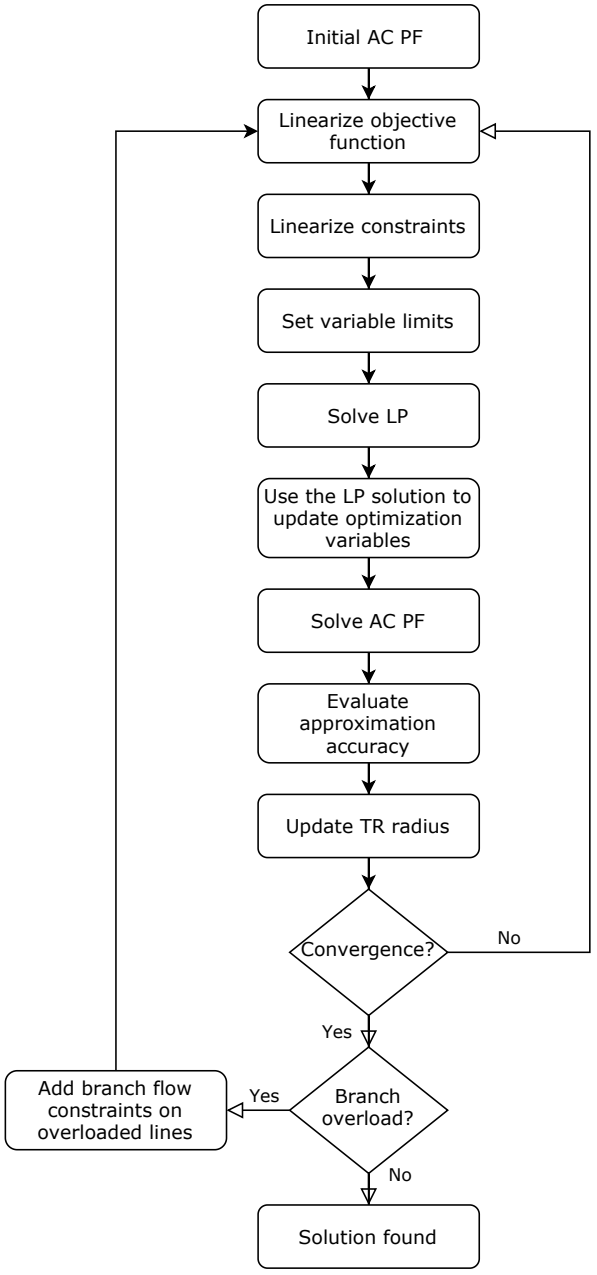


Figure 3.1: Flow chart of the ACOPF solution algorithm

From now on, the optimization will be written in terms of the changes is the original OPF variables:

$$\boldsymbol{\Delta\theta}, \boldsymbol{\Delta V}, \boldsymbol{\Delta P} \text{ and } \boldsymbol{\Delta Q},$$

where $\boldsymbol{\Delta\theta} = [\Delta\theta_1, ..., \Delta\theta_N]$, $\boldsymbol{\Delta V} = [\Delta V_1, ..., \Delta V_N]$, $\boldsymbol{\Delta P} = [\Delta P_1, ..., \Delta P_N]$ and $\boldsymbol{\Delta Q} = [\Delta Q_1, ..., \Delta Q_N]$. Because the only changes in active and reactive power we are allowing are the generator's output, $\boldsymbol{\Delta P} = \boldsymbol{\Delta P^G}$ and $\boldsymbol{\Delta Q} = \boldsymbol{\Delta Q^G}$.

Step 1 - Initial PF

Before starting to set up the LP a valid PF solution is needed as a starting point. Therefore a NR PF is run, giving the set of initial solution values:

$$\boldsymbol{P_0}, \boldsymbol{Q_0}, \boldsymbol{V_0} \text{ and } \boldsymbol{\theta_0},$$

Step 2 - Linearize the objective function

The objective function is linearized around the current operating point. As the problem is formulated by means of the changes in the OPF variables, the constant term of (2.56) is uninteresting, as it's considered constant in the LP calculation. The linearized objective function is then:

$$m_k(\boldsymbol{\Delta P^G}) = \sum_{i=1}^{N_{gen}} \frac{dF_i(P_i^G)}{dP_i^G} \Delta P_i^G, \tag{3.3}$$

Assuming the quadratic cost function of (3.2), the linearized objective function may be written as:

$$m_k(\boldsymbol{\Delta P^G}) = \boldsymbol{v^T \Delta P^G}, \tag{3.4}$$

where $\boldsymbol{v} = [v_1, ..., v_{N_{gen}}]$, and:

$$v_i = b_i + 2c_i P_i^G. \tag{3.5}$$

Although the cost function only depends on the active power production of the generators, all the variables in the optimization must be included. Therefore, the cost vector $\boldsymbol{v}$ will in reality be of length 4N, with zeros corresponding to $\Delta\theta_i$, ΔV_i, and ΔQ_i, $\forall\ i \in \boldsymbol{N}$, and ΔP_i, $\forall\ i \in \boldsymbol{M}$.

Step 3 - Linearize the constraints

The next step is to linearize the constraints, starting with the PF equality constraints. This is essentially the same as the linearization done in the NR method, as described in subsection 2.1.5. The only difference is that, in this case, all rows and columns of the Jacobian are included:

$$\begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \cdots & \frac{\partial P_1}{\partial \theta_N} & \frac{\partial P_1}{\partial V_1} & \cdots & \frac{\partial P_1}{\partial V_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_N}{\partial \theta_1} & \cdots & \frac{\partial P_N}{\partial \theta_N} & \frac{\partial P_N}{\partial V_1} & \cdots & \frac{\partial P_N}{\partial V_N} \\ \frac{\partial Q_1}{\partial \theta_1} & \cdots & \frac{\partial Q_1}{\partial \theta_N} & \frac{\partial Q_1}{\partial V_1} & \cdots & \frac{\partial Q_1}{\partial V_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_N}{\partial \theta_1} & \cdots & \frac{\partial Q_N}{\partial \theta_N} & \frac{\partial Q_N}{\partial V_1} & \cdots & \frac{\partial Q_N}{\partial V_N} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \vdots \\ \Delta\theta_N \\ \Delta V_1 \\ \vdots \\ \Delta V_N \end{bmatrix} = \begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_N \\ \Delta Q_1 \\ \vdots \\ \Delta Q_N \end{bmatrix} \tag{3.6}$$

Taking all the variables to the left hand side of the equation, the equality constraints can be expressed as:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{3.7}$$

with:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \cdots & \frac{\partial P_1}{\partial \theta_N} & \frac{\partial P_1}{\partial V_1} & \cdots & \frac{\partial P_1}{\partial V_N} & -1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & 0 & \ddots & & \vdots \\ \frac{\partial P_N}{\partial \theta_1} & \cdots & \frac{\partial P_N}{\partial \theta_N} & \frac{\partial P_N}{\partial V_1} & \cdots & \frac{\partial P_N}{\partial V_N} & \vdots & & & \vdots \\ \frac{\partial Q_1}{\partial \theta_1} & \cdots & \frac{\partial Q_1}{\partial \theta_N} & \frac{\partial Q_1}{\partial V_1} & \cdots & \frac{\partial Q_1}{\partial V_N} & \vdots & & & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & & \ddots & 0 \\ \frac{\partial Q_N}{\partial \theta_1} & \cdots & \frac{\partial Q_N}{\partial \theta_N} & \frac{\partial Q_N}{\partial V_1} & \cdots & \frac{\partial Q_N}{\partial V_N} & 0 & \cdots & 0 & -1, \end{bmatrix}$$

$$\boldsymbol{x} = [\Delta\theta_1 \dots \Delta\theta_N \;\; \Delta V_1 \dots \Delta V_N \;\; \Delta P_1 \dots \Delta P_N \;\; \Delta Q_1 \dots \Delta Q_N]^T,$$

$$\boldsymbol{b} = [0 \dots 0]^T.$$

$\boldsymbol{A}$ is a $2N \times 4N$ coefficient matrix, which is a concatenation of the $2N \times 2N$ Jacobian matrix and a $2N \times 2N$ negative identity matrix, $\boldsymbol{x}$ is the variable vector of length $4N$, and $\boldsymbol{b}$ is a vector of zeros of length $2N$.

Also, any potential branch constraints must be linearized. Although constraints of all branches could be added from the start, it's chosen to wait until an optimal solution without branch constraints is found, and then add constraints only for

those branches that are overloaded. This is to reduce the number of constraints fed to the LP solver, as most branches are usually not overloaded. This part is, therefore, skipped if no branch constraints have yet been added to the problem.

As this formulation uses active power constraints on branch flow, the linearized constraints are formulated by means of the marginal change in active power flow on line (i, j), P_{ij}, to marginal changes in the state variables: θ_i, θ_j, V_i and V_j, evaluated for the current state variables $\boldsymbol{\theta_k}$ and $\boldsymbol{V_k}$:

$$\frac{\partial P_{ij}}{\partial \theta_i}\Delta\theta_i + \frac{\partial P_{ij}}{\partial \theta_j}\Delta\theta_j + \frac{\partial P_{ij}}{\partial V_i}\Delta V_i + \frac{\partial P_{ij}}{\partial V_j}\Delta V_j \leq P_{ij}^{max} - P_{ij} \tag{3.8}$$

If the line flow P_{ij} should be negative, the constraint becomes:

$$\frac{\partial P_{ij}}{\partial \theta_i}\Delta\theta_i + \frac{\partial P_{ij}}{\partial \theta_j}\Delta\theta_j + \frac{\partial P_{ij}}{\partial V_i}\Delta V_i + \frac{\partial P_{ij}}{\partial V_j}\Delta V_j \geq -P_{ij}^{max} - P_{ij} \tag{3.9}$$

The partial derivatives in Equations (3.8) and (3.9) can be found by partial differentiation of Equation (2.10). Using the composite terms defined in (2.22) and (2.23), they can be written as:

$$\frac{\partial P_{ij}}{\partial \theta_i} = V_i V_j U_{ij} \tag{3.10}$$

$$\frac{\partial P_{ij}}{\partial \theta_j} = -V_i V_j U ij \tag{3.11}$$

$$\frac{\partial P_{ij}}{\partial V_i} = 2V_i G_{ij} - V_j T_{ij} \tag{3.12}$$

$$\frac{\partial P_{ij}}{\partial V_j} = -V_j T_{ij} \tag{3.13}$$

The branch flow constraints can be included with the other constraints by adding new rows of length 4N to $\boldsymbol{A}$ on the form:

$$[\frac{\partial P_{ij}}{\partial \theta_i}\ \ 0\ldots0\ \ \frac{\partial P_{ij}}{\partial \theta_j}\ \ 0\ldots0\ \ \frac{\partial P_{ij}}{\partial V_i}\ \ 0\ldots0\ \ \frac{\partial P_{ij}}{\partial V_j}\ \ 0\ldots\ldots0\ \ ].$$

The positions of the derivative terms above, correspond to the position of the corresponding variable in the variable vector. For instance, the term $\frac{\partial P_{13}}{\partial V_3}$ has the same position in the row as ΔV_3 has in $\boldsymbol{x}$. The value of of the r.h.s of (3.8) or (3.9) are appended to $\boldsymbol{b}$. To keep track of which constraints should be "equality", "lesser than" or "greater than", an additional vector $\boldsymbol{e}$ is used. It has the same number of elements as there are constraints, and the value at the position corresponding to a row in $\boldsymbol{A}$ tells what kind of constraint it is:

0	$\Leftrightarrow$	Equality constraint
1	$\Leftrightarrow$	Inequality "greater than" constraint
-1	$\Leftrightarrow$	Inequality "lesser than" constraint

Step 4 - Set variables bounds

The optimization variables need to be restricted. No changes should be larger than the current TR radius δ_k The voltage magnitude and angle at the reference bus are not allowed to be changed, that is:

$$\Delta V_{ref} = 0, \quad \Delta\theta_{ref} = 0 \tag{3.14a}$$

For the remaining buses, the voltage bounds are set as:

$$\theta_i^{min} - \theta_i \leq \Delta\theta_i \leq \theta_i^{max} - \theta_i, \qquad \forall\, i \neq ref \in \boldsymbol{N}, \tag{3.14b}$$

$$max(V_i^{min} - V_i,\ -\delta) \leq \Delta V_i \leq min(V_i^{max} - V_i,\ \delta), \quad \forall\, i \neq ref \in \boldsymbol{N}. \tag{3.14c}$$

The change in voltage angle are not limited by the TR radius because it has not shown to have any effect. The power at the load buses are not allowed to change, hence:

$$\Delta P_i = 0, \quad \Delta Q_i = 0, \quad \forall\, i \in \boldsymbol{M}. \tag{3.14d}$$

For the remaining buses, the power bounds are chosen as:

$$max(P_i^{G,min} - P_i,\ -\delta) \leq \Delta P_i \leq min(P_i^{G,max} - P_i,\ \delta), \quad \forall\, i \in \boldsymbol{G}, \tag{3.14e}$$

$$max(Q_i^{G,min} - Q_i,\ -\delta) \leq \Delta Q_i \leq min(Q_i^{G,max} - Q_i,\ \delta), \quad \forall\, i \in \boldsymbol{G}. \tag{3.14f}$$

The lower and upper bounds on the optimization variables, as chosen above, are stored in two different vectors $\boldsymbol{vlb}$ and $\boldsymbol{vub}$, both of size $4N$.

Step 5 - Solve LP

The linearized objective function, linearized constraints and variable bounds are fed to the LP solver. The result of the LP calculation include the suggested changes $\boldsymbol{\Delta V}$, $\boldsymbol{\Delta\theta}$, $\boldsymbol{\Delta P}$ and $\boldsymbol{\Delta Q}$, as well as the value of linearized objective function $m_k(\boldsymbol{\Delta P^G})$.

Step 6 - Update variables

Next, the LP result is used to update the state- and control variables of the original optimization problem. That is:

$$\boldsymbol{\theta_{k+1}} = \boldsymbol{\theta_k} + \boldsymbol{\Delta\theta}, \tag{3.15}$$

$$\boldsymbol{V_{k+1}} = \boldsymbol{V_k} + \boldsymbol{\Delta V}, \tag{3.16}$$

$$\boldsymbol{P^G_{k+1}} = \boldsymbol{P^G_k} + \boldsymbol{\Delta P}, \tag{3.17}$$

$$\boldsymbol{Q^G_{k+1}} = \boldsymbol{Q^G_k} + \boldsymbol{\Delta Q}. \tag{3.18}$$

Step 7 - Solve AC PF

Using the updated variables from Equations (3.15)-(3.18), a full NR PF is run. This will find an operating state that satisfies the nonlinear PF equations, and the dependent variables' actual response to the changes in control variables are obtained. The actual system operation cost $f(\boldsymbol{P^G})$ after the suggested step is also calculated.

Step 8 - Evaluate Approximation Accuracy and Update TR radius

How good the linearized problem represent the actual nonlinear problem is evaluated by recalculating the factor σ_k according to Equation (2.60). In this case:

$$\sigma_k = \frac{f(\boldsymbol{P_k^G}) - f(\boldsymbol{P_{k+1}^G})}{m_k(\boldsymbol{\Delta P^G})} \tag{3.19}$$

Following the logic used in [34], the value of $\sigma_k \in [0, 1]$ decide whether the solution step is accepted or not, and what will be the TR radius in the next iteration. The decision process looks something like:

If	$\sigma_k < \tau$:	(very bad approximation)
	solution step is rejected	
	$\delta_{k+1} = \gamma\delta_k$	
Elseif	$\sigma_k < \eta$:	(bad approximation)
	solution step is accepted	
	$\delta_{k+1} = \gamma\delta_k$	
Elseif	$\sigma_k < (1 - \eta)$:	(ok approximation)
	solution step is accepted	
	$\delta_{k+1} = \delta_k$	
Else:		(good approximation)
	solution step is accepted	
	$\delta_{k+1} = max(2\delta_k,\ \delta_{max})$	

The parameters τ, η and γ may be adjusted to achieve better convergence. Here the values $\tau = 0$, $\eta = 0.2$ and $\gamma = 0.1$ are used.

Step 10 - Check for convergence

There are several ways to test for convergence. In [9], it's suggested to look for significant changes in the variables $\boldsymbol{\Delta V}$, $\boldsymbol{\Delta P^G}$ and $\boldsymbol{\Delta Q^G}$. If there are no significant changes, the algorithm is assumed to have converged. If not, move back to *step 2* for a new iteration. It was however chosen to look at the changes in actual cost ($f(\boldsymbol{P^G})$) over the past iterations. More specifically, if the standard deviation

of the actual cost for the last 3 iterations is under the convergence criterion ϵ, then convergence is said to be reached. If the convergence criterion is met, proceed to *step 11*, otherwise return to *step 2*.

Step 11 - Check for branch overloads

Calculate the branch flows, and check if any of the branches exceed their specified limits. As active power flow limits are used here, the branch flows are calculated using Equation (2.10). If no branches are overloaded, the solution is found. Otherwise, flag the overloaded lines so that constraints on these lines are added in *Step 3* in the next iteration. Then, move to *step 2* for a new iteration.

3.2.2 SCOPF Algorithm

Now, the complete problem (3.1), including the contingency constraints (3.1i), are considered. The problem is solved using a BD approach, using the principles described in subsection 2.3.2. Figure 3.2 gives a rough idea of the program flow. The method presented here considers the possibility of allowing for some post-contingency rescheduling. It's assumed that each generator can adjust it's active power output by a certain fraction of its maximum output within "reasonable time" after a contingency occurs. This fraction is described by the factor μ. Setting $\mu = 0$, corresponds to demanding preventive security, i.e. not allowing for any post-contingency rescheduling.

Step 1 - Solve base-case ACOPF problem

The first step in the SCOPF algorithm is to solve the base-case problem as described in the previous section. That gives an optimal solution to the problem before any contingencies have been considered.

Step 2 - Solve N_c contingency subproblems

After a base-case solution is found, a subproblem for each contingency in $\boldsymbol{C}$ is solved. They may be solved in parallel, but are here solved iteratively. The objective of the subproblem is to find out if the system stays feasible during the contingency, and if not, which adjustments must be made to the base-case to make it so. Here, the considered contingencies are the removal of individual lines. When a contingency subproblem is started, the relevant line is removed from the set of branches $\boldsymbol{L}$, and the matrices $\boldsymbol{G_{bus}}$ and $\boldsymbol{B_{bus}}$, which describe the system topology, are recalculated. Then, a PF calculation is conducted, before moving on to the optimization.

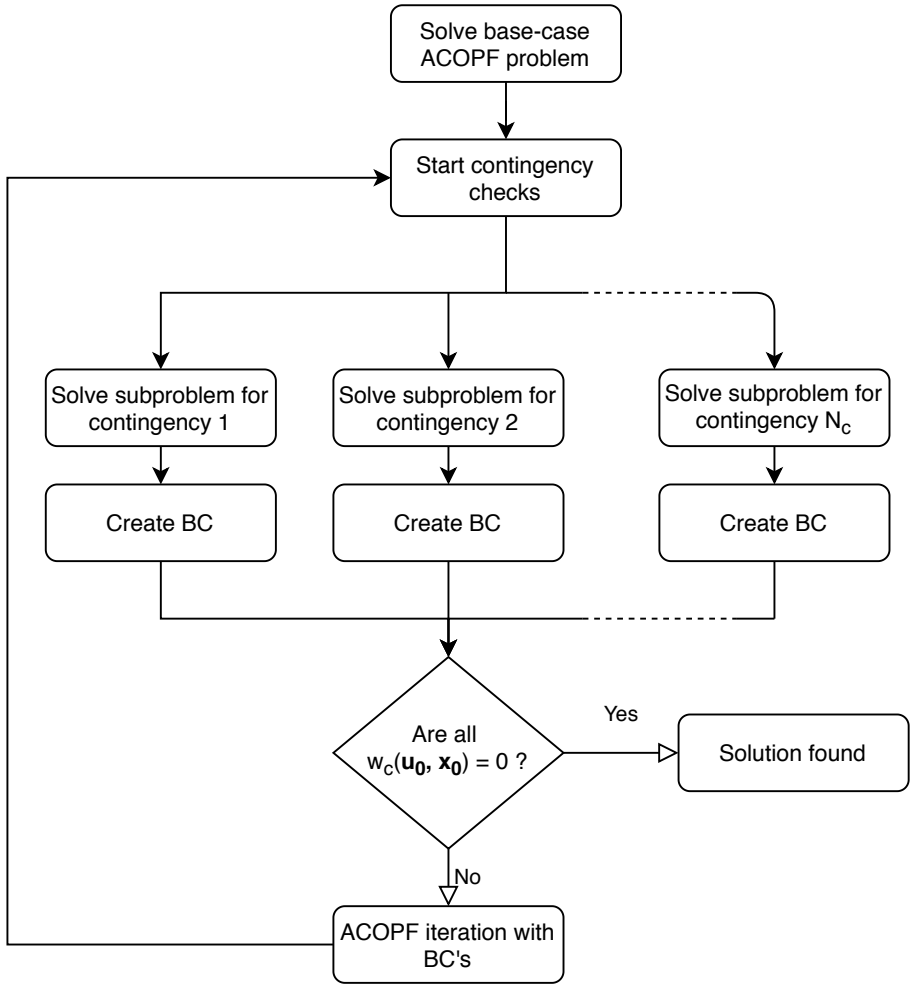


Figure 3.2: Flow chart of the SCOPF algorithm.

Step 2a - Subproblem LP calculation

The subproblem LP formulation is very similar to a single ACOPF iteration. The main difference is that instead of using a linearized objective function, all changes in active power production from the base-case solution has a positive penalty cost, which are to be minimized. In order to make it possible to penalise both up- and down- regulation, the change in active power must be split in two. The same is done for the reactive power, for easier implementation. The optimization subproblem is therefore written in terms of the variables:

$$\boldsymbol{\Delta\theta}, \boldsymbol{\Delta V}, \boldsymbol{\Delta P}\uparrow, \boldsymbol{\Delta Q}\uparrow, \boldsymbol{\Delta P}\downarrow \text{ and } \boldsymbol{\Delta Q}\downarrow,$$

where

$$\Delta P_i = \Delta P_i \uparrow - \Delta P_i \downarrow, \ \forall \, i \in \boldsymbol{N}, \tag{3.20}$$
$$\Delta Q_i = \Delta Q_i \uparrow - \Delta Q_i \downarrow, \ \forall \, i \in \boldsymbol{N}. \tag{3.21}$$

In addition, the vector of active power corrective actions $\boldsymbol{s}$ is introduced. When corrective actions are allowed Equation (3.20) changes to:

$$\Delta P_i = \Delta P_i \uparrow - \Delta P_i \downarrow + s_i, \ \forall \, i \in \boldsymbol{N}, \tag{3.22}$$

where:

$$max(P_i^{G,min} - P_i, \ \Delta_i) \leq s_i \leq min(P_i^{G,max} - P_i, \ \Delta_i), \ \forall \, i \in \boldsymbol{N},$$

where Δ_i is the allowed rescheduling at bus i, and is here calculated as:

$$\Delta_i = \mu P_i^{G,max}. \tag{3.23}$$

The subproblem objective function is:

$$w_c(\boldsymbol{\Delta P} \uparrow, \boldsymbol{\Delta P} \downarrow) = min \ \boldsymbol{v^T}[\boldsymbol{\Delta P} \uparrow \boldsymbol{\Delta P} \downarrow], \tag{3.24}$$

where $\boldsymbol{v}$ = [1,...,1] is a cost vector of length 2N. As for the optimization of the base-case, the cost vector must have elements corresponding to all the variables in the optimization. Therefore, in reality, $\boldsymbol{v}$ is of length 7N, with ones corresponding to $\boldsymbol{\Delta P} \uparrow$ and $\boldsymbol{\Delta P} \downarrow$, and zeroes corresponding to $\boldsymbol{\Delta \theta}$, $\boldsymbol{\Delta V}$, $\boldsymbol{\Delta Q} \uparrow$, $\boldsymbol{\Delta Q} \downarrow$ and $\boldsymbol{s}$.

With the variables of the subproblem, the linearized PF equations are:

$$\begin{bmatrix} \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} \\ \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta \theta} \\ \boldsymbol{\Delta V} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta P} \uparrow - \boldsymbol{\Delta P} \downarrow + \boldsymbol{s} \\ \boldsymbol{\Delta Q} \uparrow - \boldsymbol{\Delta Q} \downarrow \end{bmatrix} \tag{3.25}$$

In the standard form $\boldsymbol{Ax} = \boldsymbol{b}$, this then becomes:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{1} \\ \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \end{bmatrix},$$
$$\boldsymbol{x^T} = [\boldsymbol{\Delta \theta} \ \boldsymbol{\Delta V} \ \ \boldsymbol{\Delta P} \uparrow \boldsymbol{\Delta Q} \uparrow \boldsymbol{\Delta P} \downarrow \boldsymbol{\Delta Q} \downarrow \ \boldsymbol{s}],$$
$$\boldsymbol{b^T} = [0 \ ... \ 0],$$

where $\boldsymbol{A}$ is the a coefficient matrix of size $2N \times 7N$, $\boldsymbol{x}$ is the variable vector of length $7N$ and $\boldsymbol{b}$ is a vector of zeros of length $2N$.

Branch constraints are added for those lines which exceed their limits in the post-contingency PF calculation. This is done the same way as in *step 3* of the ACOPF algorithm. The only difference is that extra zeros must be added to the row which is appended to $\boldsymbol{A}$ so that the length of the row is $7N$.

The variables are bounded like in *step 4* of the ACOPF algorithm, but without being limited by the TR radius. That is:

$$\Delta V_{ref} = 0, \quad \Delta\theta_{ref} = 0 \tag{3.26a}$$

$$\theta_i^{min} - \theta_i \leq \Delta\theta_i \leq \theta_i^{max} - \theta_i, \qquad \forall\, i \neq ref \in \boldsymbol{N}, \tag{3.26b}$$

$$V_i^{min} - V_i \leq \Delta V_i \leq V_i^{max} - V_i, \qquad \forall\, i \neq ref \in \boldsymbol{N}. \tag{3.26c}$$

$$\Delta P_i \uparrow = 0, \;\; \Delta P_i \downarrow = 0, \;\; \Delta Q_i \uparrow = 0, \;\; \Delta Q_i \downarrow = 0, \;\; s_i = 0, \;\; \forall\, i \in \boldsymbol{M}. \tag{3.26d}$$

$$0 \leq \Delta P_i \uparrow \leq P_i^{G,max} - P_i, \qquad \forall\, i \in \boldsymbol{G}, \tag{3.26e}$$

$$0 \leq \Delta P_i \downarrow \leq -(P_i^{G,min} - P_i), \qquad \forall\, i \in \boldsymbol{G}, \tag{3.26f}$$

$$0 \leq \Delta Q_i \uparrow \leq Q_i^{G,max} - Q_i, \qquad \forall\, i \in \boldsymbol{G}, \tag{3.26g}$$

$$0 \leq \Delta Q_i \downarrow \leq -(Q_i^{G,min} - Q_i), \qquad \forall\, i \in \boldsymbol{G}. \tag{3.26h}$$

$$max(P_i^{G,min} - P_i,\ \Delta_i) \leq s_i \leq min(P_i^{G,max} - P_i,\ \Delta_i), \quad \forall\, i \in \boldsymbol{G}. \tag{3.26i}$$

With the now defined cost function, constraints and variable bounds, the LP is calculated. The resulting variable values, represent the changes that has to be made in order to make the post-contingency state feasible, according to the linearized model. The values of $\boldsymbol{s}$ will therefore be the post-contingency rescheduling of active power that is assumed to be made "within reasonable time" when we are considering CSCOPF. It should however be noted that the model is linearized around the base-case operating point. This rescheduling is therefore not accurate if the rescheduling is large.

Step 2b - Benders Cut

If a subproblem solution has a cost which is not zero, it means that adjustments in the base-case active power production must be made in order for the system to stay safe during the evaluated contingency case. The way to feed this information to the base-case problem, is to create a BC. It's a linear constraint that can be added to the base-case. Associated with the subproblem LP solution, there is a set of Lagrange multipliers, which is the marginal change in the objective function value to marginal changes in the optimization variables. For instance is $\frac{\partial w}{\partial P_2}$ how

much the objective function w would change with one unit change in P_2. The BC constraint can be formulated in terms of the base-case optimization variables as:

$$\begin{bmatrix} \frac{\partial w_c}{\partial P_1 \uparrow} & \cdots & \frac{\partial w_c}{\partial P_N \uparrow} \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_N \end{bmatrix} \geq w_c. \tag{3.27}$$

The vector of Lagrange multipliers and the value of w_c are stored, and the constraint can be added in the next ACOPF iteration the same way as the other constraints. That is by adding the vector of the Lagrange multipliers (with zeros in the positions corresponding to the other variables) to the matrix $\boldsymbol{A}$, appending w_c to $\boldsymbol{b}$ and appending a 1 to $\boldsymbol{e}$ (to mark as "greater than" constraint).

Step 3 - Finish check

If none of the subproblems led to non-zero w_c, none of the contingencies that were included led to a post-contingency system state that was infeasible after the allowed rescheduling actions. If also the convergence criterion of the ACOPF algorithm is met, the solution is found and the program can exit. If, however, there were subproblems were $w_c > 0$, or the convergence criteria is no longer met, the loop continues.

Step 4 - New iteration

A new ACOPF iteration is run, but this time with the BC's of the infeasible contingency subproblems included as constraints. Because everything is linearized, there may still be infeasible subproblems, so afterwards the algorithm returns to *step 2* for a new round of contingency checks.

3.2.3 Penalty Variable

A problem that can occur with the algorithms as described above, is that the LP problem may be infeasible when new constraints are added and the TR radius is too small for the constraints to be met. In response, two measures are made. First, when new constraints are added, the TR radius is increased. However, it's not desirable to increase it too much, as the linearizations may become too inaccurate. Therefore, also a non-negative penalty variable z is added to the master problem. This variable has a very high associated cost, and will therefore only be non-zero if the alternative is problem infeasibility. For instance, will a BC contingency constraint look like:

$$\begin{bmatrix} \frac{\partial w_c}{\partial P_1 \uparrow} & \cdots & \frac{\partial w_c}{\partial P_N \uparrow} \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_N \end{bmatrix} + z \geq w_c. \tag{3.28}$$

The program will never be allowed to finish unless $z = 0$. An iteration with $z \neq 0$ may, however, give a solution step that drives the solution in the right direction. The solution step may, therefore, be accepted, even though the temporary solution is infeasible.

3.3 Implementation

The algorithm has already been explained quite detailed, but some notes on the actual implementation are included here. The program is implemented in Python and takes an object-oriented approach. The system is described by a set of objects of defined classes. The classes included in the current implementation are ”Line” and ”Bus”, where the class ”Bus” has the two sub-classes ”GeneratorBus” and ”LoadBus”. All system data are included in the initialization of these objects. The only inputs to the ACOPF program are a list of buses and a list of lines. The SCOPF program also takes a list of lines to include in the contingency analysis.

A NR PF solver, following the principles described in subsection 2.1.5, are implemented for executing the required AC PF's. It's implemented to take the bus- and line-lists as inputs, and return a PF solution.

3.3.1 LP Solve

For solving the LP problems, the free LP solver *LP Solve* is used. It uses the revised simplex method and is fully integrable with Python. To run LP Solve from Python, the driver program *lpsolve55* is needed. For more information about LP Solve with Python, see [35].

For easy use of LP Solve in Python, a function called lp_maker is used. It's based on the code found in [36]. It's slightly modified to fit the current version of LP Solve. The function takes as input the cost vector $\boldsymbol{v}$, the constraint matrix and vectors $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{e}$, the upper and lower bounds vectors $\boldsymbol{vlb}$ and $\boldsymbol{vub}$, and a flag to specify minimization. As it's output is an LP problem (lp) on a format that can be sent to LP Solve.

The LP can then be solved by calling:

lpsolve('solve', lp).

The resulting variables and objective function value can be retrived by calling:

lpsolve('get_variables', lp)[0],
lpsolve('get_objective', lp),

respectively, and the reduced costs by calling:

lpsolve('get_reduced_costs', lp,)[0].

To find the Lagrange multipliers needed for the BC, the reduced costs are subtracted from the cost vector. The cost vector in the subproblem is a vector of ones corresponding to the active power at the generator buses.

Chapter 4

Numerical Examples

4.1 6-bus ACOPF Solution

The ACOPF algorithm described in subsection 3.2.1 is tested on the 6-bus example system depicted in Figure 4.1. The example is taken from [9]. In the calculation, all quantities are converted to the per unit (pu) system, with a base power of $S_{base} = 100MVA$. The system consists of three generator buses, where bus 1 is the slack bus. The generators at these buses should supply three load buses with a demand of 50 MW and 15 MVA each. The generators have different ratings and cost functions, as can be seen from Table A.1. The buses are interconnected by lines, which impedance data can be found in Table A.3. The objective is to find the active power generation at each generator that minimizes the total generation cost while meeting all system constraints. All buses have the voltage limits:

$$\begin{aligned} 0.95\ pu\ &\leq V_i\ \leq 1.07\ pu \\ -\pi\ rad\ &\leq \theta_i \leq \pi\ rad \end{aligned}$$

The initial PF solution is presented in Table 4.1. In this situation, the slack bus is producing the majority of the active power. The total operating cost, given the quadratic generator cost functions, is 4479 $/MWh. After 6 iterations, convergence is detected in *step 10*, at an operation cost of 4235 $/MWh, which can be seen in Table 4.2. The branch overload test in *step 11*, however, detects overloads on lines 2-4 and 3-6 (see Table 4.3). These lines are then flagged, so that branch flow constraints on these lines are added in *step 3* in the following iterations. At iteration 9, convergence is again reached, and this time no branch flows exceed their limits. The solution is therefore found, and the program exits.

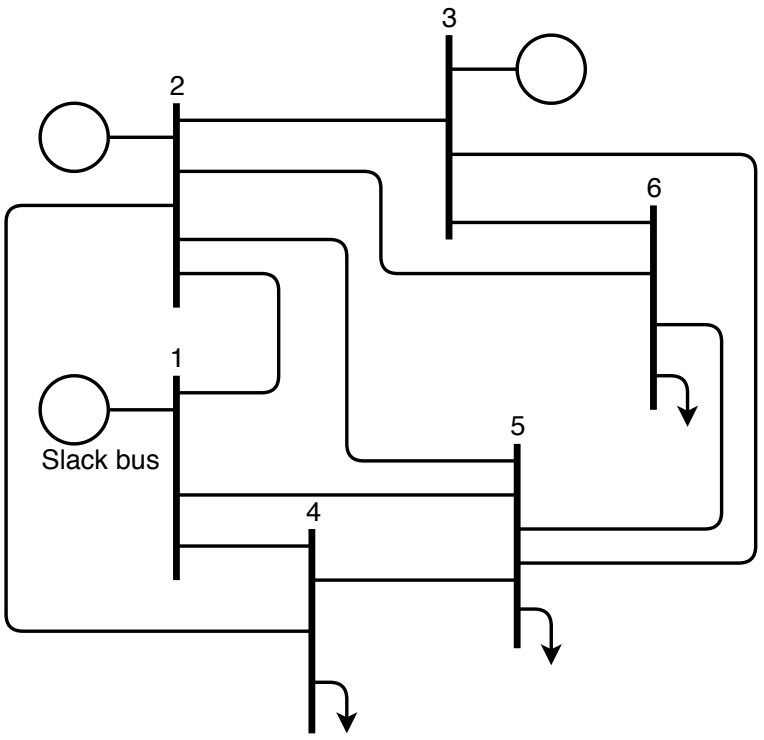


Figure 4.1: Six-bus power flow system. Adapted from [9].

The final solution is presented in Table 4.4 and Table 4.5. The resulting operating cost is 4255 $/MWh, which is a reduction of 224 $/MWh, or 5.45 %, from the initial operation, and 20 $/MWh higher than when line limits were disregarded. A plot showing the cost development through the program execution is presented in Figure 4.2. The cost increase after the inclusion of branch flow limits after iteration 6 is clearly visible.

Table 4.1: Initial PF solution for the 6-bus test system

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	$\theta(rad)$
1	2.12956	-0.10759	1.07	0
2	0.5	0.21757	1.05	-0.12525
3	0.5	0.19016	1.05	-0.15996
4	-1.0	-0.15	1.02721	-0.14748
5	-1.0	-0.15	1.02212	-0.18378
6	-1.0	-0.15	1.02458	-0.20529
Total generation	3.12956	0.300138		
Total loss	0.12956	-0.149862		
Total cost	4478.84 $/MWh			

Table 4.2: Bus info for 6-bus ACOPF result before branch constraints are added (iteration 6)

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.76485	0.02817	1.07	0
2	1.08566	0.03513	1.07	-0.01172
3	1.21975	0.02451	1.07	0.00526
4	-1.00	-0.15	1.03733	-0.06846
5	-1.00	-0.15	1.0369	-0.08036
6	-1.00	-0.15	1.04586	-0.06279
Total Generation	3.07026	0.08781		
Total Loss	0.07026	-0.36219		
Total Cost	4234.98 $/MWh			

Table 4.3: Branch flows for 6-bus ACOPF solution before line constraints are added (iteration 6).

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.05385	-0.02653	0.06003	1.0
1	4	0.40151	0.0874	0.41092	1.0
1	5	0.30953	0.04744	0.31314	1.0
2	3	-0.07463	0.01559	0.07624	0.6
2	**4**	**0.65053**	0.04214	0.65189	**0.6**
2	5	0.2663	0.03798	0.269	0.6
2	6	0.29698	0.0325	0.29875	0.6
3	5	0.35859	-0.01367	0.35885	0.6
3	**6**	**0.78626**	0.12694	0.79644	**0.6**
4	5	0.02611	-0.01175	0.02864	0.6
5	6	-0.06629	-0.0083	0.0668	0.6

Table 4.4: Bus info for 6-bus ACOPF final solution (iteration 9)

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	1.05875	-0.05138	1.07	0
2	1.29312	0.06219	1.07	-0.03311
3	0.72261	0.09587	1.06544	-0.05798
4	-1.00	-0.15	1.03808	-0.08462
5	-1.00	-0.15	1.03579	-0.11134
6	-1.00	-0.15	1.04184	-0.10942
Total Generation	3.07448	0.10668		
Total Loss	0.07448	-0.34332		
Total Cost	4254.64 $/MWh			

Table 4.5: Branch flows for 6-bus ACOPF final solution (iteration 9).

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.15288	-0.0733	0.16954	1
1	4	0.48664	0.06899	0.4915	1
1	5	0.4193	0.03306	0.4206	1
2	3	0.11305	-0.0017	0.11306	0.6
2	**4**	**0.59999**	0.05628	0.60262	**0.6**
2	5	0.29982	0.03336	0.30167	0.6
2	6	0.43058	0.01616	0.43089	0.6
3	5	0.23516	0.01901	0.23592	0.6
3	**6**	**0.60000**	0.14615	0.61754	**0.6**
4	5	0.06022	-0.02322	0.06454	0.6
5	6	-0.01247	-0.01672	0.02086	0.6

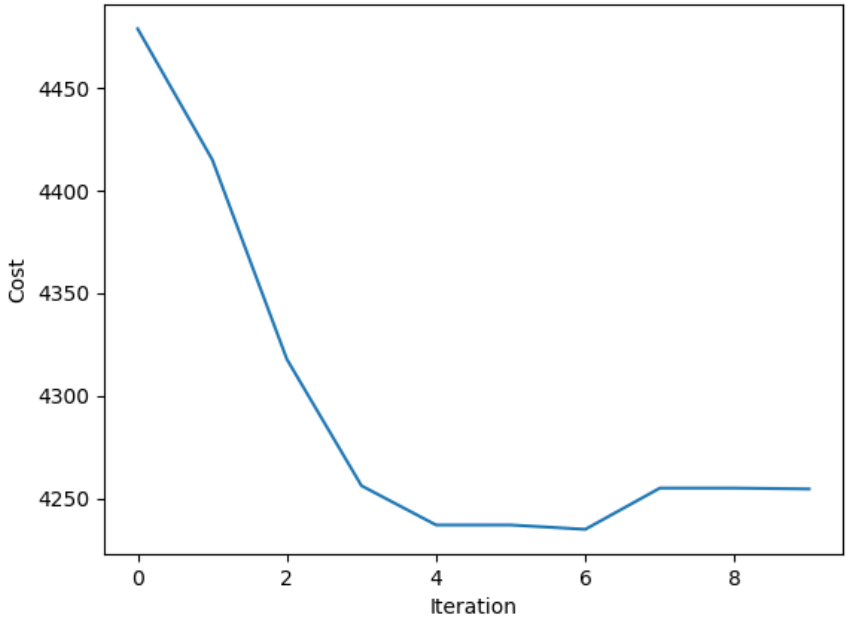


Figure 4.2: Cost-iteration plot of the 6-bus ACOPF solution.

4.2 12-bus SCOPF Solution

In this example, also found in [9], the 12-bus system depicted in Figure 4.3 are considered. In [9], the example is only solved using DC SCOPF. Here, the example is somewhat modified, and is used to demonstrate a nonlinear SCOPF solution, using the algorithm described in section 3.2. The system consists of two areas, which are copies of the 6-bus system considered in the previous section. Some line limits have been increased, and the loads and the generation costs are different. Complete system data can be found in section A.2. The essence is that *Area 1* has a higher load, but the generation cost is much lower in *Area 2*. It's therefore desirable to transfer power from *Area 2* to *Area 1*, but the amount is limited by the system constraints. The two areas are interconnected by two lines (3-9 and 4-10) with a limited transfer capacity of $0.3\ pu\ MW$ each (chosen for the sake of the example). We want to find the lowest cost dispatch which ensures the safe operation of the system, should either of the lines 3-9 or 4-10 fall out of operation. This is a SCOPF problem where we are considering the two contingencies: "failure of line 3-9" and "failure of line 4-10". The problem is solved for the three levels of security: "no security constraints", "preventive security" and "corrective security".

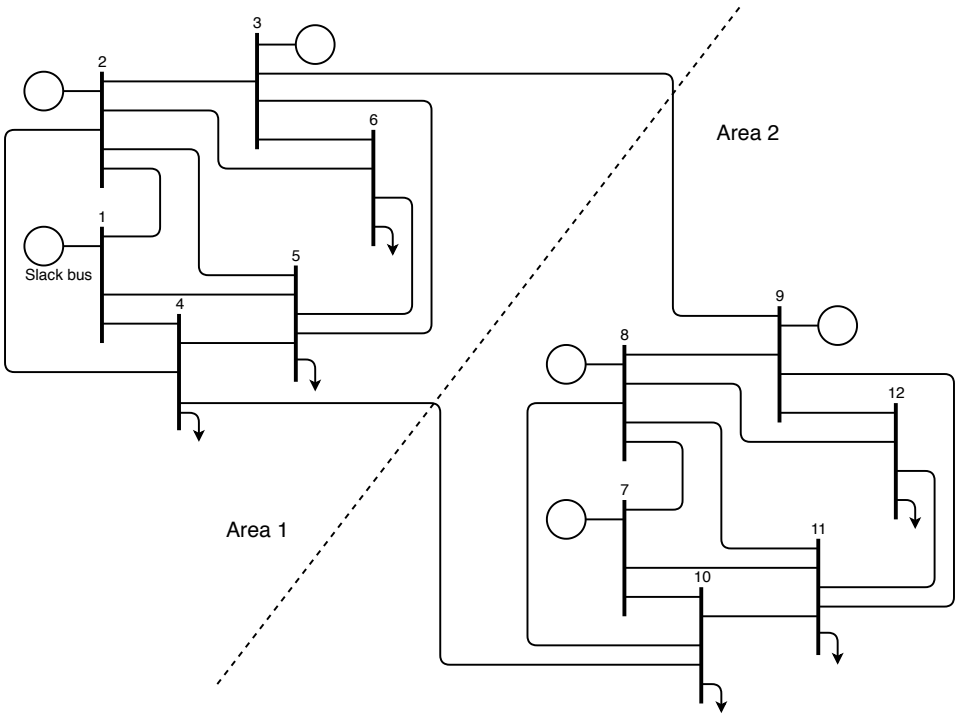


Figure 4.3: 12-bus power flow system. Based on [9]

Table 4.6: Initial PF for the 12-bus system.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	1.9239	-0.23613	1.07	0
2	0.5	-0.14059	1.05	-0.10603
3	0.5	0.20653	1.05	-0.1281
4	-1.10	-0.15	1.05891	-0.1371
5	-1.10	-0.15	1.02416	-0.17196
6	-1.10	-0.15	1.02372	-0.18545
7	1.10	-0.43405	1.05	0.19112
8	0.50	-0.34775	1.05	0.13319
9	0.50	-0.13425	1.05	0.1118
10	-0.50	-0.15	1.06886	0.09099
11	-0.50	-0.15	1.04024	0.09763
12	-0.50	-0.15	1.03638	0.09071
Total Generation	5.02390	-1.08625		
Total Loss	0.2239	-1.98625		
Total Cost	9391.88 $/MWh			

4.2.1 No Security Constraints

The solution to the ACOPF problem without considering contingencies are presented in Table 4.8. Table 4.7 shows the flow on lines 3-9 and 4-10 for the obtained solution. It is also tested what happens if one of the lines 3-9 or 4-10 is removed, and the resulting line flows are presented in Table 4.9 and Table 4.10. The complete flow tables including all lines can be found in Appendix B. Figure 4.4 shows how the total operating cost changes for each iteration of the program.

It can be observed that the operation cost is reduced from the initial cost of 9392 $/MWh (Table 4.6), to 9003 $/MWh (Table 4.8). The branch constraints during normal operation are successfully met. However, it's clear that a failure of line 3-9 would lead to overload on line 4-10 and wise versa (Table 4.9 and Table 4.10). To solve this issue, a SCOPF solution is needed.

Table 4.7: Flow on lines 3-9 and 4-10 for the final ACOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	-0.29066	0.19963	0.35261	0.3
4	10	-0.30000	0.20399	0.36278	0.3

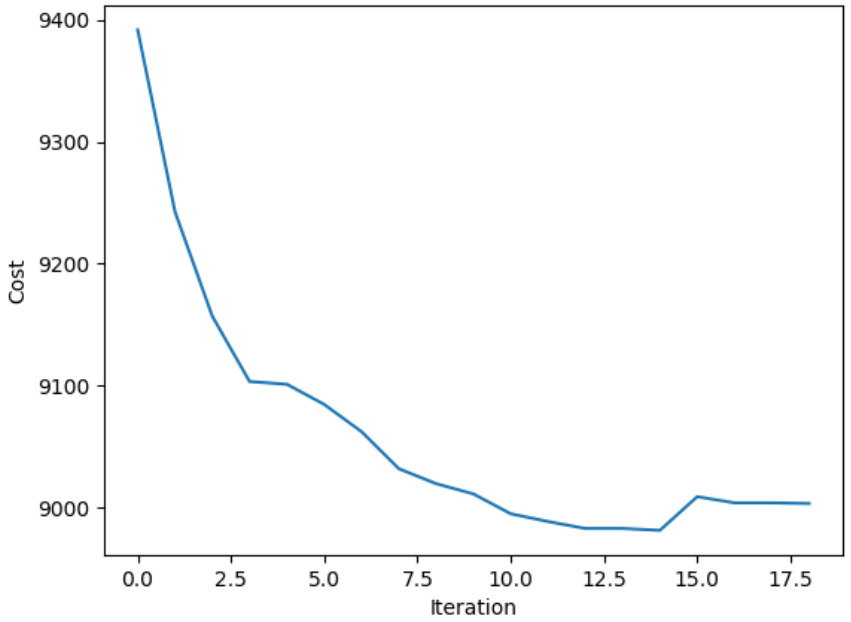


Figure 4.4: ACOPF cost-iteration plot for 12-bus system solution.

Table 4.8: Bus info for final solution of ACOPF on the 12-bus system.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.72938	-0.11696	1.07	0
2	1.13852	-0.29669	1.06902	-0.00776
3	0.91376	0.15723	1.07	-0.00112
4	-1.10	-0.15	1.06716	-0.06552
5	-1.10	-0.15	1.03817	-0.08737
6	-1.10	-0.15	1.04387	-0.07157
7	0.5	-0.45378	1.04108	0.28964
8	0.88815	-0.45897	1.05926	0.2816
9	0.83129	-0.08719	1.07	0.27523
10	-0.50	-0.15	1.07	0.21998
11	-0.50	-0.15	1.048	0.23823
12	-0.50	-0.15	1.0522	0.24645
Total Generation	5.00109	-1.25637		
Total Loss	0.20109	-2.15637		
Total Cost	9003.43 $/MWh			

Table 4.9: Flow on lines 3-9 and 4-10 for when line 3-9 is removed.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	0	0	0	0.3
4	**10**	**-0.50149**	0.44985	0.67369	**0.3**

Table 4.10: Flow on lines 3-9 and 4-10 for when line 4-10 is removed.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	**9**	**-0.52004**	0.47785	0.70625	**0.3**
4	10	0	0	0	0.3

4.2.2 Preventive Security

Now, the SCOPF algorithm is employed to ensure post-contingency security for the contingency cases ”failure of line 3-9” and ”failure of line 4-10”. We are demanding preventive security. That is, the system should stay in a feasible state after a contingency, without doing any rescheduling of active power between the generator buses. Therefore:

$$\mu = 0 \;\Rightarrow\; \Delta_i = 0 \;\forall\; i \in \boldsymbol{N}.$$

The result is presented in Table 4.11, and the changes in cost over the OPF iterations can be seen in Figure 4.5. The program spends 26 iterations before final convergence and ends at a cost of 9116 $/MWh. That is an increase of 113 $/MWh, or 1.25 %, compared to when no contingencies are considered. Table 4.12 shows that the lines 3-9 and 4-10 are now operated well below their rated limits. If one of the two lines are now disconnected, the other should not be overloaded. Table 4.13 and Table 4.14 show that this is in fact the case.

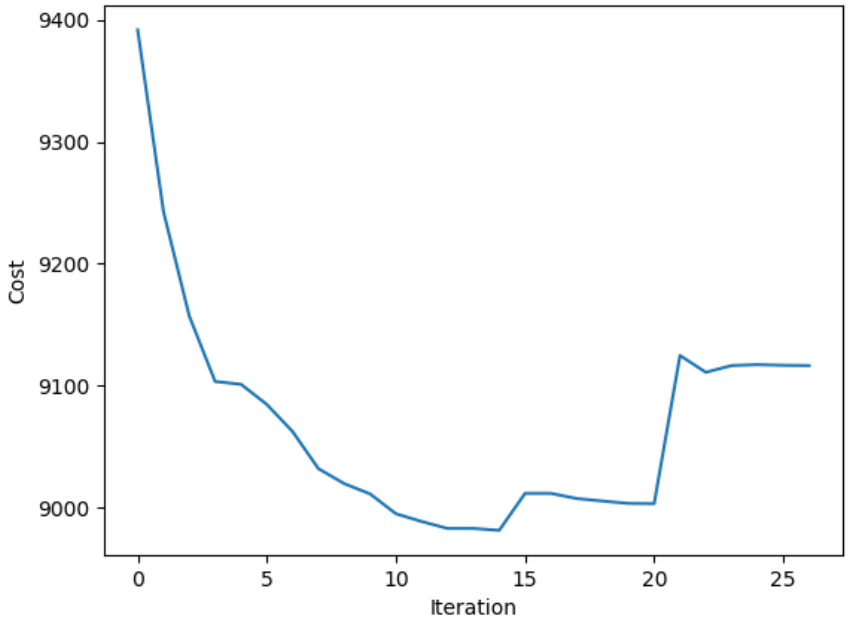


Figure 4.5: PSCOPF cost-iteration plot for 12-bus system solution.

Table 4.11: Bus info for final solution of PSCOPF on the 12-bus system.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.81571	-0.15633	1.07	0
2	1.17438	-0.31105	1.06973	-0.01156
3	1.07653	0.00988	1.0683	-0.00295
4	-1.10	-0.15	1.07	-0.0767
5	-1.10	-0.15	1.03832	-0.09075
6	-1.10	-0.15	1.04317	-0.07421
7	0.50001	-0.40223	1.04326	0.14224
8	0.71544	-0.50745	1.05436	0.13116
9	0.65294	-0.01971	1.06794	0.12594
10	-0.50	-0.15	1.07	0.07949
11	-0.50	-0.15	1.04628	0.09025
12	-0.50	-0.15	1.04941	0.09691
Total Generation	4.93502	-1.38689		
Total Loss	0.13502	-2.28689		
Total Cost	9116.48 $/MWh			

Table 4.12: Flow on lines 3-9 and 4-10 for the final PSCOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	-0.14173	0.08317	0.16433	0.3
4	10	-0.17112	0.10298	0.19972	0.3

Table 4.13: Branch flows on lines 3-9 and 4-10 for final PSCOPF solution for the 12-bus system, when line 3-9 are disconnected.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	0	0	0	0.3
4	10	-0.28759	0.19724	0.34873	0.3

Table 4.14: Branch flows on lines 3-9 and 4-10 for final PSCOPF solution for the 12-bus system, when line 4-10 are disconnected.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	-0.3002	0.20924	0.36593	0.3
4	10	0	0	0	0.3

4.2.3 Corrective Security

Instead of demanding preventive security, we are now assuming that each generator can adjust its active power output by 5% of its rated power in case of a contingency before an overload becomes critical. Therefore $\mu = 0.05$, and

$$\Delta_i = \mu P_i^{G,max} \quad \forall\, i \in \boldsymbol{G}. \tag{4.1}$$

The result is presented in Table 4.15 and a plot of the cost after each iteration is shown in Figure 4.6. The program is now converging after 24 iterations at a cost of 9040 $/MWh, which, as expected, is somewhere in between the ”no security” and the ”preventive security” solution. The cost increase from the ”no security constraints” solution is now only 0.41 %. Table 4.16 shows that the lines 3-9 and 4-10 are operated below their limits, but not as far below as with the PSCOPF solution.

To test whether the CSCOPF solution achieves the desired outcome, it’s not enough to just remove a line and run a PF analysis, as with the PSCOPF solution. Also, the corrective actions that were the prerequisites for deeming the post-contingency subproblem feasible, must be performed. For the active power, that is the value of the variables $\boldsymbol{s}$ from the last subproblem execution of one particular

contingency. In Table 4.17 and Table 4.18, these values are denoted as ΔP, since they are adjustments in the active power. In addition, the required changes in voltage magnitude from the subproblem are also included. These two quantities are the specified (controllable) variables of generator buses.

To test the contingency cases with the CSCOPF solution, the lines 3-9 and 4-10 are removed in turn, and the corresponding changes of Table 4.17 and Table 4.18 are made before a PF solution is found. Table 4.19 and Table 4.20 show the resulting flows on these lines, and it can be observed that the active PF is in fact below the limit. There are however some issues. Since the adjustments of Table 4.17 and Table 4.18 are based on linearization around the pre-contingency operating point, the suggested post-contingency corrective actions are not fully accurate. The resulting increase in losses are also not taken into account, which leads to the slack bus actually increasing its active power production by more than the allowed $\Delta_1 = 0.10$ (see Table B.8 and Table B.10). Also, here, the focus has been on the active power production and flow. It can, however, be observed that the measures made to keep the active PF on lines 3-9 and 4-10 below their limits in the post-contingency cases, actually lead to a significant increase in reactive PF on the same line. In reality, this would be counterproductive.

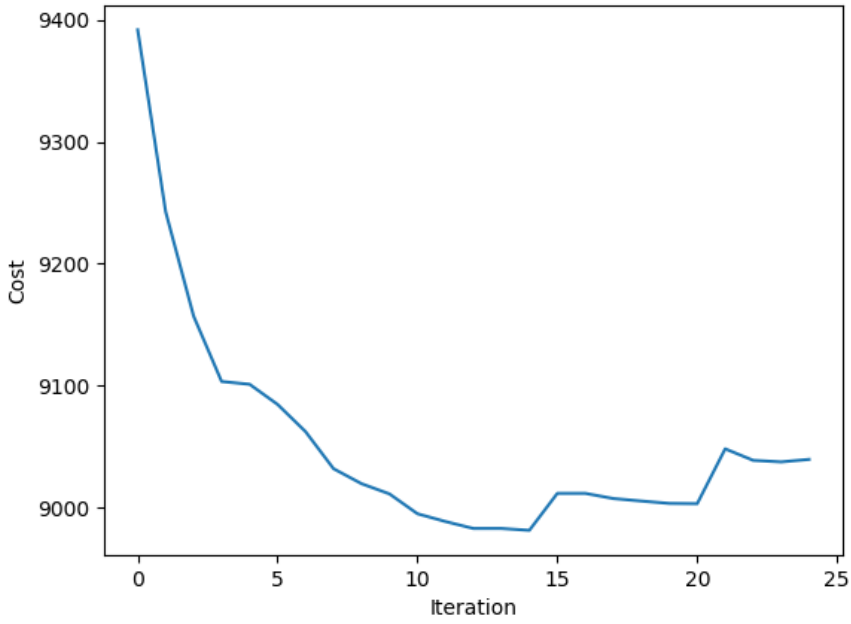


Figure 4.6: CSCOPF cost-iteration plot for 12-bus system solution.

Table 4.15: Bus info for final solution of CSCOPF on the 12-bus system.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.74797	-0.08115	1.07	0
2	1.13012	-0.33301	1.06524	-0.00665
3	1.02006	0.06621	1.06474	0.00442
4	-1.10	-0.15	1.06561	-0.0686
5	-1.10	-0.15	1.03482	-0.08609
6	-1.10	-0.15	1.03932	-0.06838
7	0.50001	-0.40559	1.04377	0.22981
8	0.81779	-0.52176	1.0563	0.22162
9	0.75193	-0.03965	1.07	0.2151
10	-0.50	-0.15	1.07	0.16359
11	-0.50	-0.15	1.04774	0.17873
12	-0.50	-0.15	1.05134	0.18636
Total Generation	4.96787	-1.31494		
Total Loss	0.16787	-2.21494		
Total Cost	9039.55 $/MWh			

Table 4.16: Branch flows for lines 3-9 and 4-10 for the final CSCOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	-0.22846	0.13872	0.26727	0.3
4	10	-0.24941	0.15711	0.29477	0.3

Table 4.17: Suggested rescheduling actions for contingency ”Failure of line 3-9”.

Bus	ΔP	ΔV	Δ
1	0.10000	0.00000	0.10
2	0.07500	-0.08714	0.075
3	-0.03531	-0.09037	0.090
7	-0.00000	-0.09377	0.100
8	-0.07500	-0.09729	0.075
9	-0.09000	-0.08881	0.090

Table 4.18: Suggested rescheduling actions for contingency ”Failure of line 4-10”.

Bus	ΔP	ΔV	Δ
1	0.10000	0.00000	0.10
2	0.07500	-0.00403	0.075
3	-0.04182	0.00526	0.090
7	-0.00001	-0.09145	0.100
8	-0.07500	-0.05570	0.075
9	-0.09000	-0.12000	0.090

Table 4.19: Branch flows for final CSCOPF solution for the 12-bus system, when line 3-9 are disconnected and the changes of Table 4.17 have been applied.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	0	0	0	0.3
4	10	-0.2966	0.26265	0.39618	0.3

Table 4.20: Branch flows for final CSCOPF solution for the 12-bus system, when line 4-10 are disconnected and the changes of Table 4.18 have been applied.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
3	9	-0.28347	0.39725	0.48802	0.3
4	10	0	0	0	0.3

Chapter 5

Discussion

Where chapter 4 provided some discussion on the studied numerical examples, this chapter provides a more general discussion on the taken approaches and the obtained results, with a special focus on success factors and improvement areas.

5.1 Method Choices

As mentioned, there are several different ways to solve both regular OPF problems and SCOPF problems. Important for this thesis was to identify a method that is not too challenging to understand, and that could be implemented in Python without using commercial software. For this reason, SLP for solving the ACOPF problem was a natural choice. It is a well used and proven method and is relatively easy to understand and implement for someone without a background in operations research. The wide availability and robustness of LP solvers also talk in favour of this approach. There may, of course, be methods better suited, and later studies could compare their performance to the one chosen here.

As previously described, the BD approach was chosen for handling the contingency constraints of the SCOPF problem. Other approaches were not widely considered, as the BD approach seemed to fit the problem well. It seems to be the most widely used method in the literature and has a long history. It fits well with the SLP method, as the subproblems produce linear constraints that can be easily added to the LP master problem. Also, the ability to calculate the subproblems in parallel can prove advantageous in future implementations using parallel computing. The issue that convergence to the global optimum cannot be guaranteed, is, of course, something to keep in mind.

For solving the subproblem and generating BC's, LP appears to be a natural

choice. That is because the chosen objective function is linear. The constraints still have to be linearized, but this linearization is quite good close to the linearization point. It may, however, be problematic in the CSCOPF case, as observed in subsection 4.2.3.

For solving the master problem in each OPF iteration, it may be discussed whether the chosen method of LP with TR is the best approach. Instead of linearizing the cost function based on the slope at the current operation point, some formulations divide the cost function into linear segments. This is sometimes referred to as *piecewise linear programming*. The advantage of this approach is that the solution may be found sooner, as it is possible to make larger steps. However, with this approach, there are more variables to keep track on, as each segment must have a designated variable in the LP calculation. It was chosen not to follow this approach, as it didn't give any clear computational advantage on the small 6-bus system. It's, however, possible that the advantage is greater for larger systems. Should that prove to be the case, it shouldn't be too much work to change to this approach. Another improvement to consider is to use QP instead of LP for the master problem. That could be advantageous for the obvious reason that the considered cost function is quadratic. This has not been tested, but should be considered in a future, more advanced implementation.

5.2 ACOPF Solution

The ACOPF algorithm described in subsection 3.2.1 and demonstrated in section 4.1 and subsection 4.2.1 shows promising results. It performs very well for the 6-bus system, where the solution was found after 9 iterations. Both for the 6-bus and the 12-bus system, an operating state that is significantly cheaper than the initial state are found. A concern is however that the number of iterations needed for convergence seems to increase with increasing system size, as the 12-bus system converges in 18 iterations. As both an LP and a PF calculation are needed for every iteration, the computational burden can potentially be quite high for large systems. The efficiency of the algorithms have not been closely studied, but it's something to consider in a more developed implementation. The efficiency of the PF calculations could, for instance, be improved by using a decoupled PF method, as described in subsection 2.1.6. The number of iterations could be reduced by using QP or piecewise linear programming, as mentioned in the previous section.

5.3 SCOPF Solution

The demonstrated handling of security constraints has proven to be good for providing preventive security and looks promising for corrective security. In subsection 4.2.2, it was demonstrated how preventive security was achieved quite precisely, and the new solution needed only 5 extra iterations after the base-case solution was found. It could also be observed how the extra security requirements led to a quite significant increase in cost. This observation can serve as a motivation for why corrective security is appealing.

In subsection 4.2.3, it's demonstrated how the operation cost can be lowered, without necessarily sacrificing security, by also considering the generators' corrective capabilities. This cost reduction could potentially be quite large for a larger system, and the savings considerable when considering a longer time horizon. As also mentioned in subsection 4.2.3, the CSCOPF solution becomes increasingly imprecise for larger post-contingency rescheduling. This is because of the nonlinearities between the pre- and post-contingency operating states, as the subproblem is linearized around the pre-contingency operating state. Further investigation into the significance of this mismatch should be considered if this method for including post-contingency rescheduling in the SCOPF problem are to be used in a future implementation.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

This thesis aimed to identify a method for solving non-linear SCOPF problems and make a prototype implementation in Python. The ACOPF problem was solved by SLP with a TR method and then combined with a BD approach for handling the contingency constraints of the SCOPF problem. A prototype Python program was successfully implemented and demonstrated on two illustrative numerical examples. The results show great promise and demonstrate the potential for cost savings if adopted.

The theoretical background has been laid out, and the proposed solution algorithms thoroughly explained. It has been shown how one can utilize available LP solvers for solving nonlinear problems by repetitive linearization. The adopted TR method helps to make steps of appropriate size from one iteration to the next. The SLP method and BD have proven to work well together. The linear constraints obtained from the BD subproblems can easily be included in the LP calculation of the SLP iterations.

The numerical examples demonstrate that the program works as intended. For both systems, a lower-cost operating state that satisfies all the included system constraints are found. With the 12-bus system, it's demonstrated how the inclusion of security constraints affect the solution. As expected, the demand for "preventive security" led to the highest operating cost. When also considering the possibility for post-contingency rescheduling, it was shown that the system can be operated at a lower cost. Testing showed that the obtained SCOPF solutions did stay feasible for the considered contingency cases. It was however noted that that the CSCOPF solution might not be completely reliable if the allowed corrective rescheduling is large.

This work has demonstrated both the usefulness of SCOPF and shown an example of how to formulate and solve the problem. This could prove a helpful step in the development of smart algorithms for planning and operation of the future distribution grid. The experiences so far have been promising, and with further work, the presented approach may prove a valuable addition to the toolbox for planning and operation of microgrids.

6.2 Further Work

This thesis work should serve as a foundation for developing a more general and practically applicable implementation of SCOPF for distribution grid applications. To reach this goal, there are several improvements and additional features that should be considered for further work. Some elements to consider are listed below.

- The test systems used here are theoretical and of limited size. Further work could include testing on larger and more realistic systems.
- In this thesis, the emphasis has been on active power dispatch. A suggestion for further work is to also include the cost of reactive power dispatch in the problem. For instance can *reactive power pricing* be considered.
- The branch flow constraints in this implementation were also based on active power. As mentioned in subsection 2.2.3, it's more realistic to use current or apparent power limits for flow constraints in distribution grids. Further work could include implementing these type of flow constraints.
- In this implementation, only one time-step is considered. The introduction of varying RES and energy storage in future distribution systems makes it necessary to optimize operation over a given period. Further work could, therefore, investigate how the algorithm can be extended to multi-period OPF.
- It could be considered to incorporate advanced controls, such as on-load tap changers, phase shifters, series and shunt capacitors, and flexible AC transmission systems devices.
- Other types of contingencies could be considered. For instance generator-, bus-, load- or shunt outages.
- Investigate and implement a contingency filtering technique for identifying relevant contingencies to include in the set of contingencies.

Bibliography

[1] L. Ødegården and S. Bhatana, "Status og prognoser for kraftsystemet 2018," Oslo, Tech. Rep., 2018.

[2] H. Abdi, S. D. Beigvand, and M. L. Scala, "A review of optimal power flow studies applied to smart grids and microgrids," *Renewable and Sustainable Energy Reviews*, vol. 71, pp. 742–766, 2017.

[3] IEEE Power and Energy Society, "IEEE standard for the specification of microgrid controllers," *IEEE Std 2030.7-2017*, pp. 1–43, Apr. 2018.

[4] F. Capitanescu, J. L. Martinez Ramos, P. Panciatici, D. Kirschen, A. Marano Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-art, challenges, and future trends in security constrained optimal power flow," *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, 2011.

[5] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016.

[6] J. Carpentier, "Contribution to the economic dispatch problem," *Bulletin de la Société Française des Electriciens*, vol. 8, pp. 431–447, 1962.

[7] O. Alsac and B. Stott, "Optimal load flow with steady-state security," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 745–751, 1974.

[8] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Transactions on Power Systems*, vol. 2, no. 1, pp. 175–180, 1987.

[9] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power generation, operation and control*, eng, 3rd ed. Hoboken, N.J: Wiley, 2014.

[10] H. W. Dommel and W. F. Tinney, “Optimal power flow solutions,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, pp. 1866–1876, 1968.

[11] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, “Optimal power flow by newton approach,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 10, pp. 2864–2880, 1984.

[12] B. Stott and E. Hobson, “Power system security control calculations using linear programming, part ii,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, no. 5, pp. 1721–1731, 1978.

[13] O. Alsac, J. Bright, M. Prais, and B. Stott, “Further developments in lp-based optimal power flow,” *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 697–711, 1990.

[14] G. L. Torres and V. H. Quintana, “An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates,” *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1211–1218, 1998.

[15] F. Capitanescu, M. Glavic, and L. Wehenkel, “An interior-point method based optimal power flow,” Jun. 2005.

[16] R. Madani, M. Ashraphijuo, and J. Lavaei, “Promises of conic relaxation for contingency-constrained optimal power flow problem,” *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1297–1307, 2016.

[17] A. Venzke and S. Chatzivasileiadis, “Convex relaxations of security constrained ac optimal power flow under uncertainty,” in *2018 Power Systems Computation Conference (PSCC)*, 2018, pp. 1–7.

[18] F. Capitanescu, “Critical review of recent advances and further developments needed in ac optimal power flow,” *Electric Power Systems Research*, vol. 136, pp. 57–68, 2016.

[19] J. D. Glover, M. S. Sarma, and T. J. Overbye, *Power system analysis and design*, eng, 5th ed. Stamford, USA: Cengage Learning, 2012.

[20] P. S. R. Murty, “Chapter 10 - power flow studies,” in *Power Systems Analysis*, P. S. R. Murty, Ed., Second Edition, Boston, USA: Butterworth-Heinemann, 2017, pp. 205–276.

[21] M. Farivar and S. H. Low, “Branch flow model: Relaxations and convexification,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3672–3679.

[22] A. S. Debs, “Load flow analysis,” in *Modern Power Systems Control and Operation*. Boston, MA: Springer US, 1988, pp. 19–86.

[23] R. Gnanavignesh and U. J. Shenoy, "Parallel sparse lu factorization of power flow jacobian using gpu," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 1857–1862.

[24] A. J. Monticelli, A. Garcia, and O. R. Saavedra, "Fast decoupled load flow: Hypothesis, derivations, and testing," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1425–1431, 1990.

[25] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 859–869, 1974.

[26] B. Stott, J. Jardim, and O. Alsac, "Dc power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[27] A. Sallam and O. Malik, *Power System Stability: Modelling, Analysis and Control*, ser. Energy Engineering. Institution of Engineering and Technology, 2015.

[28] S. Chatzivasileiadis, "Lecture notes on optimal power flow (opf)," *ArXiv*, vol. abs/1811.00943, 2018, (accessed 07.05.2020). [Online]. Available: `https://arxiv.org/abs/1811.00943`.

[29] A. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, pp. 237–260, Oct. 1972.

[30] I. B. Sperstad and H. Marthinsen, "Optimal power flow methods and their application to distribution systems with energy storage," SINTEF Energy Research, Trondheim, TR A7604, version 1.0, 2016.

[31] J. Nocedal and S. Wright, *Numerical Optimization*, eng, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer, 2006, ISBN: 0-387-40065-6.

[32] F. Palacios-Gomez, L. Lasdon, and M. Engquist, "Nonlinear optimization by successive linear programming," *Management Science*, vol. 28, no. 10, pp. 1106–1120, 1982, ISSN: 00251909, 15265501.

[33] E. Riccietti, S. Bellavia, and S. Sello, "Sequential linear programming and particle swarm optimization for the optimization of energy districts," *Engineering Optimization*, vol. 51, Jan. 2018.

[34] A. M. Giacomoni and B. F. Wollenberg, "Linear programming optimal power flow utilizing a trust region method," in *North American Power Symposium 2010*, 2010, pp. 1–6.

[35] (). Accessed: 02.06.2020, [Online]. Available: `http://lpsolve.sourceforge.net/5.5/Python.htm`.

[36] (). Accessed: 02.06.2020, [Online]. Available: https://github.com/boris-arzur/lp_solve-python/blob/master/lp_maker.py.

Appendix A

System Data

A.1 6-bus System

Table A.1: Generator data for 6-bus system. Found in [9].

Gen. bus	a	b	c	P^{min} (MW)	P^{max} (MW)	Q^{min} (MVAR)	Q^{max} (MVAR)	V (pu)
1	213.1	11.669	0.00533	50.0	200.0	-100.0	150.0	1.07
2	200.0	10.333	0.00889	37.5	150.0	-100.0	150.0	1.05
3	240.0	10.833	0.00741	45.0	180.0	-100.0	120.0	1.05

Table A.2: Load data for 6-bus system. Found in [9].

Load bus	P^L (MW)	Q^L (MVAR)
4	100.0	15.0
5	100.0	15.0
6	100.0	15.0

Table A.3: Branch data for 6-bus system. Found in [9].

From bus	To bus	R (pu)	X (pu)	B_{cap} (pu)	P_{ij}^{max} (MW)
1	2	0.10	0.20	0.04	100.0
1	4	0.05	0.20	0.04	100.0
1	5	0.08	0.30	0.06	100.0
2	3	0.05	0.25	0.06	60.0
2	4	0.05	0.10	0.02	60.0
2	5	0.10	0.30	0.04	60.0
2	6	0.07	0.20	0.05	60.0
3	5	0.12	0.26	0.05	60.0
3	6	0.02	0.10	0.02	60.0
4	5	0.20	0.40	0.08	60.0
5	6	0.10	0.30	0.06	60.0

A.2 12-bus System

Table A.4: Generator data for 12-bus system. Found in [9].

Gen. bus	a	b	c	P^{min} (MW)	P^{max} (MW)	Q^{min} (MVAR)	Q^{max} (MVAR)	V (pu)
1	319.65	17.5035	0.007995	50.0	200.0	-100.0	150.0	1.07
2	300.0	15.4995	0.013335	37.5	150.0	-100.0	150.0	1.05
3	360.0	16.2495	0.011115	45.0	180.0	-100.0	120.0	1.05
7	213.1	11.669	0.00533	50.0	200.0	-100.0	150.0	1.07
8	200.0	10.333	0.00889	37.5	150.0	-100.0	150.0	1.05
9	240.0	10.833	0.00741	45.0	180.0	-100.0	120.0	1.05

Table A.5: Load data for 12-bus system. Found in [9].

Load bus	P^L (MW)	Q^L (MVAR)
4	110.0	15.0
5	110.0	15.0
6	110.0	15.0
10	50.0	15.0
11	50.0	15.0
12	50.0	15.0

Table A.6: Branch data for 12-bus system. Found in [9], but with some adjustments.

From bus	To bus	R (pu)	X (pu)	B_{cap} (pu)	P_{ij}^{max} (MW)
1	2	0.10	0.20	0.04	100.0
1	4	0.05	0.20	0.04	100.0
1	5	0.08	0.30	0.06	100.0
2	3	0.05	0.25	0.06	60.0
2	4	0.05	0.10	0.02	120.0
2	5	0.10	0.30	0.04	60.0
2	6	0.07	0.20	0.05	60.0
3	5	0.12	0.26	0.05	60.0
3	6	0.02	0.10	0.02	120.0
4	5	0.20	0.40	0.08	60.0
5	6	0.10	0.30	0.06	60.0
7	8	0.10	0.20	0.04	100.0
7	10	0.05	0.20	0.04	100.0
7	11	0.08	0.30	0.06	100.0
8	9	0.05	0.25	0.06	60.0
8	10	0.05	0.10	0.02	60.0
8	11	0.10	0.30	0.04	60.0
8	12	0.07	0.20	0.05	60.0
9	11	0.12	0.26	0.05	60.0
9	12	0.02	0.10	0.02	120.0
10	11	0.20	0.40	0.08	60.0
11	12	0.10	0.30	0.06	60.0
3	9	0.40	0.80	0.16	30.0
4	10	0.40	0.80	0.16	30.0

Appendix B

Additional results tables

B.1 12-bus - No Contingency Constraints

Table B.1: Branch flows for final ACOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.03768	-0.0134	0.03999	1.0
1	4	0.35826	-0.06214	0.36361	1.0
1	5	0.33344	0.03872	0.33568	1.0
2	3	-0.03	0.00189	0.03005	0.6
2	4	0.54237	-0.23236	0.59005	1.2
2	5	0.30128	0.02119	0.30202	0.6
2	6	0.36241	0.01891	0.36291	0.6
3	5	0.3593	-0.01897	0.3598	0.6
3	6	0.81508	0.14427	0.82775	1.2
4	5	0.07962	0.03819	0.08831	0.6
5	6	-0.05715	-0.00021	0.05715	0.6
7	8	-0.00231	-0.09329	0.09332	1.0
7	10	0.33261	-0.22017	0.39888	1.0
7	11	0.16969	-0.06445	0.18152	1.0
8	9	0.01905	-0.04922	0.05277	0.6
8	10	0.52142	-0.35296	0.62965	0.6
8	11	0.15735	-0.00922	0.15762	0.6
8	12	0.18722	-0.02466	0.18884	0.6
9	11	0.16707	0.01637	0.16787	0.6
9	12	0.34904	0.12536	0.37087	1.2
10	11	-0.0172	0.06791	0.07005	0.6
11	12	-0.03154	-0.00401	0.03179	0.6
3	**9**	**-0.29066**	0.19963	0.35261	**0.3**
4	**10**	**-0.30000**	0.20399	0.36278	**0.3**

Table B.2: Branch flow of the 12-bus system after removal of line 3-10 with the standard ACOPF solution of subsection 4.2.1.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.09258	-0.03991	0.10082	1
1	4	0.32979	-0.0125	0.33003	1
1	5	0.38853	0.03393	0.39001	1
2	3	0.07783	-0.01905	0.08012	0.6
2	4	0.40156	-0.08243	0.40993	1.2
2	5	0.31421	0.02242	0.31501	0.6
2	6	0.43662	0.00364	0.43663	0.6
3	5	0.28472	0.01385	0.28505	0.6
3	6	0.70658	0.16865	0.72643	1.2
4	5	0.12073	0.00011	0.12073	0.6
5	6	-0.02224	-0.01334	0.02594	0.6
7	8	-0.03063	-0.0793	0.08501	1
7	10	0.4021	-0.18732	0.4436	1
7	11	0.12853	-0.04961	0.13777	1
8	9	-0.10524	-0.02334	0.10779	0.6
8	**10**	**0.72195**	-0.34654	0.80081	**0.6**
8	11	0.13888	0.00205	0.13889	0.6
8	12	0.10126	0.00028	0.10127	0.6
9	11	0.25146	-0.01089	0.2517	0.6
9	12	0.47407	0.09957	0.48441	1.2
10	11	-0.07704	0.08071	0.11158	0.6
11	12	-0.07015	0.00179	0.07017	0.6
3	9	0	0	0	0.3
4	**10**	**-0.50149**	0.44985	0.67369	**0.3**

Table B.3: Branch flow of the 12-bus system after removal of line 4-10 with the standard ACOPF solution of subsection 4.2.1.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.05226	-0.02052	0.05614	1.0
1	4	0.43506	0.1025	0.44697	1.0
1	5	0.32687	0.0593	0.3322	1.0
2	3	-0.11672	0.02073	0.11854	0.6
2	4	0.71946	0.0436	0.72078	1.2
2	5	0.2855	0.04448	0.28894	0.6
2	6	0.30226	0.03903	0.30477	0.6
3	5	0.41652	-0.01717	0.41687	0.6
3	6	0.89995	0.13962	0.91072	1.2
4	5	0.02306	-0.00951	0.02495	0.6
5	6	-0.0814	-0.00882	0.08188	0.6
7	8	0.04558	-0.11643	0.12504	1.0
7	10	0.23019	-0.02787	0.23187	1.0
7	11	0.22423	-0.05675	0.2313	1.0
8	9	0.15527	-0.07337	0.17173	0.6
8	10	0.31915	0.0862	0.33059	0.6
8	11	0.17593	0.0047	0.17599	0.6
8	12	0.28193	-0.04213	0.28507	0.6
9	11	0.07815	0.07735	0.10996	0.6
9	12	0.21278	0.172	0.27361	1.2
10	11	0.04199	-0.03614	0.0554	0.6
11	12	0.01176	-0.03023	0.03243	0.6
3	**9**	**-0.52004**	0.47785	0.70625	**0.3**
4	10	0	0	0	0.3

B.2 12-bus - PSCOPF

Table B.4: Branch flows for final PSCOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.05364	-0.025	0.05918	1
1	4	0.4168	-0.08736	0.42586	1
1	5	0.34528	0.03618	0.34717	1
2	3	-0.03661	0.01363	0.03907	0.6
2	4	0.60466	-0.28091	0.66673	1.2
2	5	0.30071	0.02339	0.30162	0.6
2	6	0.35895	0.02738	0.35999	0.6
3	5	0.36154	-0.02724	0.36257	0.6
3	6	0.82003	0.13272	0.8307	1.2
4	5	0.06524	0.05241	0.08368	0.6
5	6	-0.05864	0.00324	0.05873	0.6
7	8	0.02569	-0.07044	0.07498	1
7	10	0.29919	-0.20331	0.36173	1
7	11	0.17513	-0.05229	0.18277	1
8	9	0.01159	-0.05954	0.06066	0.6
8	10	0.40626	-0.35294	0.53816	0.6
8	11	0.14481	-0.01678	0.14578	0.6
8	12	0.17795	-0.03294	0.18097	0.6
9	11	0.16134	0.01727	0.16226	0.6
9	12	0.35184	0.13227	0.37588	1.2
10	11	0.00135	0.06294	0.06296	0.6
11	12	-0.02519	-0.00246	0.02531	0.6
3	9	-0.14173	0.08317	0.16433	0.3
4	10	-0.17112	0.10298	0.19972	0.3

Table B.5: Branch flows for final PSCOPF solution for the 12-bus system, when line 3-9 are disconnected.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.07352	-0.0346	0.08126	1.0
1	4	0.39335	-0.07281	0.40003	1.0
1	5	0.36767	0.03327	0.36917	1.0
2	3	0.01609	0.00294	0.01636	0.6
2	4	0.52819	-0.22755	0.57512	1.2
2	5	0.30771	0.02253	0.30853	0.6
2	6	0.39533	0.01923	0.3958	0.6
3	5	0.32545	-0.0131	0.32572	0.6
3	6	0.76716	0.14345	0.78045	1.2
4	5	0.08768	0.03695	0.09515	0.6
5	6	-0.04207	-0.00212	0.04212	0.6
7	8	0.01207	-0.06379	0.06492	1.0
7	10	0.33203	-0.19846	0.38681	1.0
7	11	0.15591	-0.04674	0.16277	1.0
8	9	-0.04476	-0.04819	0.06577	0.6
8	10	0.49644	-0.37029	0.61933	0.6
8	11	0.1365	-0.01293	0.13711	0.6
8	12	0.13894	-0.02223	0.14071	0.6
9	11	0.19951	0.00319	0.19953	0.6
9	12	0.40849	0.11948	0.4256	1.2
10	11	-0.02589	0.07247	0.07696	0.6
11	12	-0.04283	0.00107	0.04285	0.6
3	9	0	0	0	0.3
4	10	-0.28759	0.19724	0.34873	0.3

Table B.6: Branch flows for final PSCOPF solution for the 12-bus system, when line 4-10 are disconnected.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.05762	-0.02694	0.06361	1
1	4	0.44067	0.09919	0.45169	1
1	5	0.33656	0.05857	0.34162	1
2	3	-0.09573	0.0264	0.09931	0.6
2	4	0.71809	0.04722	0.71964	1.2
2	5	0.29084	0.04668	0.29457	0.6
2	6	0.31845	0.04302	0.32135	0.6
3	5	0.40229	-0.01802	0.40269	0.6
3	6	0.87828	0.13369	0.8884	1.2
4	5	0.02722	-0.00967	0.02888	0.6
5	6	-0.07604	-0.00849	0.07651	0.6
7	8	0.05862	-0.0863	0.10433	1
7	10	0.23279	-0.00444	0.23283	1
7	11	0.2086	-0.03922	0.21225	1
8	9	0.08529	-0.07323	0.11241	0.6
8	10	0.30448	0.06664	0.31169	0.6
8	11	0.15407	0.00078	0.15407	0.6
8	12	0.22923	-0.0401	0.23271	0.6
9	11	0.11334	0.06147	0.12894	0.6
9	12	0.27718	0.16272	0.32142	1.2
10	11	0.03041	-0.03167	0.04391	0.6
11	12	-0.00114	-0.025	0.02503	0.6
3	9	-0.3002	0.20924	0.36593	0.3
4	10	0	0	0	0.3

B.3 12-bus - CSCOPF

Table B.7: Branch flows for final CSCOPF solution for the 12-bus system.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.04054	0.00532	0.04088	1
1	4	0.37649	-0.05724	0.38082	1
1	5	0.33094	0.05091	0.33483	1
2	3	-0.04781	0.01196	0.04928	0.6
2	4	0.56941	-0.26688	0.62885	1.2
2	5	0.29834	0.02017	0.29902	0.6
2	6	0.35057	0.02592	0.35152	0.6
3	5	0.36901	-0.03041	0.37026	0.6
3	6	0.83159	0.13371	0.84227	1.2
4	5	0.07156	0.04668	0.08544	0.6
5	6	-0.06165	0.00559	0.06191	0.6
7	8	0.01004	-0.07026	0.07097	1
7	10	0.31845	-0.20428	0.37833	1
7	11	0.17152	-0.05479	0.18006	1
8	9	0.01722	-0.06122	0.0636	0.6
8	10	0.47412	-0.36272	0.59696	0.6
8	11	0.15242	-0.01725	0.1534	0.6
8	12	0.18361	-0.03459	0.18683	0.6
9	11	0.16521	0.01823	0.16621	0.6
9	12	0.35009	0.13432	0.37498	1.2
10	11	-0.00999	0.06487	0.06563	0.6
11	12	-0.02897	-0.00281	0.02911	0.6
3	9	-0.22846	0.13872	0.26727	0.3
4	10	-0.24941	0.15711	0.29477	0.3

Table B.8: PF solution after line 3-9 is removed and the corrective actions of Table 4.17 have been applied.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.93276	0.93804	1.07	0
2	1.20512	-0.9285	0.9781	0.01445
3	0.98475	-0.02705	0.97437	0.00124
4	-1.10	-0.15	0.99594	-0.06108
5	-1.10	-0.15	0.96374	-0.09258
6	-1.10	-0.15	0.94962	-0.07573
7	0.5	-0.29775	0.95	0.3833
8	0.74279	-0.48838	0.95902	0.37548
9	0.66193	0.20571	0.98119	0.38289
10	-0.50	-0.15	0.96724	0.30224
11	-0.50	-0.15	0.94973	0.32885
12	-0.50	-0.15	0.95714	0.34249
Total Generation	5.02735561	-0.5979227		
Total Loss	0.22736	-1.49792		
Total Cost	9267.35745			

Table B.9: Branch flows for final CSCOPF solution for the 12-bus system, when line 3-9 are disconnected and the changes of Table 4.17 have been applied.

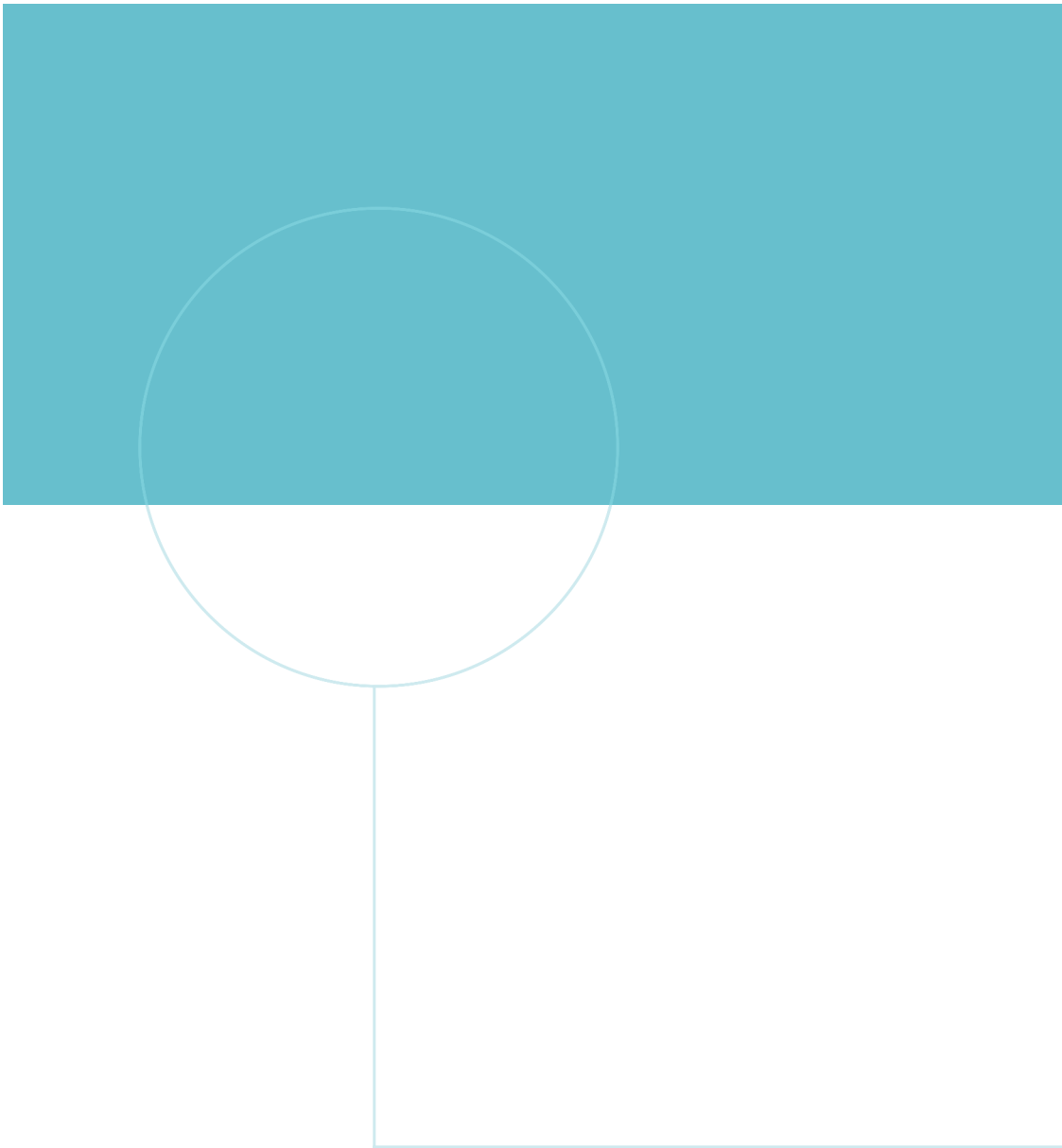
From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.13638	0.42402	0.44541	1
1	4	0.40167	0.30571	0.50478	1
1	5	0.39471	0.28845	0.48888	1
2	3	0.05131	0.00466	0.05152	0.6
2	4	0.52936	-0.41145	0.67046	1.2
2	5	0.32155	-0.04239	0.32433	0.6
2	6	0.42194	0.01049	0.42207	0.6
3	5	0.30014	-0.08281	0.31135	0.6
3	6	0.73579	0.12144	0.74574	1.2
4	5	0.09302	0.03486	0.09934	0.6
5	6	-0.03252	0.05665	0.06532	0.6
7	8	0.01143	-0.0484	0.04973	1
7	10	0.33444	-0.15039	0.36669	1
7	11	0.15413	-0.03579	0.15823	1
8	9	-0.04317	-0.07634	0.0877	0.6
8	10	0.52144	-0.31468	0.60904	0.6
8	11	0.13725	-0.01277	0.13785	0.6
8	12	0.13843	-0.03697	0.14328	0.6
9	11	0.20677	0.02854	0.20873	0.6
9	12	0.41157	0.16133	0.44206	1.2
10	11	-0.03163	0.05896	0.06691	0.6
11	12	-0.04415	-0.00847	0.04496	0.6
3	9	0	0	0	0.3
4	10	-0.2966	0.26265	0.39618	0.3

Table B.10: PF solution after line 4-10 is removed and the corrective actions of Table 4.18 have been applied.

Bus Nr.	P (pu MW)	Q (pu MVAR)	$\|V\|$ (pu)	θ (rad)
1	0.91905	0.10735	1.07	0
2	1.20512	-0.10481	1.06121	-0.01565
3	0.97824	0.44663	1.07	-0.00644
4	-1.10	-0.15	1.0286	-0.07895
5	-1.10	-0.15	1.0299	-0.09477
6	-1.10	-0.15	1.04026	-0.07846
7	0.5	-0.45501	0.95231	0.42622
8	0.74279	0.8638	1.0006	0.3899
9	0.66193	-0.63224	0.95	0.38273
10	-0.50	-0.15	0.96252	0.3687
11	-0.50	-0.15	0.94968	0.3516
12	-0.50	-0.15	0.95031	0.35184
Total Generation	5.00712	0.22572		
Total Loss	0.20712	-0.67428		
Total Cost	9229.31 $/MWh			

Table B.11: Branch flows for final CSCOPF solution for the 12-bus system, when line 4-10 are disconnected and the changes of Table 4.18 have been applied.

From Bus	To Bus	Flow (pu MW)	Flow (pu MVAR)	Flow (pu MVA)	Lim (pu MW)
1	2	0.09019	0.00265	0.09023	1
1	4	0.46463	0.12247	0.4805	1
1	5	0.36423	0.06238	0.36953	1
2	3	-0.04737	-0.02766	0.05486	0.6
2	4	0.69951	0.01811	0.69974	1.2
2	5	0.29577	0.02355	0.29671	0.6
2	6	0.34669	0.00066	0.34669	0.6
3	5	0.3773	0.00741	0.37738	0.6
3	6	0.8369	0.17965	0.85597	1.2
4	5	0.03231	-0.01916	0.03757	0.6
5	6	-0.06294	-0.01413	0.0645	0.6
7	8	0.04769	-0.25064	0.25513	1
7	10	0.23833	-0.10059	0.25869	1
7	11	0.21398	-0.0403	0.21774	1
8	9	0.06519	0.18959	0.20049	0.6
8	10	0.31664	0.22493	0.3884	0.6
8	11	0.1608	0.11858	0.19979	0.6
8	12	0.24068	0.17084	0.29515	0.6
9	11	0.09011	-0.03872	0.09808	0.6
9	12	0.26832	-0.05227	0.27336	1.2
10	11	0.04374	0.00936	0.04473	0.6
11	12	-0.00125	-0.00158	0.00202	0.6
3	9	-0.28347	0.39725	0.48802	0.3
4	10	0	0	0	0.3



NTNU

Norwegian University of
Science and Technology