THE UNIVERSITY of EDINBURGH

Structure-Exploiting Interior Point Methods for Security Constrained Optimal Power Flow Problems

Nai-Yuan Chiang

Doctor of Philosophy
University of Edinburgh
April 2013

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Nai-Yuan Chiang*)

Acknowledgement

This thesis would not have been possible without the guidance and help of several individuals, who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study, to whom I would now like to express my sincerest gratitude:

First and foremost, to my supervisor and guide Andreas Grothey, for not only providing me with numerous helpful suggestions, but also patiently guiding me to complete the PhD degree, even though I was sometimes annoying and attempted to procrastinate until the very last minutes. More specifically, Andreas gave me a fair amount of freedom and space, enabling me to develop my specialties and dig deeper into my true interests, for which I would like to show my deepest appreciation.

Secondly, I would like to express my special gratitude to my second supervisor Coralia Cartis, who was always there to support me, as well as gently remind me to keep up my progress as planned.

Furthermore, I would like to thank the University of Edinburgh, for not only offering me the opportunity to start the PhD degree, but also for providing financial assistance in the form of scholarships for over three years. I also want to thank my colleagues who share the same office with me, for those times when we discussed questions and tough topics, as they were definitely of great help during my time as a PhD student. Also, my special thanks are extended to all the staff at the University, for always helping me deal with administrative and financial issues.

What is more, as English is my second language, I would like to give special thanks to my proofreaders: Bubacarr Bah, Ronan Brennan, Pablo González-Brevis, Angel Lin, Rodrigo Lopez Sanroman, Andrew Thompson and Matthew Witherspoon, for together they enabled my dissertation to be read smoothly and correctly.

Additionally, I would like to express my appreciation to my family members, including my parents, my brother, as well as my aunt, for always being my best emotional and financial support whenever time were tough. Without them, I do not think I would have been able to finish my PhD degree as smoothly.

Last but not least, I would like to sincerely thank my partner, Chia-Jung Chang for always tolerating all of my negative emotions and also for providing encouragement, support and love.

Abstract

The aim of this research is to demonstrate some more efficient approaches to solve the n-1 security constrained optimal power flow (SCOPF) problems by using structure-exploiting primal-dual interior point methods (IPM).

Firstly, we consider a DC-SCOPF model, which is a linearized version of AC-SCOPF. One new reformulation of the DC-SCOPF model is suggested, in which most matrices that need to be factorized are constant. Consequently, most numerical factorizations and a large number of back-solve operations only need to be performed once throughout the entire IPM process. In the framework of the structure-exploiting IPM implementation, one of the major computational efforts consists of forming the Schur complement matrix, which is very computationally expensive if no further measure is applied. One remedy is to apply a preconditioned iterative method to solve the corresponding linear systems which appear in assembling the Schur complement matrix. We suggest two main schemes to pick a good and robust preconditioner for SCOPF problems based on combining different "active" contingency scenarios. The numerical results show that our new approaches are much faster than the default structure-exploiting method in OOPS, and also that it requires less memory.

The second part of this thesis goes to the standard AC-SCOPF problem, which is a nonlinear and nonconvex optimization problem. We present a new contingency generation algorithm: it starts with solving the basic OPF problem, which is a much smaller problem of the same structure, and then generates contingency scenarios dynamically when needed. Some theoretical analysis of this algorithm is shown for the linear case, while the numerical results are exciting, as this new algorithm works for both AC and DC cases. It can find all the active scenarios and significantly reduce the number of scenarios one needs to contain in the model. As a result, it speeds up the solving process and may require less IPM iterations.

Also, some heuristic algorithms are designed and presented to predict the active contingencies for the standard AC-SCOPF, based on the use of AC-OPF or DC-SCOPF. We test our heuristic algorithms on the modified IEEE 24-bus system, and also present their corresponding numerical results in the thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

During the last century, the production of electrical power has been one of the most important industries in the world. Electricity is the most popular and vital source of power in people's lives as well as in other industries. It also plays a very important role in today's world economy. There are several critical accompanied problems in this industry because of the special characteristics of electrical power.

Firstly, electricity is a secondary energy [41]. It must be converted from primary energy sources and hence the costs of supplying electrical power must reflect the price of its primary sources. Primary energy sources mainly consist of non-renewable sources and renewable sources. Generally, renewable sources, such as solar energy and wind energy, are comparatively cheap to exploit, but the problem is that they are not yet widely used. Hence, electricity is mostly converted from non-renewable energy sources, such as oil and coal. Due to the rapid growth in demand and the limited supply of oil, coal and natural gas, the cost of electrical power has been increasing in recent decades and it seems likely that this price will continue to rise in the following decades. Additionally, the carbon dioxide emission also becomes a serious problem because lots of non-renewable sources are overused.

Secondly, the energy industry must follow the government's policies and society's expectations. Recently, a big challenge has been the anti-nuclear protests which have been occurring in many developed nations worldwide. Nuclear power stations have become more and more unpopular [41]. This primarily derives from the two most serious nuclear and radioactive accidents, the Chernobyl disaster in 1986 and Fukushima disaster in 2011. These accidents have caused citizens to question the safety of nuclear power stations and worry about the inestimable loss caused by these accidents. This has also become a new pushing force for society to seek the availability of a greener and energy-saving source, such as wind and solar power. However, stable power production from wind farms is difficult to achieve because of the intermittent and inconsistent wind source. This uncertain nature of wind generation makes it difficult for a power supplier to plan a stable and secure power system.

Even though challenges persist, more and more research are being conducted on the topic of renewable energy in the power system operation plan, and it seems that wide utilization of renewable sources is near. Despite the fact that renewable sources may be free, such as wind power, it is still necessary to take generation costs into consideration due to the other costs required to maintain and operate the power generators.

Thirdly, after the conversion of primary energy into electricity, the next problem is how to transfer the energy from the power generation plants to other areas. Generally, this is classified as the transmission problem, whose only concern is to transfer large amounts of energy over long distances in a large region [61], e.g., transporting wind-generated power from wind farms in Scotland to north east England. In practice, it is inevitable to have power loss through the long distance transmission line. This power loss is mainly caused by the conductor heat loss [41], as every transmission line has a non-zero resistance. In order to decrease this unwanted and costly power loss, the general principle in transporting power is to use a high voltage transmission network. In fact, to send a fixed power, the higher the voltage, the lower the current is required in the transmission line. As a result, the resistive heating loss, which is proportional to the current squared, can be reduced to an acceptable order of magnitude. It is worth mentioning that some of the high voltage connections in the UK national grid can operate at 400 KV in practice [2].

Most important of all, the electrical power flow within the power system cannot be determined manually, since the amount and direction of a power flow must obey the Kirchhoff's laws [46]. These laws state a series of restrictions for the transmission grid, by linking voltage issue together with the power flow issue. They make the transmission problems much more difficult and imply that the meaning of a feasible transmission is not merely providing enough power to meet the demand.

Another critical issue in the transmission problem is the lack of large-scale energy storage facilities. This restricts the power generation plan, since no huge amount of energy can be saved for later use. Thus, the total amount of energy generated from plants must be equal to the total amount of the power consumed in the transmission network, including the loads in the destinations and the losses in the transmission lines.

An entire power system, at a national level, has a very large-scale transmission network. Moreover, the size of the power system may continue to expand due to the following reasons:

- with the rapid growth in electricity demand, additional generators and lines may be introduced.
- new transmission lines are being built into outlying areas with lower population, mainly because green energy is usually not produced close to the consumers.
- the use of local microgeneration is becoming more popular, as well as smart grid technologies.

Hence, the power system plan must be kept up-to-date, and most important of all, it must ensure a secure and stable power system for all the people living within this system.

Fourthly, there is the distribution problem, which is similar to the transmission problem. Its main task is to send power from its local distribution substations to individual end-users via distribution channels, which involves much smaller power flows over a short distance in local areas at a lower voltage. Additionally, while a transmission network is always interconnected, the distribution network is more likely to be built in a layered structure [61]. The power is transfered at a lower voltage in a radial way in the distribution system, from local supplier to each individual customer. Unlike the transmission system, the distribution system seldom changes its structure on a large scale, and the only significant change is the new customer connecting to the system.

Additionally, once a failure occurs, the distribution network is able to survive more easily by reconfiguring the power, while the transmission network requires more work to restore the operation to nominal level to prevent large-area effects. More details about the similarities and the differences between the transmission system and distribution system can be found in [61].

For better understanding, a summary of the power generation, transmission and distribution stages is shown schematically in Figure 1.1.

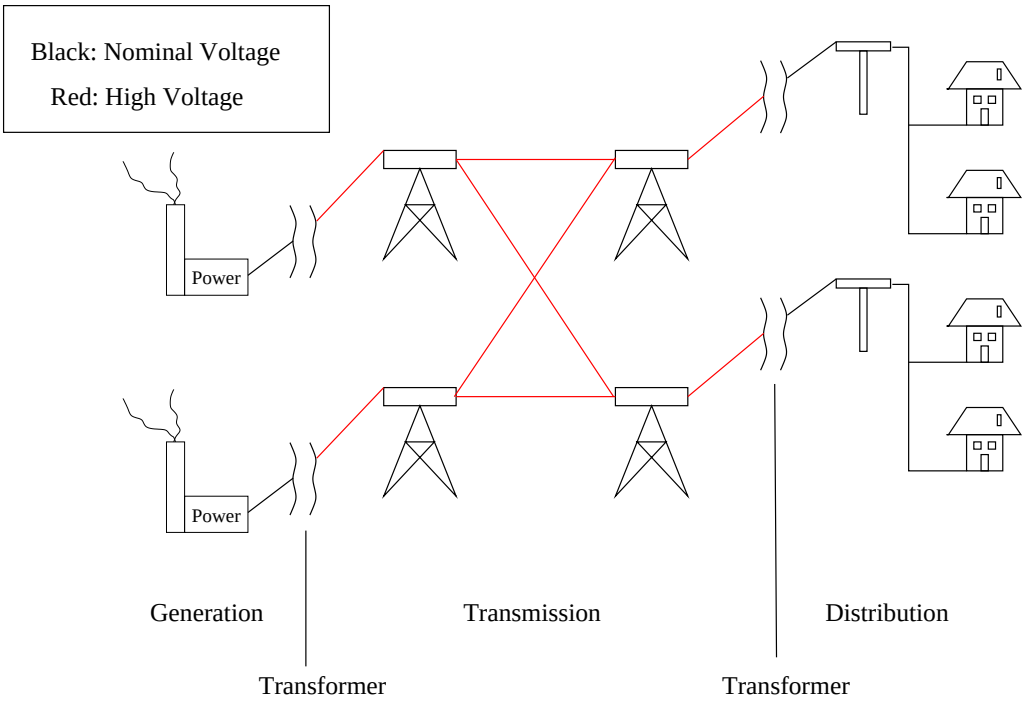

Figure 1.1: Illustration of the three major stages in the power industry.

On the other hand, the power industry is becoming more competitive than ever before due to market liberalization. Therefore, the economic issues of the power system operation should also be taken into consideration, and consequently, a series of optimization problems arise [46]. In fact, the power of optimization techniques is extensively applied to analyze, plan and operate

the power system in different kinds of questions [29]. For instance, the use of optimization tools can achieve either the minimum number of controls rescheduled, or minimum energy loss in the grid [64, 4].

One fundamental optimization application for the transmission is called the *economic dispatch* problem [46]. It determines an optimal operation plan for a group of electricity generators at the minimum generation cost, while satisfying the required loads and some operating limits such as the power generation limits and thermal flow limits. However, the economic dispatch problem does not include the Kirchhoff's laws as physical operating constraints, which restricts its application in practice. As an improvement on the economic dispatch problem, the *Optimal Power Flow (OPF)* problem was firstly announced by Carpentier in the early 1960s [20], which may be one of the best known optimization applications in the transmission problem. It successfully integrates the physical power flow restrictions and the economic issues behind the power system operation.

The *Security Constrained Optimal Power Flow (SCOPF)* problem is an extension of the OPF problem. As its name suggests, it contains some important features of reliability in the optimization model. It guarantees the whole power system can, not only work under the normal long-term base conditions, which are also used for the OPF problem, but also can survive without changing power generation plan when some predetermined contingencies occur, such as failures or outages of equipment in the power system. It is worth mentioning that there are several variants of the SCOPF models since they are quite flexible to accept customized constraints. Some of the SCOPF models may allow limited corrective actions after the accident occurs.

Additionally, there are many optimization solution methods applied to solve OPF/SCOPF problems [59], but among all these solution methods, the *interior point method (IPM)* has been the most popular and successful tool from the last decades [73]. More details about OPF and SCOPF will be given in the next chapter.

Interior point method is possibly the most important milestone in computational optimization, after the discovery of the simplex method. Both theoretical analysis and computational behavior have shown that this new algorithm has significant advantages in solving some very large problems, when compared with the simplex method. As a result, the research field of IPMs has expanded since the 1980s and there have been thousands of publications related to IPMs. It seems that this number will continue to climb for a long time to come, since IPM is still a hot topic nowadays. Despite the huge amount of related research, there still remain some interesting unsolved questions in IPM, e.g, how to pick a robust and efficient warm-start algorithm in IPM?

There are plenty of studies about IPM itself, which show that IPM has been successfully implemented to solve various optimization problems efficiently, not only for linear programming, but also for nonlinear nonconvex programming. Most of the main optimization packages, both

commercial products and research projects, have already accomplished an interior point kernel. Generally, each IPM iteration is much more expensive than a step in the simplex method. However, the computational effort is principally based on its linear algebra implementation behind the interior point framework. Generally, the interior point algorithm is able to solve a large and sparse problem much faster than a dense problem with the same dimensions.

Research also shows that structure-exploitation is strongly available in IPM. It can be done very efficiently for linear programming or quadratic programming, while it is a little bit harder for nonlinear programming. Additionally, the interior point implementation can be strongly parallelizable, while parallel simplex method is likely to be impossible. The parallel algorithm is mainly developed from exploiting the special structure of the problem [37, 57, 56], in which the full problem can be decomposed into several sub-problems. Note that this is always the case for stochastic programming and other well-structured large-scale problems such as SCOPF problems. Therefore, analysis of the problem structure may bring huge advantages into an IPM implementation.

Since both the power system problem and IPM are state-of-the-art research topics, many researchers are tempted to combine them. There are many interior point papers tagged with the power system problem and vice versa. Generally, interior point methods have proven to be robust and successful methods for solving various power system problems including OPF/SCOPF problems [72, 58, 18]. Note that there is a variety of way to solve OPF/SCOPF and this class of power system problems shows great interests in the advanced optimization tools. Although there is still some known challenges to OPF/SCOPF [52], it is still quite common to solve OPF/SCOPF problems by interior point algorithm. There are several different IPM implementations to solve OPF problems, i.e. interior point cutting plane method [48], multiple predictor-corrector IPM [75], quadratic IPM [18], and so on. For SCOPF problems, literature also shows that IPM can successfully achieve the goal [68, 49, 57]. More references about applying IPM to solve OPF/SCOPF problems can be found in [58, 72, 60] and the references therein.

Even though there are numerous publications tagged with IPM and OPF/SCOPF, none of them try to exploit the special structure of the linearized SCOPF problem, which will be presented in this thesis. Moreover, only one set of research conducted by Qiu et al [57] tries to solve the SCOPF with a structure-exploiting IPM. Their structure-exploiting technique follows the ideas presented in OOPS [37], which is a structure-exploiting IPM solver, and it has a preconditioning technique similar to the method mentioned in [56]. However, it does not provide a proper way to choose the preconditioner for the SCOPF problem and the bottleneck is the accuracy of its methodology. On the other hand, research from Petra and Anitescu [56] showed that the preconditioned method outperforms a direct approach in a structure-exploiting IPM on medium-sized problems but experiences numerical difficulties on larger problems. They use random scenarios to build the preconditioner and this scheme is not specialized for the SCOPF

problem.

The aims and the main contribution of this thesis are as follows:

- Some further advantages of solving the linearized SCOPF problem by exploiting their special structure are demonstrated and the corresponding numerical experiments are presented. As far as the author is aware, this is the first time that an IPM algorithm for solving linearized SCOPF is designed with the details of its structure.

- A preconditioned iterative method with a new methodology to pick up the preconditioner is applied to solve the Schur complement system arising in the IPM steps. Two different schemes are suggested to build up the preconditioner, based on how to integrate the contributions of the contingency scenarios into the preconditioner.

- Some sensible heuristic ideas are described to fill in the gap between the linearized SCOPF model and nonlinear SCOPF model, as well as the gap between the OPF model and SCOPF model. This work is used to find the necessary contingencies for operating the system before solving the SCOPF model. Thus the full SCOPF model, which contains all required contingencies, can be reduced to a smaller SCOPF problem and consequently an improvement in the efficiency of its solution process can be expected.

- Inspired by the warm-start techniques of IPM, an exact new IPM framework is designed to solve SCOPF problems. The most exciting aspect is that it can dynamically add the active contingencies into the SCOPF model and hence solve the problem by calling one IPM process. Some theoretical analysis of this algorithm is given for the linearized SCOPF model, while numerical experiments show that it works successfully for both linearized SCOPF problems and nonlinear SCOPF problems.

The later chapters of this thesis are organized as follows: Chapter 2 gives a review of all the fundamental background covered in this thesis as well as a further literature review for each subtopic. In Chapter 3, the structure of the linearized SCOPF model is demonstrated, and all the corresponding IPM implementations are presented with their numerical experiments. Chapter 4 gives more details about the contingency generation techniques which are utilized to reduce the set of contingency scenarios before solving the full nonlinear SCOPF model. Several advisable methods are presented and discussed to detect the most dangerous contingencies from the linearized OPF/SCOPF model or nonlinear OPF model. In Chapter 5, a new IPM framework for SCOPF problem is presented, which integrates the contingency generation techniques within an interior point linear algebra. Numerical experiments also show that it can solve both linear and nonlinear SCOPF problems efficiently, while it can detect a much smaller set of active contingencies. Chapter 6 describes the interface between OOPS and other software such as the mathematical modeling language AMPL. The last chapter will present a conclusion of this thesis and list some possible future work in this research area.

Chapter 2

Background

In this chapter, we introduce the problems and the methodologies considered in this thesis. Firstly, a brief description of power systems is given, followed by a description of OPF problems and SCOPF problems. Then, the main and basic methodologies used in this research are reviewed, such as the interior point methods and different methods for solving a linear system.

2.1 Power Transmission Problem

2.1.1 Power System and Notations

As presented in Figure 1.1, a full power system problem consists of three major problems. This thesis mainly pays attention to the transmission problem and the other two problems are simplified as local issues.

Generally, a power system transmission grid consists of buses (nodes) $b \in \mathcal{B}$, transmission lines $l \in \mathcal{L}$ and power generators $g \in \mathcal{G}$. Power generators are allocated at buses, i.e. for each generator g, $o_g \in \mathcal{B}$ gives the bus to which it is connected. Note that the real power generator buses are also known as PV buses, since the voltage level and real power generation at these buses are specified. Similarly, a distribution system can also be summarized and attached to a local bus. That is, all the regional loads in a distribution system can be added up as one big load, and then this summation is localized at each bus as a local demand in the transmission system.

In order to visualize our work for better understanding, a 3-bus system, which comes from an example in [46], is illustrated in Figure 2.1. There are four generators and three transmission lines in this sample system. Each arrow attached at a bus denotes the local demand, which is a summarization of the regional distribution system connected to this bus. Note that this 3-bus power system is being used as an example throughout this thesis.

Before we go to the explicit mathematical OPF/SCOPF model, first we list all the necessary

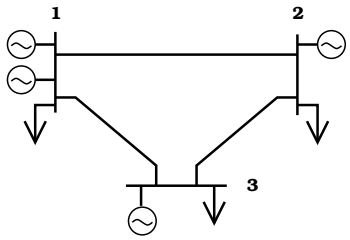

Figure 2.1: 3-bus system

notations without explanation. Further explanation will be given in the corresponding sections.

Sets:

- $\mathcal{B}$ Set of buses (nodes), indexed by b.
- $\mathcal{C}$ Set of contingencies, indexed by c.
- $\mathcal{G}$ Set of power generators, indexed by g.
- $\mathcal{L}$ Set of line connections, indexed by l. It can also be indexed by the pair (i, j), which denotes the connection that starts from bus i and ends at j. A line between bus i and bus j will have two different connections (i, j) and (j, i).
- $\mathcal{L}_i$ Set of all the line connections, one end bus of which is i.
- $\mathcal{T}$ Set of transformer connections (lines), indexed by t or pair (i, j). One end of the connection should be a 'tap bus'.
- $\mathcal{B}_{\mathcal{PV}}$ Set of PV buses, $\mathcal{B}_{\mathcal{PV}} \subset \mathcal{B}$.

Parameters:

- b_0 Reference bus.
- o_g Location of the power generator g. $o_g \in B$
- h_l Resistance of line l.
- r_l Reactance of line l.
- α_l Conductance of line l. $\alpha_l = \dfrac{h_l}{r_l^2 + h_l^2}$.
- β_l Susceptance of line l. $\beta_l = -\dfrac{r_l}{r_l^2 + h_l^2}$.
- γ_l Shunt capacitance of line l.
- d_b^P, d_b^Q Real and reactive power demand at bus b.
- f_l^+ Flow limit for line l in base case.

- $f_{c,l}^{+}$ Flow limit for line l in contingency c.
- $V_b^{\pm}$ Minimum and maximum voltage level at bus b in base case.
- $V_{c,b}^{\pm}$ Minimum and maximum voltage level at bus b in contingency c.
- $p_g^{\pm}$, $q_g^{\pm}$ Real and reactive power output limits for generator g.
- $\tau_t^{\pm}$ Tap ratio limits for transformer t.
- $\tau_{c,t}^{\pm}$ Tap ratio limits for transformer t in contingency c.
- c_{0g}, c_{1g}, c_{2g} Constant, linear and quadratic term of the power generation cost at g.

Variables:

- V_b Voltage level at bus b in base case.
- $V_{c,b}$ Voltage level at bus b in contingency c.
- δ_b Phase angle at bus b in base case.
- $\delta_{c,b}$ Phase angle at bus b in contingency c.
- p_g, q_g Real and reactive power output at generator g in base case.
- $p_{c,g}, q_{c,g}$ Real and reactive power output at generator g in contingency c.
- $f_{(i,j)}^{P}, f_{(i,j)}^{Q}$ Real and reactive power flow at connection (b_i, b_j).
- $f_{c,(i,j)}^{P}, f_{c,(i,j)}^{Q}$ Real and reactive power flow at connection (b_i, b_j) in contingency c.
- τ_t Tap ratio of transformer t in contingency c.

2.1.2 Optimal Power Flow

The optimal power flow (OPF) problem may be one of the most fundamental optimization problems in the power system. This is because its operation setting satisfies all the transmission and generation constraints and moves towards the optimization goal. There is no fixed objective function of this optimization problem and it can be specialized to satisfy the requirements case by case [4], One of the most typical and popular criteria is the minimum of the real power generating cost, and we use a quadratic generation cost function (2.1) as our major objective function in this thesis.

$$\sum_{g \in G} (c_{0g} + c_{1g} p_g + c_{2g} p_g^2) \tag{2.1}$$

An OPF model also requires a stable power system in a long term steady state. Hence, given $\delta_{ij} := \delta_i - \delta_j$, an OPF model should contain all following constraints:

- Power Generation Constraints:

$$p_g^- \leq p_g \leq p_g^+, \quad \forall g \in \mathcal{G} \tag{2.2}$$

$$q_g^- \leq q_g \leq q_g^+, \quad \forall g \in \mathcal{G} \tag{2.3}$$

- Voltage Level Constraints:

$$V_b^- \leq V_b \leq V_b^+, \quad \forall b \in \mathcal{B} \tag{2.4}$$

- Thermal Flow Constraints:

$$\left(f^P_{(i,j)}\right)^2 + \left(f^Q_{(i,j)}\right)^2 \leq \left(f^+_{(i,j)}\right)^2, \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{T}, \tag{2.5}$$

- Reference Bus Constraint:

$$\delta_{b_0} = 0, \tag{2.6}$$

- Kirchhoff's Current Law:

$$\sum_{g|o_g=b} p_g - d_b^P = \sum_{(i,j)\in\mathcal{L}_i} f^P_{(i,j)}, \quad \forall b \in \mathcal{B}, \tag{2.7}$$

$$\sum_{g|o_g=b} q_g - d_b^Q = \sum_{(i,j)\in\mathcal{L}_i} f^Q_{(i,j)} - \frac{(V_b)^2}{2} \sum_{(i,j)\in\mathcal{L}_i} \gamma_{(i,j)}, \quad \forall b \in \mathcal{B} \tag{2.8}$$

- Kirchhoff's Voltage Law:

$$f^P_{(i,j)} = \begin{cases} \alpha_l V_i^2 - V_i V_j[\alpha_l \cos(\delta_{ij}) + \beta_l \sin(\delta_{ij})], & (i,j) \in \mathcal{L} \\ \alpha_l (\frac{V_i}{\tau_t})^2 - \frac{V_i V_j}{\tau_t}[\alpha_t \cos(\delta_{ij}) + \beta_t \sin(\delta_{ij})], & (i,j) \in \mathcal{T}, \text{ bus } i \text{ tapped} \\ \alpha_l V_i^2 - \frac{V_i V_j}{\tau_t}[\alpha_t \cos(\delta_{ij}) + \beta_t \sin(\delta_{ij})], & (i,j) \in \mathcal{T}, \text{ bus } j \text{ tapped} \end{cases} \tag{2.9}$$

$$f^Q_{(i,j)} = \begin{cases} -\beta_l V_i^2 - V_i V_j[\alpha_l \sin(\delta_{ij}) - \beta_l \cos(\delta_{ij})], & \text{otherwise} \\ -\beta_l (\frac{V_i}{\tau_t})^2 - \frac{V_i V_j}{\tau_t}[\alpha_l \sin(\delta_{ij}) - \beta_l \cos(\delta_{ij})], & \text{bus } i \text{ tapped} \\ -\beta_l V_i^2 - \frac{V_i V_j}{\tau_t}[\alpha_l \sin(\delta_{ij}) - \beta_l \cos(\delta_{ij})], & \text{bus } j \text{ tapped} \end{cases} \tag{2.10}$$

The first two constraints are obvious, as a stable power system should take into account generation level constraints (2.2) and voltage constraints (2.4). The thermal flow constraints (2.5) give physical limits of the power flow consisting of the real and reactive power. Constraint (2.6) is called the reference bus constraint, which forces the phase angle at reference bus to be zero.

The last two constraints are typical physical equalities, which establish the conservation of energy in the power system. Kirchhoff's Current Law (KCL) (2.7)-(2.8) is applied to any node in the power system network. It asserts that the sum of incoming power flow and power generation at any particular intersection bus must be equal to the sum of outgoing power flow and power consumption at this bus. Kirchhoff's Voltage Law (KVL) states that the algebraic sum of voltage differences across any closed loop in the network must equal zero at any instant. This conservative property can be stated for the AC power system in complex form, which results in a simple and straightforward expression. Conversely, Kirchhoff's voltage law can be stated in hybrid form [29] as (2.9) and (2.10), for real and reactive power respectively. These sophisticated expressions are derived by applying Kirchhoff's voltage law as the complex Ohm's law. For more details about how to get these expressions and the complex Ohm's law please refer to [29].

Although the OPF model is able to work effectively and obtain a minimal operating cost solution for a long term stable system, it is not necessarily secure against equipment failure. Once a failure occurs, the OPF operation solution could become unable to avert a loss of electric power due to a drop in voltage in a power generator. Additionally, the most severe case of a power outage is called a blackout, which indicates a total loss of power in an area. For instance, if one transmission line is blocked off accidentally, the power flow on this blocked line needs to find another way to move to its destination which may finally culminate in overheating the other transmission lines. If there is no remedial measures applied quickly enough to restore the power system to the nominal level, a domino effect may occur as a series of lines are triggered. As a result, the entire power system may fail very quickly such that unpredictable large-scale blackouts may occur, consequently leading to an enormous loss. Therefore, in order to guarantee a stable power system, power system planners must take care of several important reliability features in the OPF model.

Last but not least, all the variables of an OPF model can be classified into two catalogues:

- Control variables: As the name suggests, these variables can be controlled by a power system operator, e.g. bus voltage and real power generation at PV buses. These variables are the major concern in operating the power system, since they can characteristic the behavior of the system. It is also known as 'decision variable'.

- State variables: These variables reflect the current states of a power system network. They can be determined after all the control variables have been given. Reactive power

generation and all the flow variables fall under this catalogue.

Generally, the degree of freedom for a optimization problem is equal to the number of independent constraints distracted by the number of variables. This is the minimum number of independent variables which can characteristic the behavior of the entire system. In OPF model, the number of the control variables are equal to the degree of freedom in a power transmission network.

One obvious problem associated with the operation of a power system is the load flow problem. That is, with the given control variables, this problem tries to solve its corresponding state variables and therewith define the current states of a power system.

2.1.3 Security Constrained Optimal Power Flow

The idea of steady-state security is firstly published by Alsac and Stott [8]. It suggests adding some outage-contingency constraints into the OPF formulation to make the model reliable. In the later publication [64], Alsac, Stott and Monticelli gave a classification of power system static security levels, and showed that the use of contingency-constrained OPF model is to keep the power system in the secure or a correctively secure state. This contingency-constrained OPF problem is then widely known as the Security Constrained Optimal Power Flow (SCOPF) problem in the later publications.

As an improvement of the OPF problem, the SCOPF problem seeks a long-term operation plan under the base-case condition, while the whole transmission network can successfully keep working under any predetermined emergency contingencies caused by the loss of segments of the network, such as losing transmission lines or power generators.

In this thesis, only line failures are taken into account. Therefore, a given set $\mathcal{C}$ defining all the predetermined contingencies, $c \in \mathcal{C}$, can be the index of the tripped transmission line and consequently $\mathcal{C} \subseteq \mathcal{L}$. The security model uses the same objective function as previously in (2.1) and it should contain all following constraints explicitly:

- Reactive Power generation Constraints, $\forall c \in \mathcal{C}$:

$$q^{-}_{c,g} \leq q_{c,g} \leq q^{+}_{c,g}, \quad \forall g \in \mathcal{G} \tag{2.11}$$

- Voltage Level Constraints, $\forall c \in \mathcal{C}$:

$$V^{-}_{c,b} \leq V_{c,b} \leq V^{+}_{c,b}, \quad \forall b \in \mathcal{B} \tag{2.12}$$

- Thermal Flow Constraints, $\forall c \in \mathcal{C}$:

$$\left(f^{P}_{c,(i,j)}\right)^2 + \left(f^{Q}_{c,(i,j)}\right)^2 \leq \left(f^{+}_{c,(i,j)}\right)^2, \quad \forall (i,j) \in \mathcal{L}, \tag{2.13}$$

- Reference Bus Constraint, $\forall c \in \mathcal{C}$:

$$\delta_{c,b_0} = 0, \tag{2.14}$$

- Kirchhoff's Current Law, $\forall c \in \mathcal{C}$:

$$\sum_{g|o_g=b} p_{c,g} - d_b^P = \sum_{(i,j)\in\mathcal{L}_i\setminus\{c\}} f_{c,(i,j)}^P, \quad \forall b \in \mathcal{B}, \tag{2.15}$$

$$\sum_{g|o_g=b} q_{c,g} - d_b^Q = \sum_{(i,j)\in\mathcal{L}_i\setminus\{c\}} f_{c,(i,j)}^Q - \frac{(V_b)^2}{2} \sum_{(i,j)\in\mathcal{L}_i\setminus\{c\}} \gamma_{(i,j)}, \quad \forall b \in \mathcal{B} \tag{2.16}$$

- Kirchhoff's Voltage Law, $\forall c \in \mathcal{C}$:

$$f_{c,(i,j)}^P = \begin{cases} \alpha_l V_{c,i}^2 - V_{c,i}V_{c,j}[\alpha_l \cos(\delta_{c,ij}) + \beta_l \sin(\delta_{c,ij})], & \text{otherwise} \\ \alpha_l (\frac{V_{c,i}}{\tau_t})^2 - \frac{V_{c,i}V_{c,j}}{\tau_t}[\alpha_t \cos(\delta_{c,ij}) + \beta_t \sin(\delta_{c,ij})], & i \text{ tapped} \\ \alpha_l V_{c,i}^2 - \frac{V_{c,i}V_{c,j}}{\tau_t}[\alpha_t \cos(\delta_{c,ij}) + \beta_t \sin(\delta_{c,ij})], & j \text{ tapped} \end{cases} \tag{2.17}$$

$$f_{c,(i,j)}^Q = \begin{cases} -\beta_l V_{c,i}^2 - V_{c,i}V_{c,j}[\alpha_l \sin(\delta_{c,ij}) - \beta_l \cos(\delta_{c,ij})], & \text{otherwise} \\ -\beta_l (\frac{V_{c,i}}{\tau_t})^2 - \frac{V_{c,i}V_{c,j}}{\tau_t}[\alpha_l \sin(\delta_{c,ij}) - \beta_l \cos(\delta_{c,ij})], & i \text{ tapped} \\ -\beta_l V_{c,i}^2 - \frac{V_{c,i}V_{c,j}}{\tau_t}[\alpha_l \sin(\delta_{c,ij}) - \beta_l \cos(\delta_{c,ij})], & j \text{ tapped} \end{cases} \tag{2.18}$$

- Controllable voltage variables, also known as PV bus constraints, $\forall c \in \mathcal{C}$:

$$V_{c,b} = V_b, \qquad \forall b \in \mathcal{B}_{\mathcal{PV}} \tag{2.19}$$

$$p_{c,g} = p_g, \qquad \forall g \in \{g|o_g \in \mathcal{B}_{\mathcal{PV}} \setminus \{b_0\}\} \tag{2.20}$$

Constraints (2.11)-(2.18) are similar to the constraints (2.3)-(2.10) of the OPF model. For each line outage contingency c, one reduced power system network can be defined as missing line c. Constraints (2.12)-(2.18) are used to ensure every reduced transmission network is feasible. Additional attention needs to be paid to the last constraints (2.19) and (2.20). These two constraints declare that the voltage level and real power generation on the PV bus should remain the same as the base-case plan, regardless to which contingency they refer. The only exception is the reference bus as its real power generation can be modified to refill the power transmission loss occurring in each contingency. In other words, only changes of the real power generation at the reference bus are allowed for correction actions after a failure occurs.

The security criteria used in the SCOPF model is commonly expressed as "n-x", where 'x' denotes the type of contingencies [1]. Note that this notation can have different meanings in

different countries. For instance, in the definitions given by the National Grid [1]: 'n-1' refers to losing any one of the items of the power system; 'n-2' refers to the simultaneous loss of any two items; 'n-d' refers to the loss of two circuits in a double circuit line. However, the 'n-1' criteria in ENTSO-E standard [3] can also include the loss of a double circuit on the same pylon.

For clarity, the 'n-x' criteria in the following content uses the first definition. More specifically, the main concern in this thesis is the SCOPF problem with n-1 line outage criteria, which means that the loss of any single transmission line should not lead to any other line exceeding its flow limit. Note that this one-line-outage criteria is commonly treated as the contingency case in both research and development literature [4].

In the real world, there still exists the inevitability of two or more lines or generators failing within a short period. These occurrences can be covered by using a model with higher security criteria, e.g. the n-2 model and so on, to guard against such incidents. Obviously, a full n-2 model can easily become infeasible since it requires at least three transmission lines being connected to each bus in the grid. In practice, the compromise is to use the n-1 SCOPF as the least security requirement, with varying degrees of security throughout the power system to ensure robustness. For instance, several significant line outages can be included as n-2, or maybe higher, security constraints, to ensure that these critical lines are not lost. The GB system is operated to n-d criteria (on double circuit lines) on its main transmission system but it is acceptable to relax the security criteria to n-1 in good weather [5].

Generally, the higher the security requirement applied to a SCOPF problem, the more contingencies are required to be added into the mathematical model, which will make the problem significantly larger. In fact, the typically very large size is the major downside of the full SCOPF model. The dimension of the problem is necessarily much higher than for the traditional OPF model, since all the predetermined contingencies must be included. This characteristic makes the SCOPF problem difficult to solve, especially for large systems and when many contingencies are considered in the model. For instance, if a SCOPF problem consists of a power system with m transmission lines and is assessed against the n-1 security criteria, the size of its full n-1 SCOPF model is roughly $m + 1$ times that of the corresponding OPF model.

Depending on the size of the power system, trying to solve the SCOPF problem with too many binding contingencies inevitably requires prohibitive computational efforts, such as memory usage and solution time. However, most contingencies do not constrain the optimal solution of the SCOPF problem. Only very few of the contingencies are required to be added into the SCOPF model [64]. From a mathematical point of view, that means only a certain number of contingencies contain active boundary constraints at the optimal solution, while other contingencies only contain inactive boundary constraints. Hence, if these 'inactive' contingencies are omitted from the SCOPF model, the solution is guaranteed to be unchanged. By adding fewer contingencies into the SCOPF problem, the size of the problem can be reduced significantly,

decreasing the complexity of its numerical behavior.

In practice, solving the reduced security model several times and adding contingency constraints iteratively mitigates the need to solve a large nonlinear SCOPF problem. This method consists of a trial-and-error framework which is used to detect violated constraints and then to update the reduced SCOPF model. This process keeps solving the revised model until no constraint violation can be found. The idea of iterative solution techniques for SCOPF was firstly suggested by Alsac and Stott in [8].

Generally, this iterative solution technique can be demonstrated schematically in the flow chart of Figure 2.2.

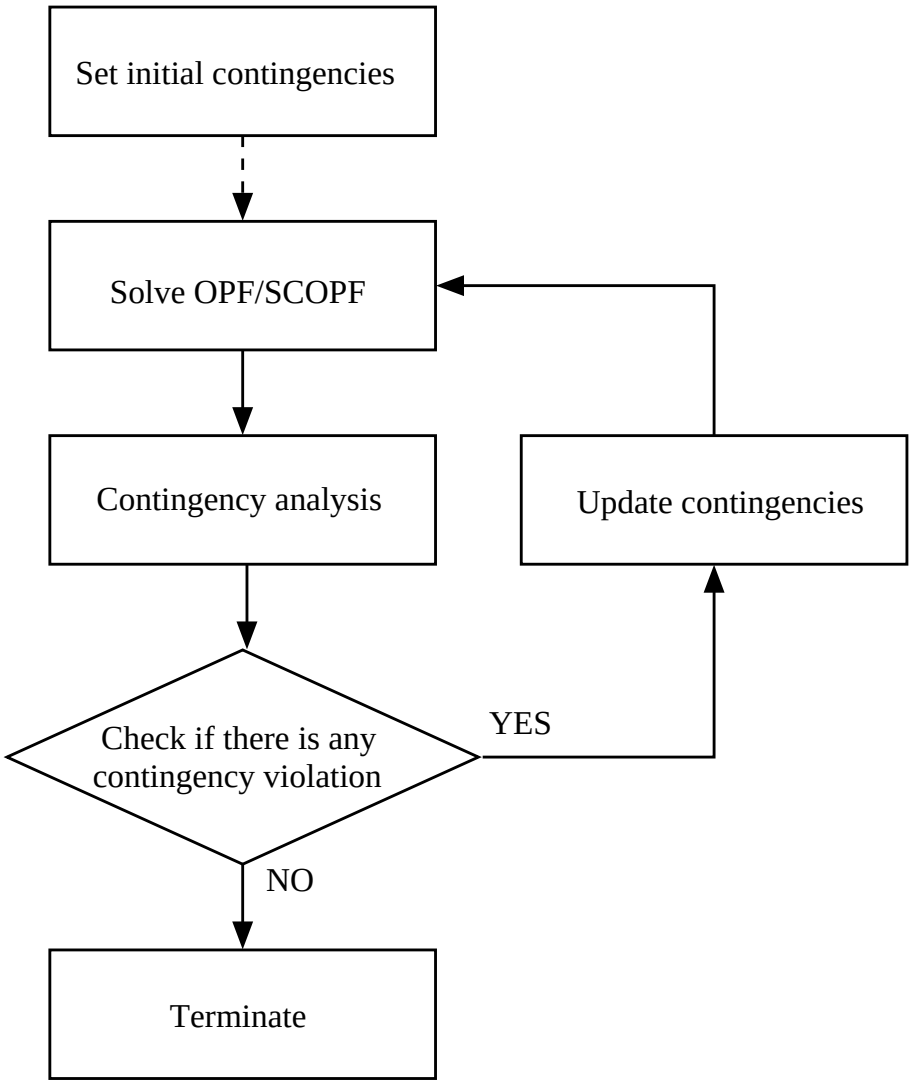

Figure 2.2: Flow chart for solving SCOPF

The dotted line in Figure 2.2 is an optional step. If no initial contingency is given, the algorithm starts from solving the OPF problem as the base-case condition; otherwise it can use the given contingency scenarios as well as the base case to build the initial SCOPF model. After solving this system, we can get a base-case operating plan for the power system. Then,

some contingency analysis must be performed to check whether or not all the other constraints given by the required contingencies (which have not yet been included in the SCOPF model) are feasible under the given base-case operating plan. This step can be done by running some simulation techniques or by solving a local load flow problem explicitly for each contingency. If there are no constraint violations in any of the contingencies, that means the current SCOPF model already contains all the necessary contingencies to find the solution to the SCOPF problem. Conversely, if contingency violations are detected, the constraints corresponding to the detected contingencies should be added into the SCOPF model and the algorithm should be restarted to solve the revised model.

This trial-and-error solution technique is also referred to as the 'outer process' in [4]. This outer process has a major computational influence on finding a successful SCOPF solution. That is, one main disadvantage of this iterative method is that it may require constructing and solving the SCOPF model several times before finding the final solution. If several large SCOPF models have to be constructed and solved, the computational time could be excessive and it may be even worse than the solution time required by solving the SCOPF model with full contingencies directly.

If all such dominant contingencies can be obtained beforehand by making a good prediction of the initial contingencies before start the solution process, or if they can be obtained very quickly during the outer process by some methodology for analyzing unused contingencies, the computational efforts can be decreased dramatically. In fact, this outer process has been widely studied and many techniques have been developed to make it more efficient, based on different filtering techniques to detect binding contingencies in the reduced model [68, 27, 50, 17, 19].

Last but not least, the SCOPF model presented above exhibits good structure. It is built with several sub-problems defined by each contingency, which is similar to a typical stochastic programming problem. However, the above SCOPF model does not contain any probabilities and random variables, and consequently all the SCOPF problems mentioned in this thesis are deterministic programs and only support finding a feasible operation plan for each contingency case. More specifically, all above SCOPF frameworks use a minimum long-term cost function as their objective and do not involve any optimization of the reduced power systems in the contingency cases. The above SCOPF model does not consider the cost of additive power generation at the reference bus in each contingency, which is used to cover the energy loss in the transmission network. Even though the SCOPF model can ensure that the full power system can survive under any predetermined contingency case, in practice, once a failure occurs, a power system planner must analyze the cause rapidly, and try to restore the power system to the nominal operation level as soon as possible.

2.1.4 DC Formulations

The formulation presented in the previous two sections demonstrates that the standard model of a SCOPF problem is nonlinear and nonconvex. Compared to linear programming, nonlinear programming problems are generally much more difficult to solve. In practice, in order to have a quick understanding of the OPF/SCOPF problem, the use of a linearized model is widely accepted. The

In this section, for reason of simplicity, only the linearized OPF model will be derived. The linearized SCOPF model can be easily extended from OPF by the same methodology presented in the previous section, so that the linearized SCOPF model is given at the end of this section without further discussion.

Firstly, the approximated objective function becomes

$$\sum_{g \in G} c_g p_g, \tag{2.21}$$

which is a linear cost function. Then, special attention should be paid to how to linearize the nonlinear constraints (2.5)-(2.10).

The linearization is derived under the following assumptions:

- Resistance of the transmission lines can be neglected since they are much smaller than the reactance.

- The voltage level at each bus is equal to the nominal level V. This assumption comes from the very tight bounds of the voltage level at each bus, e.g. the perturbation is required to be less than 6 percent of the nominal level V in our OPF model. This assumption indicates that the tap ratio is equal to one and the voltage level can become a constant parameter in the linearized model and hence the voltage level variables can be ignored.

- The phase angle difference between each two adjacent buses is small enough. Hence the first order Taylor approximation at point zero is sensible, which implies the following approximations:

$$\sin(\delta_{ij}) \approx \delta_{ij} \equiv \delta_i - \delta_j, \cos(\delta_i - \delta_j) \approx 1.$$

From the first assumption and definition of conductance, it is trivial that in this case $\alpha_l = \frac{h_l}{r_l^2 + h_l^2} = 0$ and $\beta_l = -\frac{r_l}{r_l^2 + h_l^2} = -\frac{1}{r_l}$. Therefore, the Kirchoff's voltage law (2.9)-(2.10)

indicates that the real and reactive power flows can be simplified as

$$f^P_{(i,j)} = 0 - V^2[0 - \frac{1}{r_l}\delta_{ij}] = \frac{V^2}{r_l}\delta_{ij}, \tag{2.22}$$

$$f^Q_{(i,j)} = -\beta_l V^2 - V^2[0 - \beta_l * 1] = 0. \tag{2.23}$$

This resulting linear constraint demonstrates there is no reactive power flow in the linearized model. As a result, all the constraints corresponding to the reactive power now can be eliminated from the model.

On the other hand, a high voltage transmission system is adopted to decrease the power loss by reducing the current in the lines. Hence, it is quite reasonable to ignore the energy loss in high voltage transmission systems. In fact, this consequence is hidden within the above assumptions: as described in the previous chapter, the main energy loss throughout the power transmission system is the thermal loss due to the resistance in the lines. Since the first assumption states that resistance now can be ignored and there is no reactive power in the linearized approximation, it implies that there is no real power loss in the transmission system. As a result, there is no need to care about the different connections between any two buses, i.e. connections (b, b') and (b', b) between bus b and b'. Bearing in mind that there is no real power in the linearized model, the real power flow $f^P_{(i,j)}$ now can be simplified as f_l for each transmission line l. Last but not least, since there is no loss in the transmission system, the sum of real power generation should equal the sum of the loads throughout the grid.

Let parameters a_{bl} denote the bus-line (node-arc) incidence for the power network, where $a_{bl} = -1$ if b is the start bus of line l; 1 if b is the end bus of line l; otherwise takes 0. Then it is trivial to obtain the linearized constraints for the OPF model as follows:

- Linearized Thermal Flow Constraints:

$$-f_l^+ \leq f_l \leq f_l^+, \quad \forall l \in \mathcal{L} \cup \mathcal{T} \tag{2.24}$$

- Linearized Kirchhoff's Current Law:

$$\sum_{g|o_g=b} p_g + \sum_{l \in \mathcal{L}\cup\mathcal{T}} a_{bl} f_l = d_b, \quad \forall b \in \mathcal{B}. \tag{2.25}$$

- Linearized Kirchhoff's Voltage Law:

$$f_l = -\frac{V^2}{r_l} \sum_{b \in \mathcal{B}} a_{bl}\delta_b, \quad \forall l \in \mathcal{L} \cup \mathcal{T}. \tag{2.26}$$

Give $\mathbf{H} \in \mathcal{R}^{|\mathcal{B}|\times|\mathcal{L}\cup\mathcal{T}|} = \{a_{bl}\}$ the bus-line (node-arc) incidence matrix for the power network and let $\mathbf{J}$ denotes the bus-generator incidence matrix for the power network, where its element

equals 1 if a particular generator is attached to this line bus; takes 0 elsewhere. The equations of (2.25) and (2.26) for the whole system can be represented in matrix form as the following (2.27) and (2.28), respectively.

$$\mathbf{J}p + \mathbf{H}f = d, \tag{2.27}$$

$$f = -\frac{V^2}{r}\mathbf{H}^\top \delta. \tag{2.28}$$

Similarly, the security constraints (2.11)-(2.17) can also be easily linearized by the above scheme. The linearized Kirchhoff's laws for security constraints can also be stated in matrix form as:

$$\mathbf{J}p + \mathbf{H}_c f_c = d, \quad \forall c \in \mathcal{C}, \tag{2.29}$$

$$f_c = -\frac{V^2}{r}\mathbf{H}_c^\top \delta_c, \quad \forall c \in \mathcal{C}, \tag{2.30}$$

where $\mathbf{H}_c \in \mathcal{R}^{|\mathcal{B}|\times|\mathcal{L}\cup\mathcal{T}|}$ gives the bus-line incidence matrix for the contingency c. However, in order to imply the contingency constraint $f_c = 0$, there is a dummy column in $\mathbf{H}_c$ with all zero entries. The bus-generator incidence matrix $\mathbf{J}$ stays the same for all the contingencies since no generator outage is allowed in the model. Obviously, an individual generator can only be allocated at one particular bus, hence there is only one non-zero element in each column of $\mathbf{J}$.

It is worth emphasizing again that this linearized model only contains real power flow and this linearization approach is a classical way to simplify the AC model [65]. To distinguish the linearized model from the standard AC model, it is referred to as the "DC" model in the remainder of this thesis.

2.1.5 Discussion

From the mathematical formulation of its constraints (2.2)-(2.10), it is obvious that only the first three constraints are inequality constraints. In addition, the first two, the power generation constraints (2.2) and voltage level constraints (2.4), are representative boundary constraints of the corresponding variables. After replacing the inequality constraint (2.13) with an equality constraint by adding slacks, the AC SCOPF model can be formulated as follows:

$$\begin{aligned} \min: \quad & f(x) \\ \text{s.t:} \quad & g(x) = 0 \\ & x \geq 0 \end{aligned} \tag{2.31}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are nonlinear functions. Additionally, the second derivatives of both these functions exist and they are twice continuously differentiable. Note that the DC formulation can also be written in this form, since the linearized thermal constraint (2.24) can be referred as the boundaries of a real power flow variable.

To clarify and simplify the definition, notation (2.31) is also used as the standard form of nonlinear programming problems in this thesis. It is possible to use other forms to describe the nonlinear model, but, no matter which form is applied, it can be easily reformulated into the standard form (2.31) by adding slack variables or doing simple substitutions. From a mathematical point of view, it is obvious that for the same NLP model, all the different reformulations should be equivalent to each other, i.e. the solution of the models should be the same. Unfortunately, this is not true in practice, since different formulations may have different numerical behavior and some may cause numerical difficulties. This issue is quite common in computational optimization. A similar question is discussed in the later section as to which expression should be used to solve Newton's system. In our implementation, the problem of how to convert inequality constraints to equality constraints requires further discussion, as presented in Chapter 4.

Corresponding to the standard form (2.31), the *Lagrangian* function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ for the original problem (2.31) can be defined as:

$$L(x, y, z) := f(x) - g(x)^\top y - z^\top x \tag{2.32}$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are defined as the *Lagrangian multipliers*, corresponding to the equality constraints and the inequality constraints, respectively.

Prior to the advent of the interior point methods (which are considered in the next section), the first approach used to solve the SCOPF problem was presented in Alsac and Stott's publication [8]. It consists of a penalty function approach for nonlinear problems.

The outline of their penalty function approach is summarized as follows:

- Introduce a penalty function for the inequality constraints.
- Relax all the equality constraints ONLY by using the Lagrangian of its penalty function. As a result, the third term in the right-hand-side of the corresponding Lagrangian function (2.32) does not exist anymore.
- Find the stationary point for this Lagrangian function as the optimal solution to the SCOPF problem.
- Use an initial guess of the control variables in the base case to find the solutions for the state variables and Lagrangian multipliers in sequence.

- Apply Newton's method to solve for the equality constraints to find iterates of the control variables.
- Use the control variables to detect inequality constraint violations.
- Add penalties to the violated constraints.
- Adjust the control variables using a gradient method which consists of projections from other contingencies.

From the above description, it is obvious that this original method for solving the SCOPF problem only adds violated constraints into the security model as opposed to the whole contingency, which is slightly different from the approach presented in the diagram in Figure 2.2.

This class of solution approaches is called the decomposed approach [7, 64], which focuses on the optimization in the base-case space. The control variables are optimized, while each violated contingency constraint is added into the model via linearization. This linearization is applied by means of a projection into the space of control variables in the base case. That is, the projected constraint only includes the base-case control variables. This projection can be achieved by the use of large-perturbation sensitivity analysis [64].

For a DC formulation, this decomposed approach can be very efficient, since the imposed constraints are linear and all the coefficients are fixed. As a result, these additive constraints can be valid throughout the solution process. This decomposition method is also known as the Benders decomposition while each additive inequality constraint is a Benders cut [53, 4]. For the AC case, this decomposed approach can also be applied with Benders cuts. However, these Benders cuts, which come in a linear form, must be recalculated at every iteration since now the model is nonlinear. These cuts may introduce new linearization error in the nonlinear model and consequently may increase the number of iteration in the outer process [64]. Additionally, the contingency analysis step must be performed in a non-decomposed manner in order to check whether or not all the contingency constraints are satisfied. Note that in the real world, the decomposed approach is able to provide a good practical solution, and hence most industrial codes have been implemented with decomposed approach since the announce of SCOPF in 1970s [8].

On the other hand, the 'non-decomposed approach' [64] aims to explicitly solve the SCOPF model with all the contingency constraints. The non-decomposed approach may provide a more accurate solution since there is no linearization error caused by the Benders cuts, which means this approach solves a proper AC-SCOPF problem with nonlinear constraints. The downside of a non-decomposed approach is that it may suffer from requiring solution of very large NLPs. However, for a non-decomposed approach, the iterative solution technique can also be applied to solving the SCOPF problem from the basic case, while some other techniques are designed to reduced the number of additive contingencies, e.g. the filtering techniques mentioned previously.

As a result, adding an entire contingency into the reduced SCOPF model is also quite common in literature. Most of all, Qiu et al [57] also show that a non-decomposed approach, even for the nonlinear AC case, is parallelizable with the use of interior point methods.

In this thesis, we focus on the non-decomposed approach [64], while we bear in mind that the iterative solution techniques are efficient and useful in practice. In Chapter 4, we will present a specialized IPM method which incorporates the iterative SCOPF solution techniques with the iterative process in IPM. In the following section, a primal-dual interior point method is introduced. It can be applied as a standard optimization solver to solve all the reduced SCOPF models appearing in the iterative solution techniques. It is well-known that IPMs can be successful in efficiently and robustly solving nonlinear problems.

2.2 Primal-Dual Interior Point Methods

The first interior point method is generally acknowledged as Karmarkar's polynomial algorithm in 1984, which is considered as a primal method. Karmarkar's work attracted optimizers' interest since it is a polynomial-time algorithm, which should be much faster than the simplex method for the large-scale problem.

After more than two decades of research on these methods, there are several variations of the interior point methods. Among these interior point methods, the primal-dual methods have been unanimously accepted as the most successful ones, both in terms of theoretical analysis and computational behavior [74].

In this section, we present the fundamental framework of the primal-dual interior point methods and their related linear algebra.

Firstly, the *Karush-Kuhn-Tucker (KKT)* conditions [74, 14, 54], also known as the first-order optimality conditions, can be stated in the following theorems:

Theorem 2.1 (Karush-Kuhn-Tucker conditions (NLP))**.** *Let x be a local solution of* (2.31)*. If the Jacobian of constraints $\bigtriangledown g(x)$ has full column rank, then there must exist a pair (y, z), such that the following conditions hold:*

$$\bigtriangledown f(x) - \bigtriangledown g(x)y - z = 0 \tag{2.33}$$

$$g(x) = 0 \tag{2.34}$$

$$\boldsymbol{XZ}e = 0 \tag{2.35}$$

$$x, z \geq 0 \tag{2.36}$$

where

$$\boldsymbol{X} = \mathrm{diag}(x_1, x_2, \ldots, x_n), \quad \boldsymbol{Z} = \mathrm{diag}(z_1, z_2, \ldots, z_n),$$

and e is the vector of ones, i.e. $e = (1, \ldots, 1)^\top$ in appropriate dimension.

These results can be applied to both linear and nonlinear cases. However, we can obtain a much clearer and straightforward formulation if we are dealing with linear programming. Since the main spirit of interior point framework can be applied to both linear and nonlinear programming, for better understanding and clarity, we only state the idea of interior point method for linear programming in the rest of this section. The distinctive features of nonlinear IPM implementation will be covered in Chapter 4.

2.2.1 Linear Algebra in Linear Programming

Firstly, we consider the linear program with the following form:

$$\begin{aligned} (LP:) \quad \min \quad & c^\top x \\ \text{s.t.} \quad & \mathbf{A}x = b \\ & x \geq 0 \end{aligned} \tag{2.37}$$

where $x, c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$. Note that matrix $\mathbf{A}$ is commonly assumed full row rank, meaning the rows of the matrix are linearly independent. This problem (2.37) is called the *primal* problem for linear programming, or referred to as the *LP* problem for short.

Duality Theory

Before we present a key result in this section, firstly we need to introduce the Lagrangian function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ for the LP problem (2.37):

$$L(x, y, z) := c^\top x - y^\top (\mathbf{A}x - b) - z^\top x = -b^\top y + (c - \mathbf{A}^\top y - z)^\top x \tag{2.38}$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are defined as the *Lagrangian multipliers*. Then, a *Lagrangian dual function* $L_D : \times\mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ can be defined as

$$L_D(y, z) := \inf_x \; L(x, y, z) = -b^\top y + \inf_x \; \{(c - \mathbf{A}^\top y - z)^\top x\} \tag{2.39}$$

Observe that the second term in (2.39) is bounded below if and only if the 'linear coefficient' $c - \mathbf{A}^\top y - z$ is exactly equal to zero. That is:

$$L_D(y, z) = \begin{cases} -b^\top y, & \text{if } \mathbf{A}^\top y + z = c \\ -\infty, & \text{otherwise} \end{cases} \tag{2.40}$$

For each pair (y, z) with $z > 0$, we can have $L_D(y, z) = \inf_x \; L(x, y, z) \leq L(x^*, y, z) \leq c^\top x^*$, where x^* is the optimal solution of the original problem (2.37). The Lagrangian dual function $L_D(y, z)$ gives a lower bound of the optimal objective $c^\top x$. Hence, the next step is to define

a so-called *dual problem* to find which pair of (y, z) can generate the tightest bound for its LP problem. The dual problem for the LP problem (2.37) is stated as:

$$\begin{aligned} (LD:) \quad \max \quad & b^\top y \\ \text{s.t.} \quad & \mathbf{A}^\top y + z = c \\ & z \geq 0 \end{aligned} \tag{2.41}$$

Note that for each LP problem (2.37), its associated dual problem (2.41) is also a linear program, and most important of all, its dual problem is unique. Hence it is quite natural that the Lagrangian multipliers y and z are also known as the *dual variables*. Similarly, problem (2.41) is also known as the *LD* problem for simplicity.

These two problems, LP and LD, are always considered as a primal-dual pair of a linear program, since they are strongly related to each other. Given $\Omega_P := \{x \in \mathbb{R}^n | \mathbf{A}x = b, x \geq 0\}$ and $\Omega_D := \{(y, z) \in \mathbb{R}^m \times \mathbb{R}^n | \mathbf{A}^\top y + z = c, z \geq 0\}$, for any LP feasible solution $x \in \Omega_P$ and LD feasible solution $(y, z) \in \Omega_D$, the duality theory, as described above, guarantees that the following inequality is always true

$$c^\top x \geq b^\top y. \tag{2.42}$$

This property is called *Weak Duality*. It shows that the dual objective $b^\top y$ provides a lower bound for the primal problem, while the LP problem provides an upper bound for its LD problem. A *Dual Gap* is defined as the difference $c^\top x - b^\top y$.

In addition, we have so-called *Strong Duality* for the primal-dual pair, defined as:

$$\min_{x \in \Omega_P} c^\top x = \max_{y \in \Omega_D} b^\top y. \tag{2.43}$$

That is, if both LP and LD are feasible and x^* and y^* are the optimal solutions for LP and LD respectively, then $c^\top x^* = b^\top y^*$. Furthermore, if the strong duality holds, then there is no dual gap between LP and LD. This property is important as it is widely used as termination criteria for solving linear programming problems by IPM.

KKT Conditions

In order to solve the LP problem (2.37), firstly the KKT conditions for linear programming is derived from Theorem 2.1.

Corollary 2.1 (Karush-Kuhn-Tucker conditions (LP))**.** *Let x be an optimal solution of* (2.37)*. Then there must exist a pair (y, z), which is the optimal solution of* (2.41)*. In addition, the*

following conditions hold:

$$\begin{aligned} c - \boldsymbol{A}^\top y - z &= 0, && \text{(2.44a)} \\ \boldsymbol{A}x - b &= 0, && \text{(2.44b)} \\ \boldsymbol{XZ}e &= 0 && \text{(2.44c)} \\ x, z &\geq 0 && \text{(2.44d)} \end{aligned}$$

where

$$\boldsymbol{X} = \operatorname{diag}(x_1, x_2, \ldots, x_n), \quad \boldsymbol{Z} = \operatorname{diag}(z_1, z_2, \ldots, z_n), \quad e = (1, \ldots, 1).$$

It is trivial to observe that the first two equality equations (2.44a) and (2.44b) are exactly the same as the constraints in the dual problem (2.41) and primal problem (2.37), respectively. Hence, they can be referred to as primal and dual feasibility constraints. The last nonlinear equality equation (2.44c) in the KKT conditions is widely known as the *complementarity constraint/condition*. It indicates that for all $i = 1, \ldots, n$, there is no such pair of (x_i, z_i) that both variables x and z are positive, and at least one of x_i and z_i has to be zero. Briefly put, the KKT conditions are necessary conditions which indicate the characteristic of the solution to (2.37).

2.2.2 Interior Point Method for Linear Programming

Now we are in place to consider the linear programming problem LP together with its dual problem LD. The primal-dual *feasible set* $\mathcal{F}$ is defined as

$$\mathcal{F} := \{(x, y, z) \mid \mathbf{A}x = b, \mathbf{A}^\top y + z = c, (x, z) \geq 0\}. \tag{2.45}$$

However, most interior point methods require the iterations to be strictly feasible, hence the origin of its name *interior point*. The *strictly feasible set* $\mathcal{F}^0$ is defined as

$$\mathcal{F}^0 := \{(x, y, z) \mid \mathbf{A}x = b, \mathbf{A}^\top y + z = c, (x, z) > 0\}. \tag{2.46}$$

Central Path

The usual transformation in IPM is to eliminate the inequality constraints in (2.37), by replacing them with a logarithmic barrier term.

$$\begin{aligned} (LP_\mu :) \quad \min \quad & \varphi(x) := c^\top x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} \quad & \mathbf{A}x = b \end{aligned} \tag{2.47}$$

where $\mu > 0$ is a barrier parameter.

After applying the KKT conditions (Theorem 2.1), the optimal solution (x^*, y^*, z^*) should satisfy the set of equations:

$$\begin{aligned} \mathbf{A}x &= b, && \text{(2.48a)} \\ \mathbf{A}^\top y + z &= c, && \text{(2.48b)} \\ \mathbf{XZ}e &= \mu e, && \text{(2.48c)} \\ x, z &\geq 0. && \text{(2.48d)} \end{aligned}$$

The conditions (2.48) can be treated as an approximation of the optimality condition (2.44) for the original LP problem (2.37). The only difference between these two KKT conditions is the third equality condition (2.48c). It overrides the original complementarity condition (2.44c), as it requires all the pairwise products $x_i z_i$, $i = 1, \ldots, n$, to have the same positive value μ.

When μ is considered changeable, we can define a set $\mathcal{C}$ to record all the solutions of the perturbed KKT conditions (2.48). It is called the *central path* and its notation is given as

$$\mathcal{C} = \{(x_\mu, y_\mu, z_\mu) \in \mathcal{F}^0 \mid \mu > 0\}. \tag{2.49}$$

One of the most important properties of the central path for LP is that for each $\mu > 0$, the solution to the corresponding KKT system (2.48) is unique if and only if the strictly feasible set $\mathcal{F}^0$ is not empty. The proof of this essential property can be found in [74]. This powerful property, as well as the property of continuity and connectedness of the central path, can guarantee that the solutions to the sequence of barrier problems (2.47) can converge to a primal-dual solution to the original LP problem (2.37), at exactly the same time as μ is decreasing to zero.

Long-Step Path-Following Methods

For the reason of simplicity, a function $F_\mu : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m}$ is defined as

$$F_\mu(x, y, z) = \begin{bmatrix} \mathbf{A}x - b \\ \mathbf{A}^\top y + z - c \\ \mathbf{XZ}e - \mu e \end{bmatrix}. \tag{2.50}$$

The KKT conditions (2.48) can be replaced by the following system

$$\begin{aligned} F_\mu(x, y, z) &= 0, && \text{(2.51a)} \\ x, s &> 0 && \text{(2.51b)} \end{aligned}$$

To ensure convergence, the generic framework of interior point methods is to apply Newton's

method to solve a sequence of nonlinear systems (2.51a), while it forces a reduction of the barrier parameter μ between steps. Particularly, for each barrier parameter μ, the Newton direction $(\Delta x, \Delta y, \Delta z)$ is obtained by solving the set of linear equations:

$$J_\mu(x, y, z) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = -F_\mu(x, y, z) \tag{2.52}$$

where matrix J is the Jacobian of F_μ. As each KKT system (2.51a) is solved by Newton's method towards a point on the central path $\mathcal{C}$, this class of interior point methods is classified as the *path-following methods*.

However, the long-step path-following methods may not generate a point exactly on the central path $\mathcal{C}$, or even not close to $\mathcal{C}$, due to the following reasons:

- Since the KKT system is a nonlinear system and Newton's method is a linear approximation method at current point, taking a full Newton's step cannot ensure condition (2.48c) being satisfied exactly, i.e. $(x_i + \Delta x_i)(z_i + \Delta z_i) \neq \mu$.
- A full Newton's step is usually rejected, since the positive condition (2.51b) would be violated once a full step is accepted.

In general, long-step path-following methods would generate a sequence of points, which stay within a 'good' neighborhood of $\mathcal{C}$. Given constant $\gamma \in (0, 1)$, the most popular neighborhoods used in theoretical analysis for a long-step method is one of the following two:

- ∞-Neighborhood

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, y, z) \in \mathcal{F}^0 \mid XZe \geq \gamma\mu e\}. \tag{2.53}$$

- Symmetric Neighborhood

$$\mathcal{N}_s(\gamma) = \{(x, y, z) \in \mathcal{F}^0 \mid \gamma\mu e \leq XZe \leq \frac{1}{\gamma}\mu e\}. \tag{2.54}$$

An instance of the central path and its neighborhood is visualized in Figure 2.3.

Different path-following methods, based on different choices of the neighborhood, would select a proper reduction parameter σ and adjust the step size $\alpha \in (0, 1]$ to guarantee all the iterates stay in the chosen neighborhood. The next iterate is then defined as

$$(x^+, y^+, z^+) = (x, y, z) + \alpha(\Delta x, \Delta y, \Delta z); \tag{2.55}$$

where $(\Delta x, \Delta y, \Delta z)$ is the solution of the Newton equation system for F_μ.

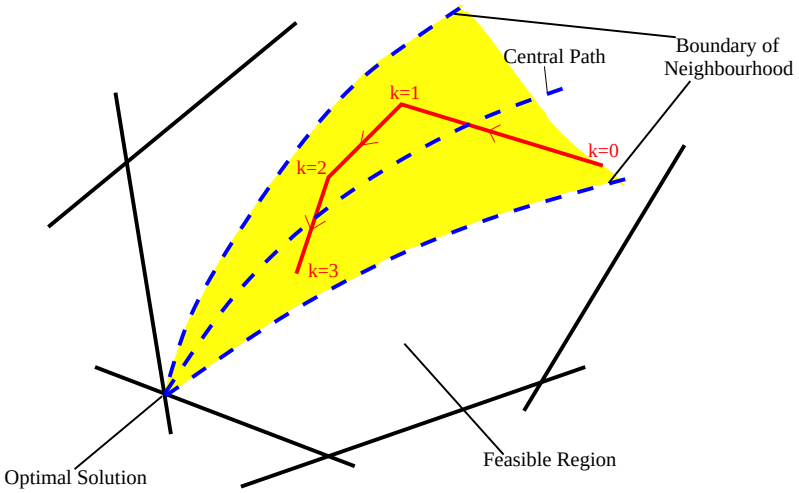

Figure 2.3: An example of a central path and its neighborhood

The theory of IPM ensures this path-following IPM algorithm can converge, meaning the solutions to the barrier problem (2.47) approach the solution of the original problem as $\mu \to 0$.

Most of all, theoretical analysis also gives us the complexity analysis of the IPM algorithm. According to [74], the long-step path-following IPM for LP can converge in $\mathcal{O}(n \ln(1/tol))$ steps, where *tol* is the required tolerance. However, literature [34] also states that IPM performs much better than the theoretical analysis in practice.

Some Equivalents of the KKT System

Recalling the Newton equation system (2.50) and the notation of the LP problem (2.37), we can write down the *full KKT system* for (2.37) as follows:

$$\begin{bmatrix} \mathbf{A} & 0 & 0 \\ 0 & \mathbf{A}^\top & I \\ \mathbf{Z} & 0 & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix}, \tag{2.56}$$

where

$$\begin{aligned} \xi_p &= b - \mathbf{A}x \\ \xi_d &= c - \mathbf{A}^\top y - z \\ \xi_\mu &= \sigma\mu e - \mathbf{X}\mathbf{Z}e \end{aligned} \tag{2.57}$$

where $\sigma \in [0, 1]$ is a reduction parameter of μ. It has also been known as the centering parameter in [74].

Note that system (2.56) is also referred to as the 3×3 *KKT system*. The size of this system is $(2n + m) \times (2n + m)$ and the matrix involved is nonsymmetric.

From the last row of (2.56), we can observe the following equivalent substitution

$$\Delta z = \mathbf{X}^{-1}(\xi_\mu - \mathbf{Z}\Delta x). \tag{2.58}$$

Therefore, the full system (2.56) can be reduced to a 2×2 *augmented system* after eliminating Δz by (2.58). The augmented system is derived as

$$\begin{bmatrix} -\Theta & \mathbf{A}^\top \\ \mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}. \tag{2.59}$$

where $\Theta := \mathbf{X}^{-1}\mathbf{Z}$. This augmented system is a $(n+m)\times(n+m)$ system. One main advantages of this system is that the augmented system matrix $\begin{bmatrix} -\Theta & \mathbf{A}^\top \\ \mathbf{A} & 0 \end{bmatrix}$ is symmetric and indefinite.

Furthermore, the augmented system can be further reduced to the 1×1 system by substituting $\Delta x = \Theta(\mathbf{A}^\top \Delta y - (\xi_d - X^{-1}\xi_\mu))$. The 1×1 system is called *normal equation system*, obtained as

$$\mathbf{A}\Theta^{-1}\mathbf{A}^\top \Delta y = \mathbf{A}\Theta^{-1}(\xi_d - X^{-1}\xi_\mu) + \xi_p. \tag{2.60}$$

The normal equation system has dimension $m \times m$, and matrix $\mathbf{A}\Theta^{-1}\mathbf{A}^\top$ is symmetric and positive definite.

The main advantage of the augmented systems (2.59) is that the matrix involved is symmetric and usually sparse. Nevertheless, one of the well-known challenges is that the diagonal matrix $-\mathbf{X}^{-1}\mathbf{Z}$ may be ill-conditioned. The reason for this numerical difficulty is the following: as far as the IPM algorithm converges, the complementarity condition is approaching to be satisfied. Consequently, one of the iterates, x_i and z_i, is gradually closing to zero. This may bring very large or very small entries into the diagonal elements $x_i^{-1}z_i$ and causes trouble numerically. One of the remedies to achieving a stable factorization of the augmented system is the use of a regularization technique, which is stated in the later section about regularization.

Conversely, the major advantage of the normal equation (2.60) is that it is positive definite. Hence it is eligible to apply Cholesky decomposition with confidence. One weakness of this normal equation is that it sometimes needs to accept a dramatical loss of sparsity, since matrix $\mathbf{A}\mathbf{X}\mathbf{Z}^{-1}\mathbf{A}^\top$ may be very dense due to a dense column in $\mathbf{A}$. This is also inefficient to apply Cholesky decomposition in the large-scale problems. Another disadvantage, which is the most important one relating to the structure-exploiting IPM algorithm, which will be covered in a later chapter, is that it may destroy the problem structure and becomes difficult for parallelization of the nonlinear problems.

In general, it is difficult to make a fair comparison between augmented system (2.59) and normal equation (2.60), since each has its own pros and cons. In practice, many modern IPM packages can also be categorized by this specification: CPLEX[76] and XPRESS[77] use the nor-

mal equation formulation (2.60). OOQP[78], LOQO[66, 12, 67], KNITRO[16, 15], IPOPT[70] and OOPS[37, 79] use augmented system (2.59). HOPDM[33, 9] has the flexibility to use both systems.

From the point of view of supporting better parallelization properties and keeping the problem specific structure, which is very important for the linearized SCOPF model, we use the augmented system formulation in all of our IPM algorithms and implementations. The principal framework of a primal-dual interior point method in augmented system form is stated in Algorithm 2.1.

Algorithm 2.1: Primal-Dual Interior Point Method (Augmented System)

Step 1. Initialization:
Given an initial point (x_0, y_0, z_0), where $x_0 > 0$ and $z_0 > 0$.
Set initial barrier parameter $\mu_0 > 0$ and convergence tolerance $tol > 0$.

Step 2. Start IPM Loop:
Do:
Solve augmented system (2.59) at current point (x^k, y^k, z^k).
Compute Δz^k by (2.58).
Choose step size α_k so that $(x_k + \alpha_k \Delta x^k, z^k + \alpha_k \Delta z^k) > 0$.
Set new iterate as:
$(x^{k+1}, y^{k+1}, z^{k+1}) = (x^k + \alpha^k \Delta x^k, y^k + \alpha^k \Delta y^k, z^k + \alpha^k \Delta z^k)$
Compute new residual $\xi = (\xi_p, \xi_d, \xi_\mu)$ by (2.57).
Reduce μ and set $k = k + 1$.
Until $\|\xi\|_\infty \leq tol$

At the end of this section, it is worth mentioning that there is no existing mature IPM solver, which is designed to solve the unreduced full system (2.56). The current state of the art shows the full system (2.56) is symmetrizable [21] and hence it is able to apply the same sparse symmetric linear system solver as the one used for augmented system. Most importantly, [21] also shows an interesting and exciting result as the largest eigenvalue of the full system is far smaller than those of 2×2 or 1×1 systems. More details about the sparse symmetric linear system solver are described in Chapter 6.

2.3 Solving a Linear System

In this section, several methods used to solve the linear systems arising in IPM are presented. Firstly, to simplify the notation, we use the form of

$$\mathcal{H}u = s \tag{2.61}$$

to denote the linear system, i.e. the augmented system (2.59). A common assumption is given as $\mathcal{H}$ is non-singular. Note that we also assume that $\mathcal{H}$ is symmetric, since an IPM algorithm

mainly focuses on the symmetric matrix arising in the system (2.59).

This section starts from the Cholesky factorization method, which is a direct method, and then followed by the iterative methods, which are approximation methods.

2.3.1 Cholesky Factorization

One of the most popular methods to solve a linear system is the use of Cholesky factorization. The basic Cholesky factorization algorithm decomposes a positive definite matrix $\mathbf{A}$ into a product of lower triangular matrix $\mathbf{L}$ with its transpose, such as

$$\mathcal{H} = \mathbf{L}\mathbf{L}^\top. \tag{2.62}$$

A variant of the Cholesky factorization is the $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorization as

$$\mathcal{H} = \mathbf{L}\mathbf{D}\mathbf{L}^\top = \begin{pmatrix} 1_{11} & & \\ \vdots & \ddots & \\ l_{n1} & \dots & 1_{nn} \end{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \begin{pmatrix} 1_{11} & \dots & l_{1n} \\ & \ddots & \vdots \\ & & 1_{nn} \end{pmatrix} \tag{2.63}$$

where matrix $\mathbf{D}$ is a diagonal matrix. The $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorization is more widely used than the $\mathbf{L}\mathbf{L}^\top$ factorization, since it can also be applied to certain symmetric indefinite matrices. In fact, if matrix $\mathcal{H}$ is positive definite, all the elements of matrix $\mathbf{D}$ are positive numbers; otherwise it can have negative values. With Cholesky factorization, the solution of the linear system $\mathcal{H}u = s$ can be obtained from the following sequence:

$$\begin{aligned} \dot{u} &= \mathbf{L}^{-1}s, \\ \ddot{u} &= \mathbf{D}^{-1}\dot{u}, \\ u &= \mathbf{L}^{-\top}\ddot{u}, \end{aligned} \tag{2.64}$$

Despite the fact that this factorization may be applied to most of invertible symmetric matrices, it may fail or become unstable because of a zero pivot or an extremely small pivot, to which no careful pivoting technique is applied.

In order to overcome this difficulty, one common remedy is to perform the $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorization with 2×2 matrix pivots. Similar to the expression in (2.63), the $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorization with 2×2 block pivots can be presented as (2.65). where each element of the matrices is a block matrix. More specifically, each diagonal element of the matrices $\mathbf{D}$ and $\mathbf{L}$ is now a 1×1 or 2×2 submatrix, while each non-diagonal part of $\mathbf{L}$ can have more choices as a 1×2 or 2×1

block matrix.

$$\mathcal{H} = \mathbf{L}\mathbf{D}\mathbf{L}^\top = \begin{pmatrix} \mathbf{I}_{11} & & \\ \vdots & \ddots & \\ \mathbf{L}_{n1} & \dots & \mathbf{I}_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{D}_1 & & \\ & \ddots & \\ & & \mathbf{D}_n \end{pmatrix} \begin{pmatrix} \mathbf{I}_{11} & \dots & \mathbf{L}_{1n} \\ & \ddots & \vdots \\ & & \mathbf{I}_{nn} \end{pmatrix} \tag{2.65}$$

On the other hand, a symmetric indefinite matrix $\mathcal{H}$ is *quasi-definite* [66], if it has the following form:

$$\mathcal{H} = \begin{bmatrix} -\mathbf{E} & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{F} \end{bmatrix}, \tag{2.66}$$

where $\mathbf{E}$ and $\mathbf{F}$ are positive definite matrices. Despite such a matrix still being indefinite, it is eligible for a $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ Cholesky factorization without block pivots. Most important of all, the quasi-definite matrix is strongly factorizable, i.e. whatever permutation of rows and columns is applied, it can still be factorized without block pivots.

As a result, there are two ways to $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorize the symmetric indefinite matrix in the augmented system formulation, $\begin{bmatrix} -\Theta & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$. The first method is to factorize this indefinite matrix with block pivots. The second method is applying $\mathbf{L}\mathbf{D}\mathbf{L}^\top$ factorization with regularization techniques. Such regularization techniques can slightly modify the indefinite augmented system by adding a perturbation to the diagonal term [9]. The corresponding regularized augmented system matrix can be demonstrated as follows:

$$\begin{bmatrix} -\Theta - \mathbf{R}_p & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{R}_d \end{bmatrix} \tag{2.67}$$

where perturbation $\mathbf{R}_p$ and $\mathbf{R}_d$ are the regularization terms for the primal part and dual part, respectively. Both $\mathbf{R}_p$ and $\mathbf{R}_d$ are diagonal matrices with only positive elements. This outcome matrix (2.67) is quasi-definite and as a result, it can be factorized without 2×2 pivots.

Ordering Techniques

When Cholesky factorization is applied to a sparse matrix, it is worth highlighting the sparsity of the matrix, especially in a large-scale case. This is because the Cholesky factorization for a sparse matrix usually needs to introduce nonzero fill-in during its process. To illustrate this process for better understanding, a small matrix is used as an example in Figure 2.4. On the left hand side of Figure 2.4, a 3×3 matrix is presented in which the black squares indicate the nonzero elements and zero elsewhere. If the $(1, 1)$ element, which is highlighted in the red circle, is picked to be the first pivot to perform the Cholesky factorization, after one step of Cholesky factorization, namely by Gaussian elimination, the middle product before the lower

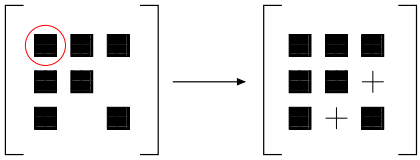

Figure 2.4: A 3×3 matrix and its Cholesky factorization

triangular factor $\mathbf{L}$ is shown in the right part of Figure 2.4, where "+" denotes a nonzero fill-in occurrence during the factorization. This means the factorization method can add nonzero into its outcome factor $\mathbf{L}$.

Conversely, Gaussian elimination may introduce less nonzero fill-in if rows and columns of a matrix can be reordered beforehand. For example, switching the first row/column and third row/column of this 3×3 matrix and starting Gaussian elimination from the updated $(1, 1)$ element, as shown on the left hand side of Figure 2.5, the Cholesky factorization would not add new nonzero fill-in into the outcome, which is shown on the right hand side of Figure 2.5.

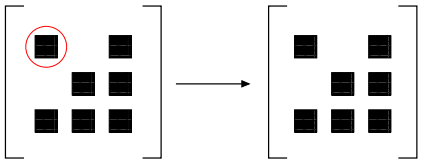

Figure 2.5: A 3×3 matrix and its Cholesky factorization (After ordering)

It is worth mentioning that each fill-in requires additional storage, and also additional arithmetic operations in the next step of Gaussian elimination as well as in the corresponding back-solve sequence that are used to recover the solution. Hence, they may significantly increase the computational effort if a bad ordering of pivots is used in a large-scale problem.

One common heuristic algorithm used to permute the rows and columns of the sparse matrix is named *minimum degree ordering*. It picks up the diagonal pivot in such a way that the selected pivot always has fewest nonzeros in its corresponding row and column. In practice, this algorithm is not computationally expensive and its performance is quite good [74]. Therefore, it is more efficient to analyze the sparsity for a problem and perform ordering techniques before starting the factorization. Another popular heuristic algorithm is called *nested dissection*, which can give a bound of the number of fill-in by a graph partition theory. Note that nested dissection method is used by code Metis [44].

Generally, in order to achieve better numerical performance, it is common to use a *symbolic factorization* to analyze the nonzero structure of the problem, as described above, while the exact values of the non-zeros can be ignored. This symbolic factorization only needs to be called once through the entire process since the nonzero structure of a particular problem is fixed. The use of symbolic factorization can lead to enormous computational saving in the later *numerical factorization*, which really factorizes the matrix $\mathcal{H}$ with its numerical entries.

Conclusion

Generally, when we solve an augmented system arising in the IPM process, the use of $\mathbf{LDL}^\top$ factorization with 2×2 pivots can be stable with a good ordering. However, the major disadvantage of this method is that it is difficult to predict the factorization process. It is because the 2×2 block pivots are only used when a 1×1 pivot meets a numerical difficulty. Thus, each step of factorization depends on the current data, which implies that the places of block matrices in the Cholesky factors $\mathbf{L}$ and $\mathbf{D}$ may vary between iterations in the IPM framework. Additionally, the use of a symbolic factorization beforehand is meaningless once a 2×2 block pivot is involved.

On the contrary, if regularization technique is applied firstly or dynamically during the numerical factorization, the greatest advantage is that the modified matrix (2.67) is factorizable without 2×2 pivots. By avoiding the need of using block pivots, we can achieve improved efficiency due to the symbolic factorization and easier access to the triangular matrix $\mathbf{L}$ and diagonal matrix $\mathbf{D}$. Furthermore, with a reasonable choice of $\mathbf{R}_p$ the ill-conditioning of the augmented matrix in a certain extent can be remedied. That is, by adding the primal regularization into the diagonal term $\mathbf{X}^{-1}\mathbf{Z}$, the diagonal pivots can be well perturbed with the similar order of magnitude. Hence the pivots can be guaranteed to be sufficiently bounded away from zero and resolve the numerical problem caused by ill-conditioning. Hence the regularizations not only make the system $\mathbf{LDL}^\top$ factorizable without block pivots, but also play a key role in maintaining the stability of the solver. Last but not least, the disadvantage of regularization techniques is that it requires modifying the augmented system. Hence the perturbation can never be too large, otherwise the problem can be totally changed.

In practice, these two methods are widely implemented. Algorithms and software for solving the sparse Cholesky factorization with block pivots are highly developed and widely tested. One of the most popular sparse Cholesky codes is MA57, which is further discussed in Chapter 6. To the author's knowledge, there is no such widely accepted standard code for factorization with regularization technique. Nonlinear solvers LOQO[66] and IPOPT [70] use different algorithms to compute the elements of the positive definite diagonal matrices $\mathbf{R}_p$ and $\mathbf{R}_d$, but their approaches are similar as they add perturbations to make augmented system matrices quasi-definite. In [66], Vanderbei computes the regularization terms before producing the factorization. On the other hand, Altman and Gondzio's regularization technique [9] is a dynamic method, which obtains the entries of regularization terms corresponding to its pivot during the numerical factorization. This method may be the most efficient regularization technique for linear and quadratic optimization. In our implementation, we consider both regularization techniques in our nonlinear IPM approach, which is discussed in the later chapters.

The default approach in this thesis uses a Cholesky factorization with dynamic regularization to solve the resulting linear system in IPM. Compared with other methods, this approach is

more stable and robust. However, the computational effort of Cholesky factorization may become very expensive if it is applied to a large and dense matrix. Most of all, it requires an explicit representation of the matrix that should be factorized, while in some cases, it is very difficult and expensive to get such an explicit form of a matrix. An example of this kind of matrix is shown in the later chapter about the structure-exploiting interior point method. In order to overcome this difficulty, a preconditioned iterative method may be applied to solve this troublesome case.

2.3.2 Iterative Methods

Generally, an iterative method solves the linear system (2.61) by evaluating a sequence of approximate solution with increasing accuracy. The major advantage of an iterative method is that it may only require matrix-vector products, which are relatively cheap in the large-scale sparse system. There are lots of iterative methods which are designed to solve different types of linear systems (2.61) case by case, e.g. depending on whether or not the matrix $\mathcal{H}$ is positive definite.

There is a class of iterative methods named as Krylov subspace methods, which may be the most successful iterative methods to solve large linear systems currently[62]. All Krylov subspace methods are projection methods and they need to build up a sequence of Krylov subspaces

$$K_j(\mathcal{H}, r_{init}) = span\{r_{init}, \mathcal{H}r_{init}, \mathcal{H}^2 r_{init}, \ldots, \mathcal{H}^{j-1} r_{init}\},\ j \geq 1 \tag{2.68}$$

where $r_{init} = s - \mathcal{H}u_{init}$ is the initial residual at the given initial guess of solution u_{init}.

A Krylov subspace method tries to find a sequence of approximate solutions u_j, $j = 1, \ldots, n$ from their own affine subspaces $u_{init} + K_j(\mathcal{H}, r_{init})$, respectively. The approximation u_j keeps improving with the increasing dimension j of the spanned subspace, and the iterative algorithm terminates once the termination criteria is satisfied. Common termination criteria require that the residual at current approximation should be sufficiently small. That is, $r_j = \|s - \mathcal{H}u_j\| \leq tol$, where tol is a given tolerance. However, if current approximation u_j cannot meet this specific requirement, the iterative method would start another search on the subspace with one more dimension, i.e. find approximation solution u_{j+1} from the subspace $u_{init} + K_{j+1}(\mathcal{H}, u_{init})$ where the dimension of the updated Krylov subspace $K_{j+1}(\mathcal{H}, r_{init})$ is $j + 1$. In general, the larger the Krylov subspace spanned, the better an approximation of a solution can be obtained by the iterative process. Obviously, Krylov subspace methods should converge in no more than n iterations, since they form a basis of $\mathcal{H}$ with up to n dimensions.

The major difference between variants of Krylov subspace method is how they construct the basis of Krylov subspace, i.e. how one algorithm finds a set of orthogonal vectors to span the Krylov subspace.

The most well-known iterative method from Krylov subspace families is the conjugate gra-

dient (CG) method, but it only supports the symmetric and positive definite linear system. As our focal point is the augmented system (2.59), which is a symmetric and indefinite matrix, CG is not eligible anymore and an alternative is required to solve this linear system. Possible approaches include the use of Generalized Minimal Residual Method (GMRES) or BiConjugate Gradient Method (BiCG), as both of these Krylov subspace methods are eligible to solve the non-symmetric indefinite linear system.

The biconjugate gradient method relies on a biorthogonality sequence instead of an orthogonality sequence. This means that it needs one extra basis to build the subspace

$$K_j(\mathcal{H}^\top, w_{init}) = span\{w_{init}, \mathcal{H}^\top w_{init}, (\mathcal{H}^\top)^2 w_{init}, \ldots, (\mathcal{H}^\top)^{j-1} w_{init}\}, \quad j \geq 1, \tag{2.69}$$

and find $w_j \in K_j(\mathcal{H}^\top, w_{init})$ and $r_j \in K_j(\mathcal{H}, r_{init})$ to satisfy the biorthogonality condition

$$r_i^\top w_j = 0, \quad \forall i \neq j.$$

On the other hand, GMRES, as its name suggests, is oriented to approximating the solution with minimal residual in each Krylov subspace. That is, in each step, GMRES needs to solve a least square problem

$$\min_{u_j \in u_{init} + K_j(\mathcal{H}, r_{init})} \|s - \mathcal{H} u_j\|_2, \tag{2.70}$$

to get the updated approximate solution u_j.

Comparing these two methods, the most attractive advantage of BiCG is that it does not need to store previous vectors and its computational complexity is much less than GMRES [10]. However, as BiCG does not minimize a residual, its convergence may be unpredictable. In fact, several papers state that the biconjugate gradient method is numerically unstable [45, 10, 11]. Note that references also suggest replacing BiCG by BiCGStab, which is a biconjugate gradient stabilized method. However, we would like to leave it as a future work and only focus on the performances of BiCG and GMRES. Unlike BiCG, the biggest disadvantage of GMRES is that it requires a record of all previous information and consequently increases the demand for storage significantly. Moreover, it is much more complicated than BiCG as it needs more work per iteration [10]. However, since it minimizes the residual incurred in the process, it is proposed as one of the most robust Krylov subspace methods to solve the non-symmetric linear system [11]. Considering all these pros and cons, and a quick numerical experiment of both iterative methods, we finally choose GMRES as our iterative method.

GMRES

The first GMRES algorithm was developed by Saad and Schultz [63] in 1986, which gives a basic framework of GMRES and a variant called restarted GMRES. The restarted GMRES is one variation of GMRES, in order to reduce the memory requirements. The idea is to restart the basic GMRES algorithm from current approximation of x, after every m steps, where m is a given integer parameter. The basic GMRES algorithm and restarted GMRES algorithm (GMRES(m)) are presented in the Appendix.

We implemented the modified GMRES(m) framework suggested by [11], which is a variation of the GMRES(m) method with QR factorization techniques. This QR factorization is applied to factorize the matrices arising in solving the least square problem (2.70). The use of QR factorization can transfer the involved matrix into an upper triangular formulation, and hence the solution to the least square problem (2.70) can be obtained by back-solving the triangular matrix and its corresponding right hand side. For more details about the QR factorization techniques in GMRES, please see [62, 11]. It is worth mentioning that we do not follow the way in [11] to do the QR factorization by Gram-Schmidt orthogonalization. Instead, as suggested in another version of GMRES presented in [62], Given rotations are applied to perform the QR factorization in our implementation. The full algorithm is stated in Algorithm 2.2. Note that this outcome of strategies is most similar to the GMRES implementation in Matlab [51].

As mentioned before, the restarted GMRES algorithm is guaranteed to converge in up to n steps, where n is the dimension of the linear system. Unfortunately, this may not be the case in practice due to the accumulation of numerical errors which may prevent the convergence of the method. Most importantly, compared to a direct method, another weakness of iterative methods is the problem of accuracy. The iterative method of GMRES may stagnate because no improvement can be achieved, and as a result, the algorithm fails to converge and terminate [11]. One possible remedy to overcome these difficulties is the use of a preconditioning technique. The preconditioning technique would transform the original linear system into a new one with the same solution, e.g. the solution of linear system $\mathbf{M}^{-1}\mathcal{H}u = \mathbf{M}^{-1}s$ is equivalent to the solution of (2.61), where nonsingular matrix $\mathbf{M}$ is called a preconditioner to this problem.

Generally, a good preconditioner $\mathbf{M}$ must have following properties:

- Linear system $\mathbf{M}^{-1}u = s$ is easy to solve.
- Matrix $\mathbf{M}^{-1}\mathcal{H}$ is close to identity matrix $\mathbf{I}$.
- Matrix $\mathbf{M}^{-1}\mathcal{H}$ is well-conditioned. Its condition number should be smaller than that of the original matrix $\mathcal{H}$.

If a good preconditioner is applied to solve the problem, it can

- Typically reduce the conditioning number of the problem. Hence the numerical behavior should be more stable.

- Reduce the number of steps in the iterative methods, which means less storage is required in GMRES algorithm and the efficiency is improved.

Therefore, both efficiency and robustness can be improved by the use of a good preconditioner.

A preconditioned version of GMRES method is equivalent to applying corresponding GMRES method to solve the preconditioned linear system

$$\mathbf{M}^{-1}\mathcal{H}u = \mathbf{M}^{-1}s \tag{2.71}$$

For the reason of simplicity, we do not state the corresponding preconditioned variant of Algorithm 2.2, since it can be easily derived from the algorithm and (2.71). More details about the iterative methods and preconditioning techniques, including their convergence theorem, can be found in [71, 62, 45] and their citations.

Last but not least, the major concern of using an iterative method to solve the linear system arising in the interior point method is to avoid the computationally expensive solutions from the direct method.

However, it means that the solution to the Newton system of equations (2.52) is evaluated approximately. It results in an inexact Newton direction for current IPM step. Hence, this inexact Newton step brings a series of questions, such as whether or not this inexact step is allowed to be taken from the current iterate and whether the global convergence of the interior point algorithm is still guaranteed.

In fact, Korzak [47], Curtis et al [26] and Al-Jeiroudi [6] present the convergence of their own inexact infeasible interior point algorithm, respectively. In [47], Korzak develops an inexact variant of the IPM algorithm of Kojima, Megiddo, and Mizuno, for linear programming. He also gives a convergence analysis of his inexact interior point algorithm, assuming that all the iterates are bounded; Curtis, Schenk and Wächter's work [26] consists of a sufficient merit function approximation reduction termination (SMART) tests in the IPM framework, and they also prove the convergence of their algorithm; Al-Jeiroudi [6] proves the convergence of the inexact infeasible path-following interior point algorithm with PCG as the iterative solver. For more details and background about inexact infeasible primal-dual IPM, see [6].

Unlike the other eearlier mentioned papers, Petra and Anitescu [56], as well as Qiu et al [57], present a parallel IPM implementation in which inexact step is only required for a small portion of the linear system arising in the IPM process. An analysis of convergence behavior is also presented in Petra and Anitescu's work [56], and PCG is the iterative method they have implemented. On the other hand, Qiu's paper [57] pays more attention to the parallel implementation of their interior point algorithm in which GMRES is the iterative solver. Further

details about these two papers will be discussed in Chapter 3.

Algorithm 2.2: Modified GMRES Algorithm

Step 1. Initialization:
Given an initial guess u_{init}, compute $r_{init} = s - \mathcal{H}u_{init}$, $\beta := \|r_0\|_2$, $v_1 := r_0/\beta$.

Step 2. Start Inner Loop to produce projection: For $j = 1, 2, \ldots, m$, do:
$w_j := \mathcal{H}v_j$
For $i = 1, 2, \ldots, j$, do
$h_{i,j} := w_j^\top v_i$
$w_j := w_j - h_{i,j}v_i$
end-do
$h_{j+1,j} := \|w_j\|_2$, if $h_{j+1,j} = 0$, set $m := j$ and go to **Step 3**
$v_{j+1} := w_j/h_{j+1,j}$
Define $\mathbf{V}_m := (v_1, \ldots, v_m)$, $e_1 := (1, 0, \ldots, 0)^\top$ and $\bar{\mathbf{H}}_m := \{h_{i,j}\}_{1\le i\le m+1, 1\le j\le m}$.

Step 2.1. Compute Givens rotation:
Apply Ω_k, $k = 1, \ldots, j-1$ to the j-th column of $\bar{\mathbf{H}}_j$
Compute parameter s_j and c_j of the j-th rotation

$$\Omega_j = \begin{pmatrix} I & & & \\ & c_j & s_j & \\ & -s_j & c_j & \\ & & & I \end{pmatrix} \begin{matrix} \\ \text{row i} \\ \text{row i+1} \\ \\ \end{matrix}$$

as:

$$s_j = \frac{h_{j+1,j}}{\sqrt{(h_{j,j}^{(j-1)})^2 + h_{j+1,j}^2}}, c_j = \frac{h_{j,j}^{(j-1)}}{\sqrt{(h_{j,j}^{(j-1)})^2 + h_{j+1,j}^2}}.$$

Define $\bar{R}_j = \Omega_j\Omega_{j-1}\ldots\Omega_1\bar{\mathbf{H}}_j$ and $\bar{g}_j = \Omega_j\Omega_{j-1}\ldots\Omega_1(\beta e_1)$.
Define R_j, a square matrix, and g_j by removing the last row of $\bar{R}_j$ and $\bar{g}_j$.

Step 2.2. Check Convergence:
If $|\bar{g}_j^{(j+1)}| < tol$, compute $u_j := u_{init} + \mathbf{V}_j y_j$, where $y_j = R_j^{-1}g_j$, **STOP**.
else go back to **Step 2**.
end-do (**Inner loop**)

Step 3. Restart:
Compute $u_m := u_{init} + \mathbf{V}_m y_m$, where $y_m = R_m^{-1}g_m$.
Compute $r_m := s - \mathcal{H}u_m$. If $\|r_m\|_2 < tol$, **STOP**.
else set $r_0 := r_m$, $\beta := \|r_m\|_2$, $v_1 := r_m/\beta$ and **Restart** go to **Step 2**.

Chapter 3

Solving DC-SCOPF

In this chapter, we focus on the techniques used to solve the DC-SCOPF model. Firstly, we reorder the original DC-SCOPF model to show its special structure and present a new equivalent reformulation of the model. In the second part of this chapter, the linear algebra of structure-exploiting IPMs is introduced. We also demonstrate three new structure-exploiting methods to solve the reformulated DC-SCOPF model. The last section of this chapter looks at the iterative methods which are used to solve certain linear systems arising in the IPM process. All numerical results are given at the end of each respective section.

3.1 Structure Exploitation for DC-SCOPF

For convenience, the linearized SCOPF model in Chapter 2 is repeated here:

DC-SCOPF Model:

$$\min: \quad c^\top p \tag{3.1}$$

$$s.t \quad p^- \le p \le p^+, \tag{3.2}$$

$$-f^+ \le f_c \le f^+, \quad \forall c \in \mathcal{C} \tag{3.3}$$

$$\mathbf{J}p + \mathbf{H}_c f_c = d, \quad \forall c \in \mathcal{C} \tag{3.4}$$

$$f_c = -\frac{V^2}{r}\mathbf{H}_c^\top \delta_c, \quad \forall c \in \mathcal{C} \tag{3.5}$$

$$\delta_{c,b\_0} = 0, \quad \forall c \in \mathcal{C}. \tag{3.6}$$

From the description of DC-SCOPF model in Chapter 2, we observe that the only global variables, which do not depend on the contingency case, are the real power generation p. Hence, if this global variable is ignored, the rest of the problem can be decomposed into separate sub-problems for each contingency. Since only contingencies corresponding to the outage of a single line are taken into consideration, each sub-problem is given by a particular reduced network, and it is similar in structure to the original OPF problem.

We use the same notation as in the previous chapter, i.e., subscript 0 denotes the sub-

problem given by the base-case condition, and other subscript numbers denote the contingencies corresponding to the outage of transmission line l. In addition, we use subscript G for the global variables.

All the variables are reordered by contingency, appending the global variables at the end, that is, the variables of the DC-SCOPF problem are reordered according to the sequence (3.7)

$$(f_{0,l}, \delta_{0,b}),\ (f_{1,l}, \delta_{1,b}),\ \ \ldots, (f_{|\mathcal{C}|,l},\ \delta_{|\mathcal{C}|,b}),\ p_G, \tag{3.7}$$

After this reordering, the whole constraint coefficient matrix of DC-SCOPF can be reformed as a large sparse bordered block diagonal matrix in which each diagonal block matrix corresponds to one contingency scenario. The bordered block diagonal structure is illustrated in Figure 3.1. From this figure, it is obvious that the size of the SCOPF problem can become very large as the number of contingencies increases: for each extra line outage considered in the network, one additional diagonal block is required to be added into the model.

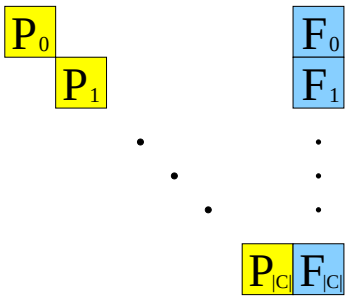

Figure 3.1: Block angular matrix

Obviously, there are many fixed variables in the DC-SCOPF model (3.2)-(3.6). For example, the phase angle at reference bus and the flow across a tripped line are forced to be equal to zero. To analyze the structure of DC-SCOPF model, we first need to consider how to deal with these fixed variables, which are represented as fixed columns in the corresponding linear system. Generally, all these fixed variables can be removed automatically by the use of a mathematical modeling language such as AMPL [31]. The advantage of using such a mathematical modeling language is that its presolve function can remove the fixed columns and the corresponding constraints/rows before solving the model, which means it can reduce the size of a problem and speed up the solution process in most cases. However, the weakness is that its algorithm is a "black box" which makes it difficult for a user to understand which actual matrix was used in solving the problem. Therefore, in order to present our new formulation which exploits the special structure of the DC-SCOPF model, it is necessary to switch off the presolve function from the modeling language. Instead, fixed variables are removed manually by not introducing line flow variables of the collapsed line, and furthermore by not introducing reference bus voltage phase variables for the contingency scenarios. In other words, variables $f_{c,c}$ and δ_{c,b_0}

are eliminated from the DC-SCOPF model. More details about the mathematical modeling languages can be found in chapter 6.

Moreover, in each contingency c, the Kirchoff Voltage Law is always presented before the Kirchoff Current Law, and similarly the flow variables are always stated before the voltage phase variables. That is, the following sequence (3.8) is used to reorder all the local variables for each contingency c:

$$f_{c,l_1},\ f_{c,l_2},\ \ldots,\ f_{c,l_{|\mathcal{L}|}},\ \delta_{c,b_1},\ \delta_{c,b_2},\ldots,\ \delta_{c,b_{|\mathcal{B}|}}, \tag{3.8}$$

After removing all the fixed variables and reordering other variables according to sequences (3.8) and (3.7), the coefficient matrix still retains the bordered block diagonal form shown in Figure 3.1. However, in the latter case, the diagonal block matrices are closer to being symmetric, which will be exploited later.

The 3-bus system presented in Chapter 2 is used as an example to illustrate the structure of the coefficient matrix of the DC-SCOPF problem. This small example only contains the base-case scenario and one contingency created by losing transmission line from bus 1 to bus 3. Its coefficient matrix is visualized in Figure 3.2, where the black blocks indicate the non-zero elements.

In detail, the first diagonal block corresponds to the basic case condition with no line outage, while the next block corresponds to the single contingency. The first three rows, together with the 7th and 8th rows, capture the Kirchoff's Voltage Law, while the other rows correspond to the Kirchoff's Current Law for each respective scenario. For comparison, the matrix with fixed column is given in Figure 3.3. The extra rows, along with the 7th row and the last row, together give the reference bus constraints, while the 8th row comes from the line outage constraint.

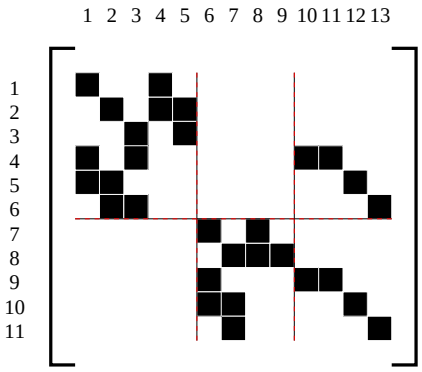

Figure 3.2: 3-bus example without fixed columns

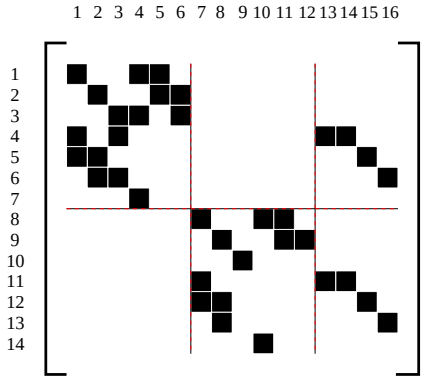

Figure 3.3: 3-bus example

3.1.1 Reformulation of the DC-SCOPF Model

The straightforward DC-SCOPF model can be represented as a bordered block diagonal structure as shown above, and each diagonal block of the block angular matrix can be modified to be symmetric. In the 3-bus example, it is obvious to observe that each diagonal block in Figure 3.2 can be symmetric if the 4th row and 9th row can be removed, which are the KCL constraints applied to the reference bus b_0.

Fortunately, due to the fact that there is no line loss in the DC model, the KCL constraint at one specific bus can be replaced by the total generation constraint $\sum_{g\in\mathcal{G}} p_g = \sum_{b\in\mathcal{B}} d_b$. Therefore, to achieve the symmetric diagonal block form, the KCL constraints at the reference bus b_0 are eliminated, while the total generation constraint is added into the model.

Note that there are many different ways to make the diagonal blocks symmetric, depending on how the fixed variables are introduced and how the substitution is performed. The approach we took in our implementation is superior for the following reasons:

- There are no fixed columns or corresponding row constraints.
- The size of the matrix is the smallest, as well as having the minimum number of nonzero elements.
- The row border contains the most zero entries, and its dimension is one.

For convenience and simplicity, the notation of matrix $\mathbf{H}_c$ is now used to indicate the bus-line incidence matrix for the reduced power network in contingency c, where both the columns/rows corresponding to the reference bus and missing line are eliminated. Hence, $\mathbf{H}_c \in \mathcal{R}^{(|\mathcal{B}|-1)\times(|\mathcal{L}\cup\mathcal{T}|-1)}$ for the contingency case, and $\mathbf{H}_c \in \mathcal{R}^{(|\mathcal{B}|-1)\times|\mathcal{L}\cup\mathcal{T}|}$ for the base case without line outage. Similarly, $\mathbf{J} \in \mathcal{R}^{(|\mathcal{B}|-1)\times|\mathcal{G}|}$ is the bus-generator incidence matrix without the contributions from the reference bus and, indeed, J is the same for all contingencies.

Writing $e = (1, 1, \ldots, 1)$ in the appropriate dimension, the new reformulated model can be stated as

Modified Model:	min	$c^\top p$	(3.9)
	$s.t$	$p^- \le p \le p^+,$	(3.10)
		$-f^+ \le f_c \le f^+, \quad \forall c \in \mathcal{C},$	(3.11)
		$f_c = -\frac{V^2}{r_l}\mathbf{A_c}^\top \delta_c, \quad \forall c \in \mathcal{C},$	(3.12)
		$\mathbf{J}p + \mathbf{A_c} f_c = d, \quad \forall c \in \mathcal{C},$	(3.13)
		$e^\top p = e^\top d.$	(3.14)

Note that, with this reformulation, the coefficient matrix still can have a block angular form, but this time it contains one more diagonal block at the bottom. Its structure is shown in Figure 3.4, which can also be summarized in a bordered block form as Figure 3.5, where

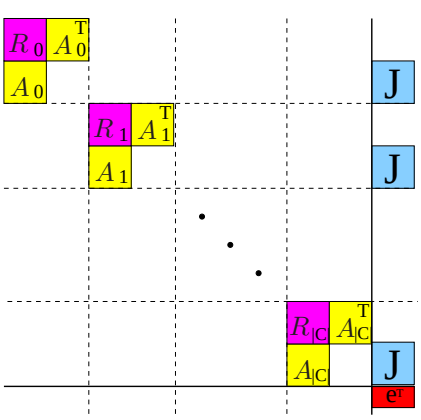

Figure 3.4: Constraint matrix of the modified model.

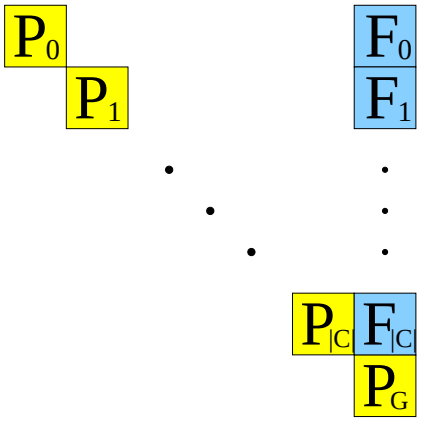

Figure 3.5: Block structure of the constraint matrix.

$$\mathbf{P}_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{H}_c^\top \\ \mathbf{H}_c & \mathbf{0} \end{bmatrix}, \quad \mathbf{R}_c = \operatorname{diag}(\frac{r_1}{V^2}, \frac{r_2}{V^2}, \ldots, \frac{r_{c-1}}{V^2}, \frac{r_{c+1}}{V^2}, \ldots, \frac{r_{|\mathcal{L}|}}{V^2}) \tag{3.15}$$

and

$$\mathbf{F}_c = \mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{J} \end{bmatrix}. \tag{3.16}$$

An example can be found from the corresponding sparsity structure of the 3-bus system in Figure 3.6.

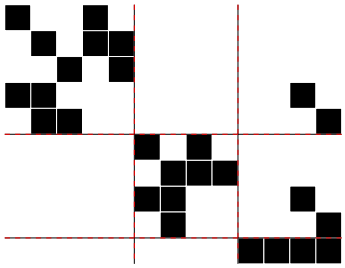

Figure 3.6: New formulation with symmetric diagonal blocks.

The major advantage of applying this new formulation is that each block diagonal matrix $\mathbf{P}_c$ is symmetric and it shares the same structure as the augmented system matrix for the KKT system (2.59), and it is natural to ask whether this structure can be exploited. In fact, the leading diagonal block of the matrices $\mathbf{P}_c$ is a diagonal matrix $\mathbf{R}_c$ whose entries are ratios of the parameters r_l and V. Since the parameter r_l denotes the reactance of the transmission line l, and V denotes the voltage level in the power system, the ratios $\frac{r_l}{V^2}$ are bounded away from zero and all the entries of $\mathbf{R}_c$ have the same order of magnitude. In other words, the matrices $\mathbf{R}_c$ must be well-scaled. On the other hand, the constant matrix $\mathbf{H}_c$ is a node-arc incidence matrix with full row rank. Therefore, the whole matrix $\mathbf{P}_c$ is invertible and is eligible for the

$\mathbf{LDL}^\top$ factorization (2.63). The factorization can start its pivoting from the top-left diagonal block $\mathbf{R}_c$ and no more regularization is required to maintain the stability.

3.2 Structure-Exploiting IPM for LP

In the previous chapter, we have stated that the main computational effort of the IPM is in solving the linear system (2.59). This can be done by obtaining $\mathbf{LDL}^\top$ factors of the augmented system matrix (2.59) and then producing the back-solve sequence (2.64). However, during this process, we do not require any information from the problem structure. If we take the specific block structure into consideration, we can expect to gain more advantages during the IPM process. In what follows, we explore how the block angular structure of the coefficient matrix affects an IPM implementation.

Recalling the structured coefficient matrix for the DC-SCOPF problem from Figure 3.5, its corresponding augmented system matrix can be expressed as in Figure 3.7, where the matrices $Q_i, i \in \{0, 1, \ldots, |C|, G\}$ represent the appropriate sections of the diagonal matrices $-\Theta$. After

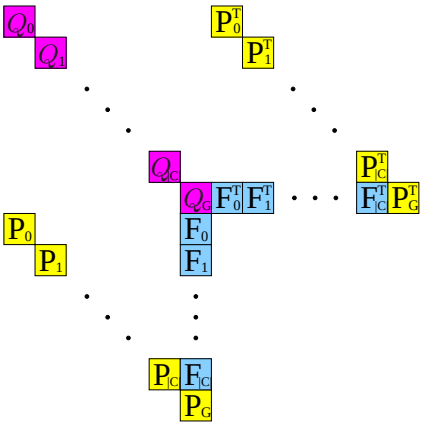

Figure 3.7: Augmented system matrix.

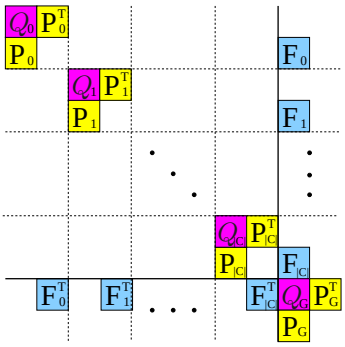

Figure 3.8: Reordered augmented system matrix

reordering the rows and columns of the matrix, the reordered augmented system matrix can be expressed as in Figure 3.8. This reordered matrix is double bordered block diagonal and each diagonal block matrix is itself an augmented system formulation corresponding to a contingency sub-problem. Note that this structure-exploiting method is also applied in [36, 37].

To simplify the notation, we use the following form (3.17) [37]

$$\Phi = \begin{pmatrix} \Phi_0 & & & & \mathbf{B}_1^\top \\ & \Phi_1 & & & \mathbf{B}_2^\top \\ & & \ddots & & \vdots \\ & & & \Phi_n & \mathbf{B}_n^\top \\ \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_n & \Phi_G \end{pmatrix}, \tag{3.17}$$

where $n = |\mathcal{C}|$, $\Phi_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{P}_i^\top \\ \mathbf{P}_i & \mathbf{0} \end{bmatrix}$ and $\mathbf{B}_i^\top = \mathbf{B}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} \end{bmatrix}$ for all $i = 0, 1, \ldots, n, G$, to represent the reordered augmented system matrix in Figure 3.8. Note that we can drop index i from $\mathbf{B}_i^\top$, since $\mathbf{B}$ does not depend on the contingency i.

The augmented system can be generalized to the following linear system

$$\Phi \boldsymbol{x} = \mathbf{b}, \tag{3.18}$$

where $\boldsymbol{x} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n, \boldsymbol{x}_G)$ and $\mathbf{b} = (\mathbf{b}_0, \ldots, \mathbf{b}_n, \mathbf{b}_G)$.

In order to solve the augmented system (3.18), the linear algebra implementation in OOPS [36], as well as the framework used by others [43, 57, 56], uses a Schur complement factorization method. That is, the solution to (3.18) is calculated from the following sequence of linear equations:

$$\boldsymbol{z}_i = \Phi_i^{-1}\mathbf{b}_i, \quad i = 0, \ldots, n, \tag{3.19}$$

$$\boldsymbol{x}_G = \mathbf{C}^{-1}(\mathbf{b}_G - \sum_{i=0}^{n} \mathbf{B}_i \boldsymbol{z}_i), \tag{3.20}$$

$$\boldsymbol{x}_i = \boldsymbol{z}_i - \Phi_i^{-1}\mathbf{B}_i^\top \boldsymbol{x}_G, \quad i = 0, \ldots, n, \tag{3.21}$$

where

$$\mathbf{C} = \Phi_G - \sum_{i=0}^{n} \mathbf{B}_i \Phi_i^{-1} \mathbf{B}_i^\top \tag{3.22}$$

is the Schur complement matrix. It is worth mentioning that the dimension of the matrix $\mathbf{C}$ is the number of global variables. For the DC-SCOPF model, this value is equal to the number of real power generators, n_g, plus one.

From equations (3.19) - (3.21), it is obvious that the factorization of the entire augmented system matrix Φ can be avoided. As an alternative, we need to find a sequence of solutions to the smaller augmented systems $\Phi_i \tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_i$, where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{b}}$ denote any appropriate variables and right hand side appearing in this linear system in scenario i.

Then the question is how to calculate these different steps efficiently. In this section, we assume all the above linear systems are solved by the direct method. Hence, the Schur complement matrix $\mathbf{C}$ is required to be formed explicitly, which is very computationally expensive.

In the rest of this section, two main approaches to solving the above sequence of equations are demonstrated, namely:

- Apply Cholesky factorization to $\Phi_i = \mathbf{L}_i \mathbf{D}_i \mathbf{L}_i^\top$, which implies that both Cholesky factors $\mathbf{L}_i$ and $\mathbf{D}_i$ are available. This is the main approach used in the literature mentioned above, and it is briefly represented in Section 3.2.1.

- If a matrix $\mathbf{P}_i$ is symmetric and invertible, e.g. the constraint matrix of a DC-SCOPF

problem with the modified model mentioned in previous section, we can design some new approaches to computing the solution to the linear system $\Phi_i \tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_i$ by exploiting this special property. These approaches are based on applying factorization techniques to the constant matrix $\mathbf{P}_i$. Most important of all, matrix $\mathbf{P}_i$ is a sub-matrix of $\Phi_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{P}_i^\top \\ \mathbf{P}_i & \mathbf{0} \end{bmatrix}$, and its dimension is exactly half the size of Φ_i. Hence the factorization techniques and their corresponding back-solve process are applied to a smaller linear system, which implies that the new approaches can be more efficient to solve the linear system (3.18). Therefore, a very efficient technique to solve the DC-SCOPF problems can be expected. All the new approaches with factorization of $\mathbf{P}_i$ are discussed in Section 3.2.2.

3.2.1 Default Structure-Exploiting Techniques

The most straightforward way to solve the linear systems arising in equations (3.19), (3.21) and (3.22), is to apply Cholesky factorization to the corresponding matrix, namely $\Phi_i = \mathbf{L}_i \mathbf{D}_i \mathbf{L}_i^\top$.

As mentioned in [37], the above process (3.19)-(3.21) can also be viewed as applying a block Cholesky type factorization to the full matrix Φ, namely

$$\Phi = \mathbf{L}\mathbf{D}\mathbf{L}^\top, \tag{3.23}$$

where

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_0 & & & & \\ & \mathbf{L}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{L}_n & \\ \mathbf{L}_{n,0} & \mathbf{L}_{n,1} & \cdots & \mathbf{L}_{n,n} & \mathbf{L}_C \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} \mathbf{D}_0 & & & & \\ & \mathbf{D}_2 & & & \\ & & \ddots & & \\ & & & \mathbf{D}_n & \\ & & & & \mathbf{D}_C \end{pmatrix}, \tag{3.24}$$

and

$$\Phi_i = \mathbf{L}_i \mathbf{D}_i \mathbf{L}_i^\top, \quad \forall i = 0, \ldots, n \tag{3.25}$$

$$\mathbf{L}_{n,i} = \mathbf{B}_i \mathbf{L}_i^{-\top} \mathbf{D}_i^{-1}, \quad \forall i = 0, \ldots, n \tag{3.26}$$

$$\mathbf{C} = \Phi_G - \sum_{i=0}^{n} \mathbf{B}_i \Phi_i^{-1} \mathbf{B}_i^\top = \mathbf{L}_C \mathbf{D}_C \mathbf{L}_C^\top. \tag{3.27}$$

This factorization and its corresponding back-solve sequence is applied to obtain the solution

of the system (3.18). The steps of the back-solve are given by the following sequence:

$$\begin{aligned}
\boldsymbol{z}_i &= \mathbf{L}_i^{-1}\mathbf{b}_i, \quad i = 0, \ldots, n && (3.28)\\
\boldsymbol{z}_G &= \mathbf{L}_C^{-1}(\mathbf{b}_G - \sum_{i=0}^{n} \mathbf{L}_{n,i}\boldsymbol{z}_i), && (3.29)\\
\boldsymbol{y}_i &= \mathbf{D}_C^{-1}\boldsymbol{z}_i, \quad i = 0, \ldots, n, G && (3.30)\\
\boldsymbol{x}_G &= \mathbf{L}_C^{-\top}\boldsymbol{y}_G, && (3.31)\\
\boldsymbol{x}_i &= \mathbf{L}_i^{-\top}(\boldsymbol{y}_i - \mathbf{L}_{n,i}^{\top}\boldsymbol{x}_G), \quad i = 0, \ldots, n && (3.32)
\end{aligned}$$

Note that in OOPS [37], the matrices $\mathbf{L}_{n,i}$ have never been formulated explicitly. The use of such terms is only applied to illustrate the block Cholesky type factorization presented in (3.24).

As mentioned before, we can avoid factorizing the large matrix Φ directly, and instead, we find the Cholesky factors of the smaller augmented matrices Φ_i and the Schur complement matrix $\mathbf{C}$. These matrices, however, are much smaller systems than Φ.

Although huge computational efforts can be saved by the above scheme, the downside is that forming the Schur complement matrix $\mathbf{C}$ explicitly may be expensive, even for relatively small dimension n_g when $\mathbf{C}$ is dense. From the definition of $\mathbf{C}$ in (3.22), it is obvious that for each contingency i, its contribution to $\mathbf{C}$ is $\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^{\top}$. The later matrix product $\Phi_i^{-1}\mathbf{B}_i^{\top}$ requires the solution of $n_g + 1$ linear systems of the form $\Phi_i\tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_i$, one for each column $\tilde{\mathbf{b}}_i$ in $\mathbf{B}_i$. Moreover, although the matrix $\mathbf{B}_i$ may be sparse, the product $\Phi_i^{-1}\mathbf{B}_i^{\top}$ can be very dense. This awkward result can also decrease the efficiency of computing the contribution $\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^{\top}$.

In fact, it is often better to calculate $\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^{\top}$ using

$$\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^{\top} = (\mathbf{L}_i^{-1}\mathbf{B}_i^{\top})^{\top}\mathbf{D}_i^{-1}(\mathbf{L}_i^{-1}\mathbf{B}_i^{\top}), \tag{3.33}$$

since the matrix product $\mathbf{L}_i^{-1}\mathbf{B}_i^{\top}$ may be quite sparse. The implementation of OOPS [37] saves the matrix product $\mathbf{L}_i^{-1}\mathbf{B}_i^{\top}$ as a sparse matrix $\mathbf{V}_i$ and computes the Schur complement $\mathbf{V}_i^{\top}\mathbf{D}_i^{-1}\mathbf{V}_i$ by getting row-wise access to $\mathbf{V}_i$. That is, for all rows $\mathbf{r}_j$ in $\mathbf{V}_i$, OOPS updates the Schur complement: $\mathbf{C} = \mathbf{C} - (\mathbf{r}_j * \mathbf{r}_j^{\top})/\mathbf{d}_j$, where $\mathbf{d}_j$ is the jth entry of the diagonal matrix $\mathbf{D}_i$.

Additionally, matrix Φ_i must be updated iteration by iteration since the $(1,1)$ sub-matrix of Φ_i, that is Θ_i^k, depends on current IPM iterates $\boldsymbol{x}_i^k$ and $\boldsymbol{s}_i^k$, where k is the index of iterations. Therefore, the numerical factorization of the matrix Φ_i needs to be recomputed in each IPM iteration.

Note that this method is considered as the default structure-exploiting method in this thesis. It can be applied not only to the DC-SCOPF problem, but also to any other problem with a block angular form. We will use this method, and make a compariaon with three other

sophisticated alternatives which are particularly designed to solve the DC-SCOPF problem.

3.2.2 Structure-Exploitation IPM for DC-SCOPF

The augmented system matrix of the DC-SCOPF problem has a double bordered block diagonal form (3.17), where each diagonal block is of the form $\Phi_i = \begin{bmatrix} -\Theta_i & \mathbf{P}_i^\top \\ \mathbf{P}_i & \mathbf{0} \end{bmatrix}$. The default Schur complement factorization discussed above can also be applied to factorize and solve the linear systems with this system matrix. Besides, once the modified model (3.9) is applied, the matrix $\mathbf{P}_i$ can itself be of augmented system format, as shown in (3.15). Since the matrix is symmetric and invertible, an obvious question arising is how this favourable property can be exploited.

In what follows, a new method to solve the smaller augmented system $\Phi_i \boldsymbol{x}_i = \mathbf{b}_i$ is introduced, in which the use of Cholesky factorization of the matrices Φ_i can be replaced by the Cholesky factorization of the partial coefficient matrix $\mathbf{P}_i$. Obviously, the matrix $\mathbf{P}_i$ is half the size of the matrix Φ_i, while its dimension is the number of local variables corresponding to contingency i.

Note that solving the linear system $\Phi_i \tilde{\mathbf{x}}_i = \tilde{\mathbf{b}}_i$ is equivalent to finding the solution $(\tilde{\mathbf{x}}_{i,1}, \tilde{\mathbf{x}}_{i,2})$ to the system

$$\begin{bmatrix} -\Theta_i & \mathbf{P}_i^\top \\ \mathbf{P}_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{i,1} \\ \tilde{\mathbf{x}}_{i,2} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{b}}_{i,1} \\ \tilde{\mathbf{b}}_{i,2} \end{bmatrix}. \tag{3.34}$$

After we have obtained the Cholesky factors of the matrix $\mathbf{P}_i = \tilde{\mathbf{L}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{L}}_i^\top$, the solution process applied to system (3.34) can be defined by the following sequence

$$\tilde{\mathbf{x}}_{i,1} = (\tilde{\mathbf{L}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{L}}_i^\top)^{-1} \tilde{\mathbf{b}}_{i,2} \tag{3.35}$$

$$\tilde{\mathbf{x}}_{i,2} = (\tilde{\mathbf{L}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{L}}_i^\top)^{-1} (\tilde{\mathbf{b}}_{i,1} + \Theta_i \tilde{\mathbf{x}}_{i,1}), \tag{3.36}$$

For the sake of brevity, we use the notation Φ_i^{-1} to indicate the solving process (3.35)-(3.36), and hence we can follow the notation presented in (3.19)-(3.21).

Comparing the first equation (3.35) with the second one (3.36), it is obvious that (3.36) can only be solved after the first one, since it consists of the solution $\tilde{\mathbf{x}}_{i,1}$ from the first equation. Furthermore, these two equations imply that two full back-solve processes given by $(\tilde{\mathbf{L}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{L}}_i^\top)^{-1}$ are required to solve one linear system of (3.34).

Compared to the default method where factorization of the matrix Φ_i must be updated in each IPM iteration due to the changing term Θ, the major advantage of the new solution sequence (3.35)-(3.36) is that this new approach only requires factorizing the constant matrix $\mathbf{P}_i$, which is given by (3.34). Hence, the numerical factorization of the matrix $\mathbf{P}_i$ only needs to be calculated once at the very beginning of the entire IPM process, and it can be utilized in

later iterations.

Another advantage is that, as we described before, there is no need for additional pivoting and regularization to factorize $\mathbf{P}_i$ since it is well-scaled. Due to the above reasons, factorizing $\mathbf{P}_i$ and applying a back-solve process according to (3.35)-(3.36) should improve the stability of the algorithm and speed up the solution process.

On the other hand, observe that the Cholesky factors of the Schur complement matrix $\mathbf{C}$ are still required in this method, in order to solve the linear system appearing in (3.20). Hence it is necessary to have an explicit expression for the Schur complement matrix $\mathbf{C}$. However, now it is impossible to use equation (3.33) to form the Schur complement matrix $\mathbf{C}$ due to the lack of Cholesky factors of Φ_i, i.e the matrices $\mathbf{L}_i$ and $\mathbf{D}_i$ mentioned in (3.33) are not available now. Hence we need to find an alternative for evaluating $\mathbf{C}$ efficiently.

From (3.17), $\mathbf{B}_i^\top = \mathbf{B}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} \end{bmatrix}$, where $\mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{J} \end{bmatrix}$, we can represent the Schur complement contribution $\mathbf{B}\Phi_i^{-1}\mathbf{B}^\top$ by

$$\mathbf{B}\Phi_i^{-1}\mathbf{B}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{F}^\top \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \Phi_i^{-1} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} \end{bmatrix}, \tag{3.37}$$

In addition, a further step can be derived if the solution sequence (3.35)-(3.36) is taken into consideration. Recalling that $\mathbf{P}_i$ is symmetric, we can have

$$\mathbf{B}\Phi_i^{-1}\mathbf{B}^\top = \begin{bmatrix} \mathbf{0} & \mathbf{F}^\top \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_i^{-1}\mathbf{F} & 0 \\ \mathbf{P}_i^{-\top}\Theta_i\mathbf{P}_i^{-1}\mathbf{F} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{F}^\top\mathbf{P}_i^{-\top}\Theta_i\mathbf{P}_i^{-1}\mathbf{F} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.38}$$

Three alternative approaches are given to build an explicit representation of the Schur complement $\mathbf{C}$:

- *A*) Evaluate $\Phi_i^{-1}\mathbf{B}^\top$ by solving $n_g + 1$ linear systems according to the sequence (3.35)-(3.36), where n_g is the number of real power generators. In other words, the contingency contribution $\mathbf{B}\Phi_i^{-1}\mathbf{B}^\top$ is obtained by evaluating $\mathbf{F}^\top\mathbf{P}_i^{-\top}\Theta_i\mathbf{P}_i^{-1}\mathbf{f}$, from right to left for each column $\mathbf{f}$ of $\mathbf{F}$.

  Method *A* is the most straightforward among the three new methods, as it directly computes $\mathbf{F}^\top\mathbf{P}_i^{-\top}\Theta_i\mathbf{P}_i^{-1}\mathbf{F}$ from right to left, column by column.

- *B*) Observe that $\mathbf{P}_i^{-1}\mathbf{F}$ is a constant matrix, and that this term is exactly the first sub-block matrix of the solution to the linear system $\Phi_i\tilde{\mathbf{x}}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{F} \end{bmatrix}$, which can also be shown from the first equivalence in (3.38). On the other hand, the other sub-block solution $\mathbf{P}_i^{-\top}\Theta_i\mathbf{P}_i^{-1}\mathbf{F}$ needs to be calculated in every IPM iteration, since the inside diagonal term Θ_i is changing. To utilize this observation, an approach is designed to compute the fixed sub-solution $\mathbf{P}_i^{-1}\mathbf{F}$ in the first IPM step and record this partial solution for later

use. In particular, when forming the Schur complement matrix C in the later iterations of IPM, the saved data $\mathbf{P}_i^{-1}\mathbf{F}$ can be used to calculate the final nonzero product arising in (3.38). Note that in this method, the nonzero product $\mathbf{F}\mathbf{P}_i^{-\top}\Theta_i(\mathbf{P}_i^{-1}\mathbf{F})$ is computed from right to left, which means the second back-solve process defined by the left inverse of matrix $\mathbf{P}_i^{-1}$ is calculated for each column of $\Theta_i(\mathbf{P}_i^{-1}\mathbf{F})$.

- *C*) Based on Method *B*, the partial solution $\mathbf{P}_i^{-1}\mathbf{F}$ is also recorded for later use. Then, similar to the equation (3.33) in the default method, the nonzero sub-matrix arising in the last equivalence in (3.38) can be reformulated as

$$(\mathbf{F}\mathbf{P}_i^{-\top})\Theta_i(\mathbf{P}_i^{-1}\mathbf{F}) = \tilde{\mathbf{V}}_i^T\Theta_i\tilde{\mathbf{V}}_i, \tag{3.39}$$

where the matrix $\tilde{\mathbf{V}}_i$ is exactly the saved data $\mathbf{P}_i^{-1}\mathbf{F}$. After setting the row-wise access to $\tilde{\mathbf{V}}_i$, the contribution $\tilde{\mathbf{V}}_i^T\Theta_i\tilde{\mathbf{V}}_i$ can be obtained in the same way as was described in section 3.2.1.

Each of these three methods has its own strengths and weaknesses. Compared to the default method in the last section, Method *A* should require less storage demand since it only requires factorizing smaller matrices $\mathbf{P}_i$. However, for each column $\tilde{\mathbf{b}}$ of $\mathbf{B}^\top$, the operation $\Phi^{-1}\tilde{\mathbf{b}}$, according to (3.35) and (3.36), requires two full back-solves with the $\mathbf{LDL}^\top$ factors of $\mathbf{P}_i$, consisting of one each in (3.35) and (3.36) respectively. Since $\mathbf{B}^\top$ contains n_g+1 columns, the main computational work of this method is equivalent to performing $2\times(n_g+1)$ full back-solve processes for each contingency in order to build the Schur complement matrix $\mathbf{C}$ in one IPM step. On the other hand, only one full back-solve is required in the default method, albeit with the larger Cholesky factors of the matrix Φ_i whose dimension is twice that of the matrix $\mathbf{P}_i$. Nevertheless, the computational work of the default method is to solve this larger back-solve system (n_g+1) times in each IPM iteration, which is half the number required in Method *A*. Hence it is difficult to say which method is dominant over the other one.

Method *B* definitely requires more storage demands since it requires the saving of partial solutions. However, Method *B* should be faster than Method *A*, since the first full back-solve sequence (3.35), the one before applying matrix multiplication, can be eliminated by the use of the recorded data. Hence, in each iteration, it requires (n_g+1) back-solve steps, which is the same as the number appearing in the default method. However, now its back-solve is performed for the smaller constant system $\mathbf{P}_i$. As a result, Method *B* should be faster than Method *A* without doubt.

Finally, the last method, Method *C*, should have similar storage demand to Method *B*, since it saves the same data in $\tilde{\mathbf{V}}_i$. However, unlike the situation in (3.33), computing the contribution $\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^\top$ may not be efficient if the row-wise access of $\tilde{\mathbf{V}}_i = \mathbf{P}_i^{-1}\mathbf{F}$ is used. This is because, in the default method, the matrix $\mathbf{V}_i = \mathbf{L}_i^{-1}\mathbf{B}_i^T$ is typically sparse, where $\mathbf{L}_i$ is the

Cholesky factor of matrix Φ_i; on the other hand, in Method C, the matrix $\tilde{\mathbf{V}}_i$ is obtained by taking columns from $\mathbf{P}_i^{-1}\mathbf{F}$, which may be dense. In fact, it is shown in the next section that the matrix $\mathbf{P}_i^{-1}\mathbf{F}$ is extremely dense for the DC-SCOPF problem.

In this section, we have presented three methods to solve the augmented system arising in the IPM if the coefficient matrix is symmetric and invertible, and among these three methods, Method B is expected to be the fastest one to solve DC-SCOPF problems. These three new approaches, as well as the default method discussed in the previous section, are summarized in Table 3.1 for comparison.

Method	Detail	Factorize	Way to build $\mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^\top$	Save Data
Default	Section 3.2.1	Φ_i	$\mathbf{V}_i^\top\mathbf{D}_i^{-1}\mathbf{V}_i$, $\mathbf{V}_i = \mathbf{L}_i^{-1}\mathbf{B}_i^\top$	N
A	Paragraph A)	$\mathbf{P}_i$	Compute (3.38) from right to left	N
B	Paragraph B)	$\mathbf{P}_i$	Compute (3.38) from right to left	Y
C	Paragraph C)	$\mathbf{P}_i$	$\tilde{\mathbf{V}}_i^\top\Theta_i\tilde{\mathbf{V}}_i$, $\tilde{\mathbf{V}}_i = \mathbf{P}_i^{-1}\mathbf{F}$	Y

Table 3.1: Four different methods

3.2.3 Numerical Experiments

In order to test the new structure-exploiting approaches mentioned previously, they are implemented in OOPS [37], a structure-exploiting IPM solver, in the C++ programming language. Some sample power networks are used and solved by the three new approaches, as well as the default OOPS implementation. These test problems are summarized in Table 3.2. The first column indicates the number of buses in each sample system, while the third column gives the number of contingencies included in the SCOPF model, and the last column denotes the number of nonzero elements in the coefficient matrix. Note that all these models are assembled in the AMPL modeling language [31] in which the presolve function is blocked.

In what follows, the term "full contingencies" is used to denote all the contingency scenarios which can be taken into account in the SCOPF model. It may not be the 'full n-1' model since containing certain lines would cause the problem to become infeasible.

Buses	Generators	Contingencies	Variables	Constraints	Nonzeros
3	4	2	17	14	35
26	5	40	2,630	2,626	7,929
56	7	79	10,648	10,642	31,060
100	25	180	50,344	50,320	165,649
200	50	370	210,779	210,730	701,249
300	75	565	488,534	488,460	1,635,824
400	100	760	881,339	881,240	2,960,399
500	125	955	1,389,194	1,389,070	4,674,974

Table 3.2: Problem details of the test problems.

Among these test problems, the 3-bus system is the small sample problem shown in this

chapter. Its largest number of contingency scenarios is two, since it would become an infeasible problem once the transmission line from bus 1 to bus 3 is tripped. The infeasibility comes from the high local load at bus 3, while there is not enough power that can be shipped from other buses to this place. The problems with 26 and 56 buses are modified versions of the original IEEE test problems with 30 buses and 57 buses [80], respectively. Both of them require eliminating some buses and reconnecting the transformer lines to ensure the problem is fully n-1 feasible. In other words, each bus is connected to the network with at least two transmission lines to guarantee that the n-1 SCOPF formulation is meaningful and feasible. This therefore means all the one line outage contingencies can be taken into account. Other test problems with more than 56 buses are manually made up from random data, using a 2D grid-like transmission network shown in Figure 3.9, where the first bus is a PV bus. The number of generators is fixed

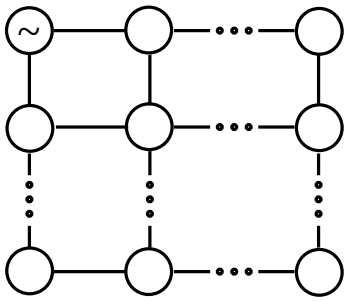

Figure 3.9: 2D grid-like transmission network.

to be one quarter of the total number of buses, and generators are connected to a bus after each three pure load buses. Other values of parameters, i.e. line reactance, line flow limits and etc, are generated as uniform random data, using the same range of magnitude as the IEEE test problems.

All these examples are solved on a 2.33GHz Intel(R) Xeon(R) computer with 4GB RAM, running Redhat Enterprise Linux. The code is compiled with compiler gcc with its optimizations control option **-O2**. The tolerance of convergence to the optimal solution is set to 10^{-6}. Each test problem is solved by OOPS under at least 80 different sets of parameters, such as different combinations of the initial regularization and the size of the rolling dynamic regularization. After finding the best parameter setting for each problem and each solution method mentioned in Table 3.1, each problem is solved at least three times to collect the average solution time (a small example may need to be run hundreds of times to get relatively accurate information). Note that the CPU time is used as the criteria for the solution time. Other information, such as the number of IPM iterations and the highest memory demand for solving these examples to reach convergence, is also recorded. All the numerical results from solving these DC-SCOPF problems are summarized in Table 3.3.

From Table 3.3, we can observe that Method A has the least memory demand among all the tested methods. This is obvious since its factorization is applied on the smaller dimensioned

	Default Method			Method A		
Buses	Time(s)	Memory	Iters	Time(s)	Memory	Iters
3	<0.01	5.2MB	8	<0.01	5.2MB	8
26	0.21	7.6MB	13	0.17	7.4MB	13
56	1.00	14.3MB	15	0.82	13.5MB	15
100	9.02	54.6MB	20	7.45	42.5MB	20
200	65.97	220MB	27	61.93	161MB	27
300	251.70	531MB	34	261.07	378MB	33
400	955.32	985MB	59	1023.71	699MB	53
500	1552.80	1593MB	49	1877.23	1109MB	47
	Method B			Method C		
Buses	Time(s)	Memory	Iters	Time(s)	Memory	Iters
3	<0.01	5.2MB	8	<0.01	5.2MB	8
26	0.17	7.5MB	13	0.16	7.5MB	13
56	0.77	14.1MB	15	0.72	14.0MB	15
100	6.16	53.1MB	20	5.39	53.1MB	20
200	45.23	244MB	27	41.88	244MB	27
300	177.39	667MB	33	167.85	663MB	33
400	655.50	1380MB	53	659.22	1366MB	54
500	1195.77	2467MB	47	1252.20	2466MB	47

Table 3.3: Test results from four methods in OOPS.

matrix $\mathbf{P}_i$ and it does not need to save any extra information. Focusing on the efficiency side, it can solve the smaller problems, up to a system of size 200, faster than the default method, but it loses its competitiveness in the larger problems. Hence it shows that, in large DC-SCOPF problems, it is better to solve a double-sized back-solve system $(n_g + 1)$ times than to use a smaller, but more dense, back-solve system $2 \times (n_g + 1)$ times.

Generally, Method B is the most efficient method for DC-SCOPF problems. For the largest problems, it can save approximately one third of the time required by the default method or Method A. On the other hand, both methods A and B require the least number of IPM iterations to reach convergence, which is an indication of the algorithm stability. It shows that the factorization of the constant matrices is much more stable, since no small pivots arise. The greatest disadvantage of Method B is that it occupies much more storage space for the large problems than either of the previous two methods does, since additional data must be saved for later use. This is due to the very dense matrix $\mathbf{P}_i^{-1}\mathbf{F}_i$ which must be recorded.

In fact, the same problems are observed in the last Method C, in which the Schur complement contribution is obtained by calculating $\tilde{\mathbf{V}}_i^\top \Theta_i \tilde{\mathbf{V}}_i$, where $\tilde{\mathbf{V}}_i = \mathbf{P}^{-1}\mathbf{F}$. The row-wise method is applied to compute this contribution, which is similar to the default OOPS method. Nevertheless, this technique is designed to reduce the memory demand for large or extremely large problems, where the matrices $\mathbf{V}_i$ in (3.33) are always quite sparse because of the Cholesky factor $\mathbf{L}_i$ of Φ_i and the sparse bordered matrix $\mathbf{B}_i$. As mentioned before, the matrix $\tilde{\mathbf{V}}_i = \mathbf{P}^{-1}\mathbf{F}$ can be very dense due to the full back-solve process from $\mathbf{P}^{-1}$, and consequently the conversion to row-wise may not save any work. For instance, for the largest test problem with 500 buses,

the sparsity ratio of each block $\tilde{\mathbf{V}}_i$ is demonstrated in Table 3.4. Obviously, Table 3.4 illustrates

	case	dimension	nonzeros	percentage
$\tilde{\mathbf{V}}_0$	OPF	126×1454	180,296	98.41%
$\tilde{\mathbf{V}}_i$	Contingency	126×1453	180,172	98.32%

Table 3.4: Sparsity behavior in the 500 buses system

that the matrices $\tilde{\mathbf{V}}_i$ are extremely dense. In order to compute the Schur complement matrix C more efficiently, the use of the state-of-the-art dense linear algebra library BLAS [28], which is a very robust and common tool in numerical computation, is suggested to calculate the contribution by (3.39). From the level 3 BLAS functions, the routine DSYRK, which provides a way to perform the matrix operation $\alpha\mathbf{A}\mathbf{A}^\top + \beta\mathbf{B}$, is applied to perform the dense symmetric matrix operation $\tilde{\mathbf{V}}_i^\top \Theta_i \tilde{\mathbf{V}}_i$ instead. However, since the matrices $\tilde{\mathbf{V}}_i$ are still too dense, numerical results show that Method C is only competitive on the smaller problem, i.e, it is the fastest algorithm to solve the problem with size less than or equal to 300 buses.

In words, if the major concern is the efficiency performance of the largest problem, Method B is the best one among these four IPM implementations in OOPS. It shows that saving the partial solution of the Schur complement matrix C can definitely speed up the process. However, the fierce storage requirement would make it awkward for solving very large problems. On the other hand, no matter which method is applied, assembling the Schur complement is still computationally expensive and the work involved is quite out of proportion, roughly more than 30%, compared to the entire process. To overcome this difficulty, the use of iterative methods may be a remedy, since it can avoid building $\mathbf{C}$ explicitly, which is presented as an extension to Method B in the next section.

3.3 Preconditioned Iterative Method in IPM

All the methods discussed so far in the previous section, i.e, the default OOPS implementation and the three new approaches, are aimed at finding the solution vector of global variables $\mathbf{x}_G$ by (3.20). That is, they apply a direct method to solve the linear system (3.40) by computing the inverse of $\mathbf{C}$ as $(\mathbf{L}_C\mathbf{D}_C\mathbf{L}_C^\top)^{-1}$, where $\mathbf{L}_C$ and $\mathbf{D}_C$ are the Cholesky factors of the Schur complement matrix $\mathbf{C}$.

$$\mathbf{C}\boldsymbol{x}_G = \mathbf{b}_G - \sum_{i=0}^{n} \mathbf{B}_i \boldsymbol{z}_i. \tag{3.40}$$

As a result, it is vital to build the matrix $\mathbf{C}$ explicitly. Moreover, if a new approach is applied to save partial solutions for later use in building $\mathbf{C}$, it requires a great deal of extra storage due to the very dense matrix product $\mathbf{P}_i^{-1}\mathbf{F}$.

However, as mentioned in Chapter 2, iterative methods can find an approximate solu-

tion to the linear system (3.40) by checking the residual of the linear system (3.40) iteratively. Additionally, one advantage of the use of an iterative solver is that the matrix-vector product $\mathbf{C}\boldsymbol{x}$ can be performed cheaply and quickly by evaluating the value $\mathbf{C}\boldsymbol{x}_G = \Phi_G \boldsymbol{x}_G - \sum_{i=0}^{n} \begin{bmatrix} \mathbf{F}^\top \mathbf{P}_i^{-\top} \Theta_i \mathbf{P}_i^{-1} \mathbf{F} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_G$ from right to left, which implies that it is not necessary to obtain the Schur complement matrix $\mathbf{C}$ in advance. Most important of all, matrix $\mathbf{F}$ is extremely sparse since it only consists of a bus-generator incidence matrix. Hence, calculating the matrix-vector products with $\mathbf{F}$ and $\mathbf{F}^\top$ can become very cheap and efficient when forming the contribution $\mathbf{F}\mathbf{P}_i^{-T}\Theta_i\mathbf{P}_i^{-1}\mathbf{F}\boldsymbol{x}_G$.

Therefore, it is worth trying to abolish the Cholesky factorization of $\mathbf{C}$, and instead apply an iterative method to solve the Schur complement system (3.40) in our IPM implementations. Note that GMRES is the main iterative method implementation covered in this thesis.

As described in Chapter 2, in order to guarantee its numerical robustness and efficiency, it is crucial to choose a good preconditioner. Unfortunately, it is still something of a "black art" to choose a good and general preconditioner accepted by all the problems, and the best choice is still heavily problem dependent. There are few suggestions in the literature covering both the preconditioning issue and the SCOPF problem. Before discussing these suggestions, we firstly go back to review the equation (3.27), which is used to build the Schur complement matrix $\mathbf{C} = \Phi_G - \sum_{i=0}^{n} \mathbf{B}_i \Phi_i^{-1} \mathbf{B}_i^T$. Obviously, it requires a summation of $\mathbf{B}_i \Phi_i^{-1} \mathbf{B}_i^T$ throughout all contingency scenarios. An appealing idea is to construct the preconditioner in the same way, but using a subset of all contingencies as an approximation. This preconditioner is calculated by

$$\mathbf{M} := \Phi_G - \frac{n}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathbf{B}_i \Phi_i^{-1} \mathbf{B}_i^T, \tag{3.41}$$

where $\mathcal{M}$ is an index set of the contingency scenarios. If all the contingencies are used to build the preconditioner, the set $\mathcal{M}$ becomes exactly the same as the contingency set $\mathcal{C}$, and $\mathbf{M} = \mathbf{C}$. It then becomes meaningless to use a preconditioner technique since no work can be saved since $\mathbf{M}/\mathbf{C}$ is built explicitly. Therefore, a sensible idea is to use as few contingencies as possible while maintaining the robustness and efficiency.

In the literature, Qiu et al [57] use only one base case scenario, i.e the OPF case, to build the preconditioner, while Petra and Anitescu [56] use a random sample of contingencies. Qiu et al. demonstrate their implementation can successfully solve SCOPF test problems and also represent an accuracy test in which they claim their algorithm has high accuracy with less than 1% relative error, compared with the results from the benchmark Matpower which is a Matlab package for solving power flow problems. They present two test problems in their numerical experiments: the IEEE 57-bus system and a 3493-bus system, which contain 9 contingencies and 79 contingencies respectively. The errors shown in [57] are between -0.667% to 0.051%,

where the largest ratio corresponds to the base value of 45MW.

The authors' approach is adopted to build the preconditioner to solve the test problems from previous section, which are listed in Table 3.2. Note that the convergence tolerance is still given as 10^{-6}. The results are given in Table 3.5.

	Base Case Preconditioned		
Bus	Time(s)	Memory	Iters
3	<0.01s	5.2MB	8
26	0.252s	7.4MB	13

Table 3.5: Using base case to build preconditioner

It is unfortunate, but not surprising, that only the 3-bus system and the 26-bus system, which are the smallest problems among the examples, can be solved by GMRES within OOPS. Note that these two test problems contain only 2 contingencies and 40 contingencies respectively. Despite the fact that these two small SCOPF problems can be solved by using an IPM with the GMRES iterative solver, numerical problems can still occur if the tolerance of GMRES is too small or too large. If the tolerance is too large, GMRES returns a point far away from the central path, which can mislead the IPM iterates into becoming infeasible. On the other hand, once the tolerance is too small, numerical trouble occurs when close to convergence, as GMRES stagnates and no improvement can be achieved. This issue is also related to question of which convergence scheme for GMRES is applied, which will be discussed later. As a result, it is no wonder that GMRES fails to converge to the required tolerance in the large test problems with more scenarios. Therefore, it is necessary to find a more robust preconditioner for DC-SCOPF problems.

On the other hand, we observe that certain contingency scenarios always introduce very large entries into the Schur complement matrix $\mathbf{C}$. In other words, certain contingency contributions $\mathbf{F}^\top \mathbf{P}_i^{-\top} \Theta_i \mathbf{P}_i^{-1} \mathbf{F}$ are much larger than those for other contingency contributions. In particular, these large entries originally come from one/few large theta values in the diagonal block matrix $\Theta_i = \mathbf{X}_i^{-1} \mathbf{Z}_i$. In some cases, this theta value $\boldsymbol{x}_j^{-1} \boldsymbol{z}_j$ can be of the order 10^6 times larger than the others in the same contingency case. In fact, this large value is caused by the extremely small slack variable $\boldsymbol{x}_j$, which is the gap between its corresponding flow variable and its flow limit.

On the other hand, observation from the numerical experiment also indicates that this specific contingency scenario typically contains these kinds of small slacks through the whole IPM process, despite the fact that the theta factor is changing with the iterates. Since small slacks imply potentially active constraints or active variable bounds, these contingencies can be claimed as *"active contingencies"*. Definitely, a natural idea is that these active contingencies are crucial and it is sensible to include them in the preconditioner.

In what follows, two main approaches to build the preconditioner set $\mathcal{M}$ are suggested,

based on different methods for including the active contingency scenario. Note that these two methods are dynamic methods as they need to detect the active contingencies in the meantime of applying the IPM solver.

- **Cumulative Approach:** Starting from building $\mathcal{M}$ by one/several given scenario(s), the problem is solved by IPM and the active contingencies are dynamically added into the set $\mathcal{M}$. As its name suggests, the number of components of $\mathcal{M}$ is at least non-decreasing and could potentially grow in pace with IPM iterations.

- **Aggressive Approach:** Unlike the cumulative approach, this method re-establishes the index set $\mathcal{M}$ at the beginning of each IPM iteration. The approach only detects the active scenarios for the previous iterate and uses them to make the set $\mathcal{M}$. Preventing active scenarios from being detected in a particular IPM step, the base-case scenario is set as active for that iteration.

After obtaining the index set $\mathcal{M}$, the preconditioner $\mathbf{M}$ is calculated by equation (3.41).

Last but not least, it is worth mentioning that in the preconditioned GMRES framework, the linear system (3.40) is replaced by

$$\mathbf{M}^{-1}\mathbf{C}\boldsymbol{x}_G = \mathbf{M}^{-1}(\mathbf{b}_G - \sum_{i=0}^{n} \mathbf{B}_i \boldsymbol{z}_i). \tag{3.42}$$

Additionally, the $\mathbf{LDL}^\top$ factorization is still required to perform $\mathbf{M} = \mathbf{L}_M\mathbf{D}_M\mathbf{L}_M^\top$, in order to evaluate $\mathbf{M}^{-1}(\mathbf{b}_G - \sum_{i=0}^{n} \mathbf{B}_i \boldsymbol{z}_i)$ and $\mathbf{M}^{-1}\mathbf{C}\boldsymbol{x}_G$. However, the back-solve operation in $\mathbf{M}^{-1}(\mathbf{b}_G - \sum_{i=0}^{n} \mathbf{B}_i \boldsymbol{z}_i)$ is only required to be performed once in the beginning of GMRES algorithm, while in each GMRES iteration, only one additive back-solve operation is required to evaluate $\mathbf{M}^{-1}\mathbf{C}\boldsymbol{x}_G$.

3.3.1 Numerical Experiments

To test our new approaches, we apply preconditioned GMRES in Method B to solve previous test problems, i.e. test problems listed in Table 3.2. Note that now Method B saves the partial solutions for a contingency if and only if this contingency is detected as active at some iteration. These saved data will be used later to build the Schur complement contribution in $\mathbf{B}$.

The base case is used to build the initial preconditioner. The criteria used to detect whether a contingency is active or not is to check if the ratio between the largest theta factor and the smallest one for each contingency is too large. That is, if the following inequality (3.43) holds for a certain contingency i with the given parameter ρ,

$$\frac{\max\{X_i^{-1}Z_i e\}}{\min\{X_i^{-1}Z_i e\}} \geq \rho, \tag{3.43}$$

then this contingency is judged to be an active contingency. In the following numerical experiments, the parameter ρ is given as 10^6.

The corresponding numerical results are shown in Table 3.6, where the second column reports the total number of scenarios included in the model. Note that this number is equal to the number of contingencies plus one from the basic OPF case. The columns named as 'FinSce' denote the number of active scenarios found from these two approaches respectively at the convergence of the IPM process. The last column denotes the largest number of active contingencies found by an aggressive approach at one particular IPM step.

		Cumulative			Aggressive			
Bus	NoSce	Time(s)	Iters	FinSce	Time(s)	Iters	FinSce	MaxSce
3	3	<0.1	8	2	<0.1	8	1	1
26	41	0.27	13	2	0.27	13	1	1
56	80	1.26	15	6	1.25	15	4	4
100	181	9.09	20	7	9.08	20	5	6
200	371	50.63	28	9	50.16	28	6	7
300	566	205.79	39	20	234.70	43	16	19
400	761	523.65	55	20	529.48	56	11	16
500	956	823.95	46	25	823.91	47	11	21

Table 3.6: GMRES with different methods to build preconditioners

As shown in Table 3.6, all the test problems can be solved by the preconditioner defined by our new approaches. Most of all, the number of active scenarios is much smaller than the total number of scenarios. For the largest test problem which contains 956 contingency scenarios, it only needs less than 3% of the scenarios to build the preconditioner and solves the Schur complement linear system successfully.

However, it becomes much more difficult to predict the whole numerical behavior of the IPM process once iterative methods are involved. In fact, the convergence tolerance in GMRES plays a very important role in these methods. Since the GMRES implementation in Matlab [51] is not specialized for solving linear systems arising in the IPM framework, its default termination criteria may not perform well in the structure-exploiting IPM implementation. For instance, GMRES now is applied to solve the linear system (3.40), $\mathbf{C}\mathbf{x}_G = \mathbf{b}_{rhs}$ for short. The termination criteria used in Matlab contains a relative tolerance $tol \times \|\mathbf{b}_{rhs}\|_2$, where tol is a given parameter. That is, the termination criteria depends on the current IPM iterate, since the vector $\mathbf{b}_{rhs}$ is changing with the IPM iterates. The GMRES implementation in Matlab tries to check whether or not the residual of the system (3.40) from the current GMRES iterate is smaller than this relative tolerance. If it detects one such iterate, it will return this iterate as the approximate solution to the system (3.40).

Unfortunately, the relative tolerance may be large due to the term $\mathbf{b}_{rhs}$. An accepted approximate solution from such a termination criteria may be not accurate enough for later use in the IPM process. In fact, from our experiments, if the approximate solution to the Schur

complement system (3.40) is not accurate enough (its corresponding Euclidean length of residual is larger than 10^{-8}), numerical problems may occur and affect IPM convergence. The worst case can break off the IPM process once the new iterate is kicked out by the feasibility issue, which returns the false conclusion that the problem is infeasible. For example, the problems with 26 and 56 buses would both be classified as infeasible problems if the convergence tolerance in GMRES is 10^{-6}. A sensible idea is to decrease the convergence tolerance proportionally to the barrier parameter μ, since the solution error may be relatively larger when μ is large. Contrarily, when μ is small in the last few IPM steps, the convergence tolerance of GMRES should be tighter in order to achieve the convergence of IPM iterates. The question about how to find a suitable convergence tolerance regarding μ in the structure-exploiting IPM is left for future work.

Despite our GMRES implementation sharing the same spirit as Matlab [51], the termination criteria of GMRES is modified in order to achieve a better performance in the IPM process. The decision is made to abandon the relative tolerance in GMRES and specify the tolerance of GMRES to be fixed at 10^{-16}. Note that this is the termination criteria applied for GMRES in all the numerical tests shown in Table 3.6.

So far this small GMRES tolerance can ensure convergences of all the test problems; however, some accompanying issues are worth further discussion.

Firstly, it is obvious that small tolerance implies more GMRES steps, leading to a more accurate approximation of the solution to the linear system (3.40). As mentioned in Chapter 2, GMRES requires all the previous information to be recorded, and consequently increases the demand for storage. Hence, small GMRES tolerance means more computational expense, in terms of both solution time and storage requirement.

On the other hand, small tolerance also implies that GMRES may stagnate, which means its two consecutive GMRES iterates are exactly the same in some particular IPM iterations. Observations from the numerical tests suggest that this issue can happen even in some earlier IPM steps. The good news is that it does not happen in the last few steps, which are the major concerns of the convergence of IPM. It is worth mentioning that this issue is also related to the preconditioning techniques and the regularization techniques. If only the base case is used to build the preconditioner, and such strict tolerance is required, none of the test problems can reach convergence of IPM, even for the smallest two test problems. They meet some numerical difficulties when close to convergence, while GMRES stagnates and the output approximation cannot achieve any improvement for IPM iterates. As a result, IPM iteration will continue until the maximum number of iterations is reached. We observe that this numerical difficulty is relevant to the bad preconditioner applied to (3.42), since it requires some relatively large regularization to perform the factorization. Anyhow, this is also a very good comparison for showing the strengths of our new schemes.

Last but not least, from such a strict tolerance, the approximate solution to the linear

system (3.40) can be regarded as the most accurate one obtained from GMRES. Therefore, it should also be a good standard to focus on the behaviors of different preconditioner schemes.

We also present Table 3.7 to demonstrate the comparison between the default method, the most successful method from previous sections, and these two iterative approaches.

	Default Method		Method B	
buses	time(s)	memory	time(s)	memory
3	<0.1	5.2MB	<0.1	5.2MB
26	0.21	7.6MB	0.17	7.5MB
56	1.00	14.3MB	0.77	14.1MB
100	9.02	54.6MB	6.16	53.1MB
200	65.97	220MB	45.23	244MB
300	251.70	531MB	177.39	667MB
400	955.32	985MB	655.50	1380MB
500	1552.80	1593MB	1195.77	2467MB
	Cumulative		Aggressive	
buses	time(s)	memory	time(s)	memory
3	<0.01	5.2MB	<0.01	5.2MB
26	0.27	7.4MB	0.27	7.4MB
56	1.26	13.5MB	1.25	13.5MB
100	9.09	43.6MB	9.08	43.6MB
200	50.63	163MB	50.16	163MB
300	205.79	387MB	234.70	387MB
400	523.65	715MB	529.48	716MB
500	823.95	1163MB	823.91	1164MB

Table 3.7: Comparison among three methods

From Table 3.7, it is obvious to conclude that there are advantages in using an iterative method rather than a direct method to solve the Schur complement linear system (3.40).

A quick observation is that their memory requirements are always less than the default method and Method B from the previous section. This is due to the fact that, with the iterative method, only partial solutions from active contingencies are recorded, which can be reused to build the Schur complement type preconditioner $\mathbf{M}$ (3.41) in the later IPM steps. In addition, since the implementation of the iterative method based IPM approach is an improvement upon Method B, it only applies a factorization technique to the smaller constant matrices. Hence its storage requirement should be the least. Note that both the memory usages of the iterative method based approaches are less than half of that of Method B.

Additionally, observe that both the cumulative approach and the aggressive approach need roughly the same amount of memory. As mentioned previously, some particular contingencies are active throughout the entire process. Both of these two preconditioning approaches can detect those contingencies sooner or later, and then the partial data for these contingencies will be saved. Therefore, these two methods require roughly the same amount of memory. Note that, for the largest two test problems, the aggressive method requires a little bit more memory. This tiny difference is due to the fact that the aggressive method would detect some other active

contingencies which cannot be found by the cumulative method.

Besides the advantages obtained from the computational storage side, the results in Table 3.7 also indicate that both of the iterative method based approaches perform much faster than the default method. Specifically around 50% of the computational time can be saved when referring to the largest test problems. Generally, the performances of these two approaches are also better than the performance of Method B for the large test problems.

Observed from the numerical experiments, incorporating GMRES into an interior point method may bring some tolerance issues into consideration. However, it can definitely speed up the solution process for DC-SCOPF problems. Additionally, it requires less storage capacity even though partial solutions are recorded to build the preconditioner. As mentioned in Chapter 2, GMRES is robust since it can minimize the residual from each iterate. The downside is that it requires additional computational cost due to its long-term recurrence. It is interesting to see if GMRES can be replaced by a short-term recurrence method, e.g. BiCGStab, in an interior point algorithm, and it can be left as a future work.

Chapter 4

Solving AC-SCOPF Problem

The underlying work in this chapter is to solve the AC-SCOPF problem. As described in Chapter 2, this problem is a nonlinear programming (NLP) problem. Similar to the linear DC-SCOPF case, the structure-exploiting interior point method is the main approach applied to solve this problem.

Firstly, the IPM implementation of OOPS is extended to a nonlinear programming solver. The distinguishing structure-exploiting feature of OOPS as well as the dynamic regularization method are retained. The nonlinear interior point algorithm implements the framework in the same spirit as IPOPT [70], which consists of a primal-dual interior point method with filter line search strategy and aims to find a local solution of a nonlinear nonconvex problem.

It is worth mentioning that this nonlinear IPM algorithm has been firstly implemented and tested in the MATLAB [51] environment without structure-exploiting strategy. It can successfully solve more than 70% of the nonlinear problems in CUTEr [38] which is a package containing a set of constrained and unconstrained test problems. After ensuring the MATLAB implementation is robust enough, the whole algorithm is coded into the structure-exploiting variant in OOPS, with programming language C++.

In the structure-exploiting implementation, a new regularization method is applied, which integrates the dynamic regularization technique with the iterative regularization suggested in IPOPT. Furthermore, a new iterative solution technique for structure-exploiting IPM is introduced, it can successfully identify the necessary scenarios dynamically during the IPM solution process.

For instance, this new scenario-generation-based IPM can integrate the iterative solution technique for SCOPF problems, which is presented in Figure 2.2, within IPM's iterative process. In other words, the contingency analysis step has been evaluated between IPM steps. As a result, solving SCOPF problem just requires one call of the IPM solver, and most important of all, it can start with an OPF problem.

This chapter is structured as follows: In Section 4.1, several critical issues about the

structure-exploiting IPM implementation for NLP are emphasized. Firstly a brief introduction of the necessary linear algebra of IPM for NLP is discussed, followed by a summary of the line search strategy and the IPM implementation in IPOPT. Additionally, the mixed regularization technique is explained, and a robust way to deal with the inequality constraint is also discussed in the end of Section 4.1. In the second part of this chapter, Section 4.2, the scenario-generation-based interior point algorithm is described A theoretical analysis of the warm-start strategy applied in this new approach is also represented in this section. At the end of this chapter, the corresponding numerical tests of the SCOPF problems are given, as well as some small examples for clarifying the process.

4.1 Structure-Exploiting IPM for NLP

4.1.1 Linear Algebra of IPM for NLP

As mentioned before, the powerful tool of primal-dual IPM has been chosen to solve the nonlinear problems appearing in this thesis. Similar to the process presented in Chapter 2, the logarithmic barrier function associated with the nonlinear problem (2.31) can be obtained as

$$\begin{aligned} \min: \quad & \phi_\mu(x) := f(x) - \mu \sum \ln x \\ s.t: \quad & g(x) = 0, \end{aligned} \tag{4.1}$$

where μ is the logarithmic barrier parameter. While μ is forced to approach zero in the IPM process, the solution of (4.1) converges to the solution of (2.31). Additionally, an assumption is made that the gradients of the constraint function g are linearly independent at every point x, i.e. the constraint Jacobian $\bigtriangledown g(x)^\top$ has full rank.

The solutions of problem (4.1) can also be characterized by solving its KKT conditions (2.33)-(2.36) with Newton's method. The Newton direction $(\Delta x, \Delta y, \Delta s)$ can be obtained by solving the following linear system:

$$\begin{bmatrix} \mathbf{-W} & \mathbf{A}^\top & \mathbf{I} \\ \mathbf{A} & 0 & 0 \\ \mathbf{Z} & 0 & \mathbf{X} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} \bigtriangledown f - \mathbf{A}^\top y - z \\ -g(x) \\ \mu e - \mathbf{XZ}e \end{bmatrix}, \tag{4.2}$$

where $\mathbf{A}^\top = \bigtriangledown g(x)$ and $\mathbf{W}$ denotes the Hessian of the Lagrangian function (2.32) respecting to the argument x, i.e. $\bigtriangledown^2_{xx} L(x, y, z)$.

Similarly to the linear case, vector Δz can be substituted by

$$\Delta z = \mu \mathbf{X}^{-1} e - z - \mathbf{X}^{-1}\mathbf{Z}\Delta x, \tag{4.3}$$

and the corresponding augmented system can be represented as

$$\begin{bmatrix} -\mathbf{W}-\Theta & \mathbf{A}^\top \\ \mathbf{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \bigtriangledown\phi_\mu - \mathbf{A}^\top y \\ -g(x) \end{bmatrix}. \tag{4.4}$$

where $\Theta := \mathbf{X}^{-1}\mathbf{Z}$.

Contrarily, in a common NLP problem, matrix $\mathbf{W}$ is a nonzero matrix. In a structured problem where the Jacobian matrix $\mathbf{A}$ has a block angular structure of Figure 3.1, the Lagrangian Hessian $\mathbf{W}$ can also be reordered to have the block structure as

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & & & & \mathbf{W}_{1,G}^\top \\ & \mathbf{W}_2 & & & \mathbf{W}_{2,G}^\top \\ & & \ddots & & \vdots \\ & & & \mathbf{W}_n & \mathbf{W}_{n,G}^\top \\ \mathbf{W}_{1,G} & \mathbf{W}_{2,G} & \cdots & \mathbf{W}_{n,G} & \mathbf{W}_G \end{pmatrix}, \tag{4.5}$$

where symmetric submatrix $\mathbf{W}_{i,j} = \bigtriangledown^2_{x_i x_j} L$ and $\mathbf{W}_i = \bigtriangledown^2_{x_i x_i} L$, if the Hessian of the objective function also has such form. Note that for all $i \neq j$ and $i, j = 1, \ldots, n$, the terms $\mathbf{W}_{i,j}$ are zero matrices due to the fact that there is no explicit constraint linking local variables x_i and x_j together. The only exception is the term $\mathbf{W}_{i,G}$ since x_G is the global variable which appears in the right-hand-side border column of Figure 3.1.

Furthermore, after reordering constraints and variables, the augmented system matrix $\Phi := \begin{bmatrix} -\mathbf{W}-\Theta & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$ can also be expressed as a symmetric double bordered block diagonal matrix as follows:

$$\Phi = \begin{pmatrix} \Phi_1 & & & & \mathbf{B}_1^\top \\ & \Phi_2 & & & \mathbf{B}_2^\top \\ & & \ddots & & \vdots \\ & & & \Phi_n & \mathbf{B}_n^\top \\ \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_n & \Phi_G \end{pmatrix}, \tag{4.6}$$

where

$$\Phi_i := \begin{bmatrix} -\mathbf{W}_i - \Theta_i & \mathbf{A}_i^\top \\ \mathbf{A}_i & \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\bigtriangledown^2_{x_i x_i} L - \Theta_i & \bigtriangledown_{x_i} g_i \\ \bigtriangledown_{x_i} g_i^\top & \mathbf{0} \end{bmatrix}$$

and

$$\mathbf{B}_i^\top := \begin{bmatrix} -\bigtriangledown^2_{x_G x_i} L \\ \bigtriangledown_{x_G} g_i^\top \end{bmatrix}, i = 1, \ldots, n$$

Hence, the structure-exploiting idea [37] can also be applied to solve this kind of structured problem for NLP. The only difficulty is that the interior point logic presented in the previous

chapter is not strong enough to solve the nonlinear problem, since the new iterates may not have sufficient decrease in the objective function and converge to a wrong solution (example can be found in [54]). Most of all, regularization is another main difficulty, as each block Φ_i may be no longer quasi-definite even after small diagonal perturbations are added to it. This is due to the fact that the top-left term $\mathbf{W}_i + \Theta_i$ does not have to be positive definite because the Hessian $\mathbf{W}_i$ is derived from a nonlinear nonconvex problem.

Last but not least, although the above discussions are applied for a certain structured NLP problem, it is worth mentioning that the AC-SCOPF problem presented in Chapter 2 has the same especial structure as mentioned. In the following sections, only the generic form of NLP will be presented, however, all the results can be applied to the AC-SCOPF problem.

4.1.2 Filter Line Search IPM

In order to achieve the global convergence property, which means that the algorithm can converge with an arbitrary initial point, the search direction and the step size from the current iterate should be guided by some special techniques. Generally, one of the most fundamental approaches for finding an acceptable iterate from the current point, is the backtracking line search method [54]. The main concept of the backtracking line search method is that it uses a trial and error method to test a sequence of candidates with decreasing step sizes along the given search direction. Backtracking line search terminates after a suitable step size is detected, while some extra requirements are satisfied.

One of the essential requirements of the backtracking line search method is the so-called *Armijo Condition* [54], which enforces a sufficient reduction in the objective function. Since IPM algorithms actually solve barrier function (4.1), for convenience, the Armijo condition is stated for the barrier function in the thesis, as

$$\phi_\mu(x_k + \alpha_k \Delta x_k) \leq \phi_\mu(x_k) + \eta \alpha_k \bigtriangledown \phi_\mu(x_k)^\top \Delta x_k, \tag{4.7}$$

where $\eta \in (0, 1)$ is a given constant; x_k is the current iterate; Δx_k is the descent step and α_k is the step size.

The Armijo condition indicates that an acceptable step size α_k from the current point must result in a sufficient reduction of the objective at the new iterate $(x_k + \alpha_k \Delta x_k)$. Defining $r(\alpha_k)$ as the right hand side of (4.7), it is trivial that function $r(\alpha_k)$ is a linear function with respect to the step size α_k. The Armijo condition can be illustrated in Figure 4.1, where inequality $\phi_\mu(x_k + \alpha_k \Delta x_k) \leq r(\alpha_k)$ defines an acceptable region for the new step size.

Figure 4.1 also implies that the interior point algorithm specialized for LP can fail to find a decent step since the new iterate may be out of the acceptable region.

Besides the Armijo condition, in practice, a line search algorithm can be more complicated while more termination conditions get involved. Wächter and Biegler [69] suggest a line search

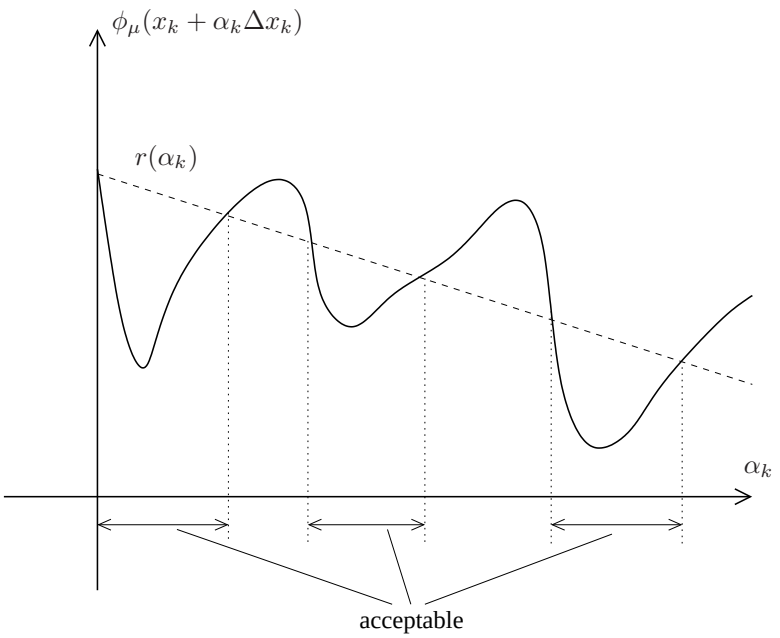

Figure 4.1: Armijo condition.

method with filter techniques, which can guarantee the global convergence of their interior point implementation IPOPT [70] for nonlinear programming.

The underlying concept of their filter techniques consists of a two-dimensional filter, $\mathcal{F} := \{(\phi, \zeta) \in \mathbb{R}^2 \mid \zeta \geq 0\}$, where ϕ and ζ are used to measure the objective ϕ_μ and constraint violation $\|g\|$ in Euclidean norm, respectively.

This filter is initialized as

$$\mathcal{F}_0 := \{(\phi, \zeta) \in \mathbb{R}^2 \mid \zeta \geq \zeta^{max}\} \tag{4.8}$$

for a given parameter ζ^{max}, and can be updated by scheme

$$\mathcal{F}_{k+1} := \mathcal{F}_k \cup \{(\phi, \zeta) \in \mathbb{R}^2 \mid \phi \geq \phi_\mu(x_k) - \gamma_\phi \|g(x_k)\|, \quad \zeta \geq (1 - \gamma_\zeta)\|g(x_k)\|\} \tag{4.9}$$

with parameter $\gamma_\phi, \gamma_\zeta \in (0, 1)$ after the kth iteration. Then filter $\mathcal{F}_{k+1}$ is used in the backtracking line search during the next iteration. It would reject a trial point $x_k + \alpha_k \Delta x_k$, if $(\phi_\mu(x_k + \alpha_k \Delta x_k),\ \|g(x_k + \alpha_k \Delta x_k)\|) \in \mathcal{F}_{k+1}$. As a result, the filter technique is used to guarantee that the new iterate can improve the objective function or the constraint violation by a certain required level.

The outline of the filter line search primal dual interior point framework of [70] is summarized in Algorithm 4.1.

Note that the above discussion is a basic idea of the line search method applied in IPOPT [70]. In fact, the line search implementation used in IPOPT is much more sophisticated. It is

Algorithm 4.1: Summary of IPOPT Framework

Step 1. Initialization

Step 2. Check convergence for the original problem. If it converges, algorithm TERMINATES.

Step 3. If necessary, update the barrier parameter and re-initialize the filter for the new barrier problem.

Step 4. Compute a search direction with regularization for correcting inertia.

Step 5. Backtracking line search with filter techniques. If line search method fails, go to Step 8.

Step 6. Update the dual variables.

Step 7. Augment the filter if necessary.

Step 8. Apply feasibility restoration phase

Step 9. Go back to Step 2 and continue with the next iteration.

worth mentioning that the filter line search in Step 5. is aimed to find an acceptable step size α_k for primal variable x_k and Lagrangian multiplier y_k. The backtracking line search starts from the largest candidate α_k^{max} which is calculated by the fraction-to-the-boundary rule:

$$\alpha_k^{max} = \max\{\ \alpha \in (0,1] \mid x_k + \alpha \Delta x_k \geq (1-\kappa) x_k\ \}, \tag{4.10}$$

where parameter $\kappa \in (0,1)$. Additionally, unlike the generic primal-dual Algorithm 2.1, a different step size α_k^z for the dual variable z_k is allowed in IPOPT. It is obtained by

$$\alpha_k^z = \max\{\ \alpha \in (0,1] \mid z_k + \alpha \Delta z_k \geq (1-\kappa) z_k\ \}. \tag{4.11}$$

in Step 6. The use of distinct step size for variable z_k makes the algorithm more efficient since α_k^z would not be restricted by the step size α_k even after the line search [70]. More details about the backtracking line search method in IPOPT implementation can be seen in [70].

Last but not least, the main use of the feasibility restoration phase (Step 8) is to overcome the numerical difficulty appearing in the solution process. As mentioned in [70], this numerical difficulty can be caused by the very tiny step size produced by the filter or the very ill-conditioned matrix which cannot be factorized successfully. Although this is a key of the robustness of the implementation, we observe that feasibility restoration phase has never been called by the solver when the test problem is feasible and a certain 'good' reformulation of the inequality constraints has been applied. Details about the reformulation of the inequality constraints are discussed in section 4.1.4.

4.1.3 Mixed Regularization Policy

As mentioned, the feasibility restoration phase is abandoned in the structure-exploiting IPM implementation. As an alternative, some other heuristic remedies are applied to deal with the numerical difficulties. Numerical experiments, which are presented at the end of this chapter, show that this heuristic approach works efficiently for the SCOPF problem.

The major difference is that we apply Altman and Gondzio's dynamic regularization [9], which computes the regularization and the numerical factorization simultaneously. The advantage of applying dynamic regularization is that the related matrix can become strongly factorizable without 2×2 pivots and hence it can improve the efficiency of the factorization. The dynamic regularization has another significant purpose, namely to detect some numerical problems. It can keep all the pivots in the same order of magnitude by forcing an additive small perturbation to the diagonal elements. As a result, it reduces the numerical problems and makes the factorization more stable.

On the other hand, the filter line search strategy in [70] requires that the matrix $\mathbf{W}_i + \Theta_i$, projected onto the null space of the constraint Jacobian $\mathbf{A}_i^\top$, is positive definite, in order to ensure global convergence. However, it is very expensive to compute the eigenvalues after building such projected matrix explicitly. Instead, a necessary condition states that if such matrix is positive definite, the iteration matrix Φ_i has *inertia* $(m, n, 0)$, that is, matrix Φ_i has exactly n positive, m negative, and no zero eigenvalues [54]. Furthermore, due to the Sylvester's law, this inertia numbers can be found by checking the signs of the pivots in a $\mathbf{LDL}^\top$ factorization, which is much cheaper and easier to compute.

Hence, even though having a inertia of $(m, n, 0)$ is a necessary condition, the IPM implementation in [70] applies a heuristic inertia correction regularization method, *IC* regularization for short, to correct the inertia of Φ_i if necessary, in order to ensure the perturbed iterate matrix have the chance to be positive definite. Additionally, they have two ways to update the IC regularization, depending on whether or not this is the first time to correct inertia. In the structure-exploiting IPM implementation, this IC regularization is retained, in order to correct the inertia of the iterate matrix.

As a result, there are two regularization methods in the structure-exploiting IPM algorithm, as one used for correcting inertia, and the other one for enhancing stability of the factorization. The two types of regularization are merged in one big factorization framework, where the corresponding regularized augmented system matrix Φ^{reg} now can be stated as follows:

$$\Phi^{reg} = \begin{bmatrix} -\mathbf{X}^{-1}\mathbf{Z} - \mathbf{R}_p^{IC} - (\mathbf{R}_p) & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{R}_d^{IC} + (\mathbf{R}_d) \end{bmatrix} \tag{4.12}$$

where perturbation $\mathbf{R}_p$ and $\mathbf{R}_d$ are the dynamic regularizations, and $\mathbf{R}_p^{IC}$ and $\mathbf{R}_d^{IC}$ are the inertia correction regularizations. Note that it is difficult to predict $\mathbf{R}_p$ and $\mathbf{R}_d$ beforehand and

they are dynamically obtained during the factorization process.

The overall integrated regularization algorithm in the structure-exploiting IPM implementation is presented in Algorithm 4.2, and also is illustrated in the flow chart of Figure 4.2. Inspired by the IC regularization used in IPOPT, in Step 4.1 and Step 4.2, different schemes are applied to compute the values of the IC regularizations $\mathbf{R}_p^{IC}$ and $\mathbf{R}_d^{IC}$, based on if previous IC regularizations are available. The use of referencing the latest IC regularization, $\mathbf{R}_p^{last}$ and $\mathbf{R}_d^{last}$, from a successive iteration is always suggested rather than computing the regularization from scratch. This is due to the assumption that the minimum successive IC regularization have the same order of magnitude between IPM iterations. The heuristic approach would try to test a smaller or same regularization for current IPM step from the historical information. That is, it would compute $\mathbf{R}_p^{IC} \leftarrow c_1\mathbf{R}_p^{last}$ and $\mathbf{R}_d^{IC} \leftarrow c_1\mathbf{R}_d^{last}$, where $0 \leq c_1 \leq 1$, for current iterate to see if the modified augmented system has a correct inertia. Additionally, once current augmented system requires further larger regularizations to correct its inertia: if $\mathbf{R}_p^{last}$ and $\mathbf{R}_d^{last}$ are available, algorithm would like to increase the regularization terms by a multiplier $c_3 > 1$; otherwise it would try a larger or same multiplier $c_2 \geq c_3$ to increase the regularization.

Algorithm 4.2: Mixed Regularization Algorithm

Step Given constant $\hat{\mathbf{R}}_{p,init}^{IC} > 0$, $\hat{\mathbf{R}}_{d,init}^{IC} > 0$, $0 < c_1 \leq 1$ and $c_2 \geq c_3 > 1$

Step 1. Initialization: Set $\mathbf{R}_p^{IC} \leftarrow 0$ and $\mathbf{R}_d^{IC} \leftarrow 0$.

Step 2. For each diagonal block matrix Φ_i^{reg}, $i = 0, 1, \ldots, n$:

2.1. Factorize the modified matrix Φ_i^{reg} with dynamic regularization. Obtain $\mathbf{R}_{p,i}$ and $\mathbf{R}_{d,i}$

2.2. Check if the inertia of matrix Φ_i^{reg} is correct. If not, break the loop and go to Step 4.

Step 3. Compute the Schur complement $\mathbf{C} = \Phi_G - \sum_{i=1}^{n} \mathbf{B}_i\Phi_i^{-1}\mathbf{B}_i^{\top}$ Attempt to factorize matrix $\mathbf{C}$ with dynamic regularization. If its inertia is correct, set $\mathbf{R}_p^{last} \leftarrow \mathbf{R}_p^{IC}$ and $\mathbf{R}_d^{last} \leftarrow \mathbf{R}_d^{IC}$, and algorithm STOPS.

Step 4. If this is the first time to apply IC regularization for current IPM step, go to Step 4.1, otherwise continue with Step 4.2:

4.1. If $\mathbf{R}_p^{last} = 0$ and $\mathbf{R}_d^{last} = 0$, set $\mathbf{R}_p^{IC} \leftarrow \hat{\mathbf{R}}_{p,init}^{IC}$ and $\mathbf{R}_d^{IC} \leftarrow \hat{\mathbf{R}}_{d,init}^{IC}$. Otherwise set $\mathbf{R}_p^{IC} \leftarrow c_1\mathbf{R}_p^{last}$ and $\mathbf{R}_d^{IC} \leftarrow c_1\mathbf{R}_d^{last}$.

4.2. If $\mathbf{R}_p^{last} = 0$ and $\mathbf{R}_d^{last} = 0$, set $\mathbf{R}_p^{IC} \leftarrow c_2\mathbf{R}_p^{IC}$ and $\mathbf{R}_d^{IC} \leftarrow c_2\mathbf{R}_d^{IC}$. Otherwise set $\mathbf{R}_p^{IC} \leftarrow c_3\mathbf{R}_p^{IC}$ and $\mathbf{R}_d^{IC} \leftarrow c_3\mathbf{R}_d^{IC}$.

Step 5. Go back to Step 2 for another trial of factorization.

In a word, this integrated regularization technique presented in Algorithm 4.2 consists of

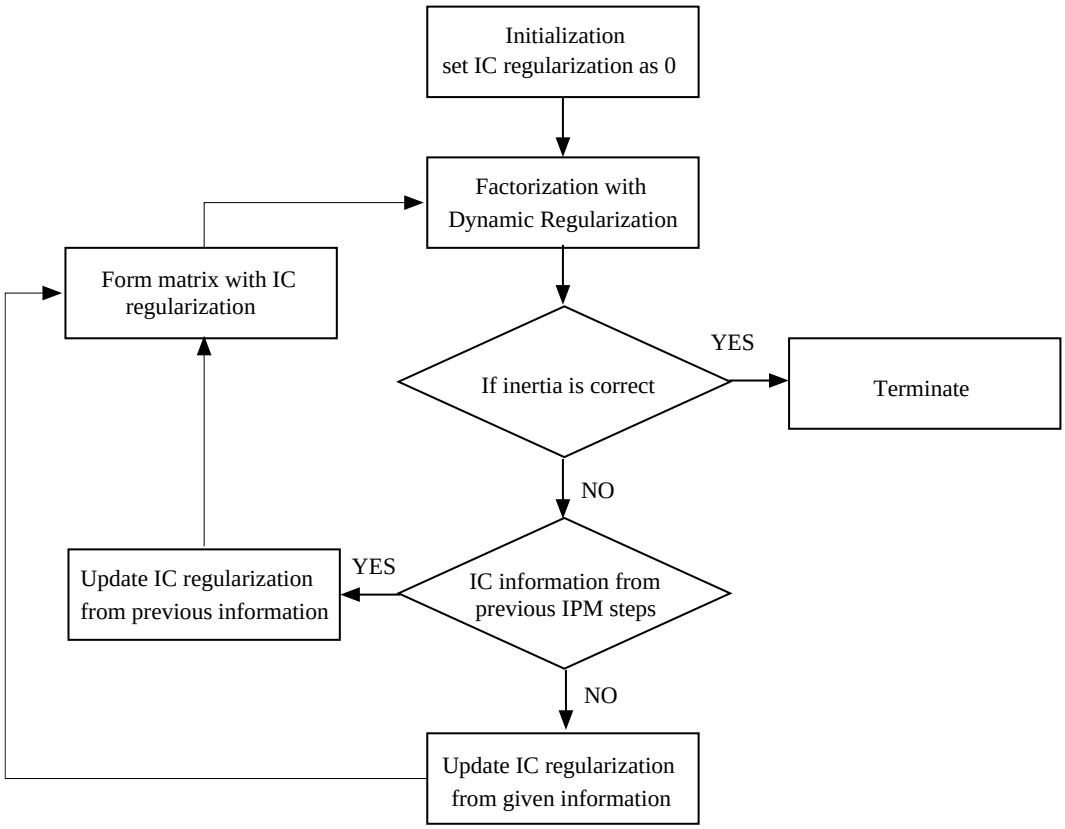

Figure 4.2: The framework of regularization and factorization in each IPM step.

a trial-and-error procedure to update the matrix, in order to get a correct inertia of the iterate matrix. Similar procedure has also been applied to find an appropriate regularization in the optimization code LOQO [66] and IPOPT [70]. On the other hand, it also contains dynamic regularization which makes the iterate matrices become quasi-definite. Consequently, the $\mathbf{LDL}^\top$ factorization can be applied without block pivots, which also implies that symbolic factorization is available in this framework.

In the end of this section, it is worth mentioning that if the dynamic regularization in Algorithm 4.2 is replaced by a linear system solver such as MA57 [85], which supports 2×2 pivots, the outcome algorithm is similar to the IC regularization scheme used in [70]. However, numerical experiments show that including a dynamic regularization may bring some advantages, e.g, the symbolic factorization. The most important advantage is that it can correct inertia slightly. That is, observed from some tests on SCOPF problems, the use of MA57 to solve linear system given by Φ_i lead to a Schur complement matrix $\mathbf{C}$ having wrong inertia. According to the trial-and-error procedure, this requires abandoning current iterate and restarting the factorization process with updated IC regularizations. Unfortunately, in the structure-exploiting implementation, this implies that the whole factorization should be recalculated from the first linear system given by Φ_0, which is quite time-consuming. Conversely, if a dynamic regularization technique is applied, the Schur complement matrix may have a correct inertia. As a

result, the process presented in Figure 4.2 can be faster than the use of a factorization with 2×2 pivots.

4.1.4 General Inequality Constraints

Recalling that our IPM implementation is a structure-exploiting variant of Wächter and Biegler's algorithm [70], which is designed to solve problems in standard form (2.31), namely with equality constraints $g(x) = 0$ and bound constraints $x \geq 0$. However, most problems contain inequality constraints since inequality constraints are more common in practice. We found the numerical performance varies a lot, due to the different methodologies applied to switch the inequality constraints to equality constraints.

Assume a general inequality constraint has the form

$$g_{LB} \leq g_I(x) \leq g_{UB}. \tag{4.13}$$

We reformulate (4.13) into the standard equality form in (2.31), by introducing slack variable s. There are two popular methods to achieve this goal. They are:

- $$g_{LB} \leq s \leq g_{UB}, \quad g_I(x) = s. \tag{4.14}$$

  Setting the constraint body equal to the slack and add bounds for the slack variable. That is, adding one slack variable $g_{LB} \leq s \leq g_{UB}$ and replacing the inequality constraint by an equality constraint $g_I(x) = s$. This method can be referred as a substitution method.

- $$g_I(x) + s_u = g_{UB}, \quad s_l \geq 0 \tag{4.15}$$
  $$g_I(x) - s_l = g_{LB}, \quad s_u \geq 0 \tag{4.16}$$

  Introducing slacks to fill in the gap between the constraint body and its boundaries. That is, adding two slack variables $s_l \geq 0$ and $s_u \geq 0$. Then the inequality constraint (4.13) can be replaced by two equality constraints $g_I(x) + s_u = g_{UB}$ and $g_I(x) - s_l = g_{LB}$.

Comparing these two methods, the second method produce one more equality constraint. It will lead to one more dual variable in the primal-dual barrier method and introduce a new column into the KKT system and the augmented system.

Moreover, after reformulating the inequality constraints, the bounds g_{LB} and g_{UB} may play different roles in the primal-dual interior point framework. With the first method, these bounds would appear in the barrier term in the objective function as $\sum \ln(s - g_{LB})$ and $\sum \ln(g_{UB} - s)$.

The second method implies that these terms become $\sum \ln(s_l - 0)$ and $\sum \ln(s_u - 0)$, while bounds g_{LB} and g_{UB} are only used to compute the residual after each IPM iteration.

Thus the first method changes the constraint boundary into a variable boundary but the second method does not. Additionally, the first method results in fewer equality constraints. Even though all the mathematical models are equivalent in theory, they may behave differently in practice since IPM implementations may have different criteria to deal with the constraint bound and the variable bound. That is, considering a variable boundary as a proper constraint or a boundary constraint may result in different treatment in IPM process. Two critical examples are presented in what follows:

- In principle, the generic primal-dual IPM framework relies on solving barrier problem (4.1) and it requires an 'interior point' from the feasible region. In other words, variable x must be strictly greater than its lower bound 0 to avoid the barrier term $\mu \sum \ln x$ becoming undefined.

  On the other hand, the equality constraint $g(x) = 0$ requires that the constraint body $g(x)$ should be exactly equal to its limit zero, in order to ensure this problem is feasible. Hence, for a feasible interior point, the constraint boundary should be strictly satisfied but the variable boundary can never be reached.

- If the equality constraints implicitly imply a component of x equal to its boundary, the algorithm might generate a sequence of iterates which lead the corresponding slack variable become too small and cause numerical problem. To overcome this tricky issue, different interior point algorithm may have its own heuristic methods as a remedy. In OOPS, Gondzio and Grothey [37] regularize the entries of $\mathbf{X}^{-1}\mathbf{Z}$ in order to normalize their magnitude to an acceptable level. Moreover, they have a heuristic method to push variables not too close to their own boundaries.

  Additionally, the implementation of IPOPT relaxes the variable bounds slightly before solving the problem, by adding small perturbation with the order of the termination tolerance. That is, for each component, modify the variable bounds by

  $$x_{LB} \leftarrow x_{LB} - tol * \max\{1, x_{LB}\}$$

  and

  $$x_{UB} \leftarrow x_{UB} + tol * \max\{1, x_{UB}\}$$

  with give termination tolerance tol, before starting the IPM main loop. These boundaries can be further relaxed whenever the variable approaches too close the current boundary.

In contrast to the above remedy approaches applied to the variable bounds, there is no further treatment for a constraint bound neither in OOPS nor in IPOPT. Therefore, the numerical

behavior of an interior point algorithm may vary, depending on whether or not treating variable boundary like a constraint boundary.

Numerical Experience of Inequality Reformulations for SCOPF

In the nonlinear OPF and SCOPF model, the inequality constraints (2.11) and (2.12) can be treated as boundaries of variables, while the line flow thermal limit (2.13) is represented as a proper inequality constraint. It is obvious that constraint (2.13) exclusively contains one side boundary, i.e. its right hand side. Therefore, no matter which method is used to add slack variables, it would require the addition of one additive slack variable and one equality constraint.

However, due to the above discussion, the difference between the constraint boundary and variable boundary approaches may not have the same effect in the IPM process. In fact, under the same parameter setting, the primal-dual implementation of IPOPT is more stable when slack variables are added using the substitution approach. It requires 43 IPM iterations to solve the IEEE 24-bus problem, while it needs 134 iterations to get the same solution if the second reformulation is applied to add slacks. What is more, the later reformulation needs to call the feasibility restoration phase in IPOPT, to capture the numerical difficulties caused by the ill-conditioned matrices.

According to the above reasons, for all the nonlinear test problems considered in this thesis, slack variables are suggested to be included in the model by substitution, in order to achieve a better numerical behavior in the filter line search IPM implementations. Note that the step of adding slack to make equality constraint has not been implemented yet. Hence slack variables need to be added into the model manually in the AMPL environment [31].

4.2 Interior Point Method with Scenario Generation

After solving the AC-SCOPF problem with full contingency scenarios, several inequality constraints are observed to be active constraints at the optimal solution. All these active inequality constraints belong to several scenarios. Hence, these scenarios can be referred to as *active scenarios*. Since we mainly focus on the SCOPF problems, the later use of *active contingencies* shares the same meaning as *active scenarios*.

As mentioned in Chapter 2, only a few contingencies from all the binding contingencies are essential when solving a SCOPF problem. From the power system operation point of view, these contingencies are critical for the operation of the system. Once all the active contingencies are taken into account in a security model, the solution to this reduced model would not produce any violated constraints when applying the contingency analysis step (see Figure 2.2).

These kind of contingencies have also been called *umbrella contingencies* in [13], since they cover all the effects caused by other contingencies.

From a mathematical point of view, each active scenario contains some active boundary constraints and some variables with nonzero reduced cost at the optimal solution to the full model. Therefore, if all the active contingencies can be obtained before solving the SCOPF problem, there is no need to solve a full SCOPF model containing all binding contingencies. The set of active contingencies, which is a subset of all required security contingencies, is sufficient to define a reduced SCOPF model whose solution is identical to that of the security model with all contingencies. The number of active contingencies cannot exceed the number of control variables, since this represents the number of degrees of freedom in the problem. It follows that a reduced SCOPF model may be much smaller than the full security model, and significant savings of computational effort are expected when solving a reduced model with active contingencies only.

Unfortunately, obtaining the active contingencies in advance appears to be a challenging problem. Even though some heuristic approaches which exploit the use of AC-OPF or DC-SCOPF as an approximation have been tested, none of them can successfully identify all the active contingencies for AC-SCOPF. Corresponding trials are presented in the next chapter.

Alternatively, we may ask whether the active scenarios can be identified during the IPM iterations dynamically. In order to answer this question, let us consider the DC case in Chapter 3. The preconditioned IPM algorithm for DC-SCOPF problem determines whether there are some variables approaching too close their boundary in each IPM iteration. This idea can also be considered as a scenario generation method by identifying potentially active constraints.

Provided the linear DC-SCOPF problem can be solved successfully, a new scenario-generation-based IPM algorithm is suggested to solve the AC-SCOPF problem. This new approach can dynamically detect potentially active scenarios during the IPM iterations, which is similar to the approach proposed for the DC case. However, the ideas behind these two approaches are totally different. The scenario generation approach used in DC-SCOPF only tries to find a better way to build the preconditioner for GMRES. Conversely, the new algorithm enhances the role of scenario generation as it slightly modifies the augmented system for the full security model and focuses on the behavior of the reduced SCOPF model defined by the detected active scenarios. The scenario generation method used in conjunction with warm-start techniques leads to an IPM implementation in which it is possible to start with a much smaller problem of the same structure, subsequently checking violated scenarios dynamically and adding the most violated scenarios to the model until no more violated scenarios can be detected.

This approach is inspired by the iterative solution technique in Figure 2.2, which integrates the 'contingency analysis' step into the 'solve OPF/SCOPF' step using warm-start techniques. Like the original iterative method in Figure 2.2, this new algorithm proceeds first by solving the AC-OPF problem, followed by checking violated contingencies and adding the most violated contingencies dynamically. However, most importantly, only one IPM solution process is required for the whole process.

Since the objective function (2.1) only consists of the real power generation under the base-case condition, there is no optimization for the contingency case sub-problem. That is, the optimal solution to the SCOPF problem with all contingencies only finds a feasible operation point for the state variables corresponding to each contingency. This can also be explained from the iterative solution technique for SCOPF (Figure 2.2), as it only requires checking whether or not the current setting, which is obtained from a reduced security model, is feasible for all the required contingencies.

On the other hand, IPM algorithms proceed by solving a sequence of barrier problems. After each IPM iteration, one interior point solution optimizing the current barrier function is available. For the SCOPF problems, these sub-optimal solutions correspond to a sequence of power system operation plans.

When solving the reduced security model, this sequence of solutions move towards the optimal operation point for the reduced SCOPF problem. Therefore, these sub-optimal solutions are potential trial points of the contingency analysis for the contingencies which are not included in the reduced security model. This is due to the fact that we believe these trial points can also report violated constraints and find potentially active contingencies for the problem.

As a result, each IPM iteration can be split into two stages: The first stage uses a reduced contingency set to build a reduced SCOPF model and generate a central point for the reduced problem. This central point is used to set the global variables, i.e. the real power generation and voltage at PV buses. In the second stage, similarly to the iterative SCOPF solution process, the global variables obtained from the first stage are fixed and used to check if there is a feasible solution for the state variables for other contingencies which are not included in the reduced model.

If no violated contingency is found with this sequence of sub-optimal solutions, the sequence converges to the optimal solution of the reduced SCOPF problem, and this solution will also be the optimal solution to the full SCOPF problem. Otherwise, several contingencies should be detected due to the constraint violations caused by the current power system control setting. If this is the case, these detected contingencies should also be incorporated into the reduced SCOPF model.

Additionally, in the second stage, instead of solving the load flow problem explicitly in each IPM step, a Newton system of equations associated with the load problem is solved to obtain the search direction from the current local variables. The advantage of this approach is that it can find a Newton direction towards a feasible solution to the load flow problem for each contingency, and it can also update the dual variables with the given barrier parameter μ. The new iterate obtained from this two-stage approach has the same dimension as that of the full SCOPF model, and it can be used as a warm-start step for the revised reduced SCOPF model which contains a greater number of active contingencies.

This new scenario-generation-based approach is presented in the following sections for linear

programming problems. Details and discussions are mainly given for the scenario generation techniques. These scenario generation techniques are based upon the structure-exploiting IPM implementation presented in the previous section. It is worth mentioning again that the filter line search technique is especially designed to guarantee the global convergence of NLPs. For LP problems, there is no need to apply this globalization technique, and consequently the corresponding steps in Algorithm 4.1 can be skipped. However, the filter techniques are still retained in the scenario-generation-based IPM implementation, even though all the theoretical analysis of this approach is in the context of LP. The reason for keeping the filter is that it is interesting to see whether this scenario generation method can also work for nonlinear problems.

In Section 4.2.1, the heuristic scenario-generation-based algorithm is presented by introducing two sets to record the scenarios used to build the reduced problem and all other scenarios, respectively. The theoretical analysis in this section explores how to detect other essential scenarios while solving the reduced model. In Section 4.2.2 the theoretical analysis of the warm-start strategy is further explained, thereby addressing two questions: what happens when a scenario is added to the reduced model and why can a detected scenario be added into the IPM process between the steps? At the end of this chapter, the numerical experiments for both linear and nonlinear test SCOPF problems are given and discussed, from which we can see that this new algorithm can successfully solve all the nonlinear test AC-SCOPF problems. Additionally, its numerical behavior is more robust than that of the default structure-exploiting IPM implementation. We have an example which can be solved by the scenario-generation-based implementation but not the default one. Therefore, we expect this scenario-generation-based IPM to be an attractive tool for solving AC-SCOPF problems.

4.2.1 Scenario Generation IPM Algorithm

Before describing the scenario-generation-based IPM algorithm proposed for solving the AC-SCOPF problem, firstly a partition of the set of contingency scenarios $\mathcal{C}$ is introduced: $\mathcal{C}_A \subseteq \mathcal{C}$ denotes the set of all potentially active scenarios, which is the exact contingency set used to build the reduced SCOPF model; $\mathcal{C}_I \subseteq \mathcal{C}$ denotes the set of all other so-called inactive scenarios. For reasons of clarity, in the rest of this thesis, the set $\mathcal{C}_A$ is referred to as the active set, while $\mathcal{C}_I$ is referred to as the inactive set. It is trivial to observe that $\mathcal{C}_A \bigcup \mathcal{C}_I = \mathcal{C}$ and $\mathcal{C}_A \bigcap \mathcal{C}_I = \emptyset$. In the following content, all the local vectors or matrices in the inactive scenario are represented with a bar, e.g. $\bar{x}$ denotes the local variables in the inactive scenarios, such as the state variables in the SCOPF model. Vectors or matrices without a bar correspond to the local variables in the reduced problem. Additionally, the subscript 0 is used for terms given by the global variables, such as the real generation and PV bus voltage in the SCOPF model. Also, let us assume that the full model, which contains all the given contingencies, is feasible.

With the partition $\mathcal{C}_A$ and $\mathcal{C}_I$ and the use of the subscript 0, the structured LP problem

with full scenarios can be stated in the following form

$$\begin{aligned} \min: \quad & c^\top x + c_0^\top x_0 \\ \text{s.t:} \quad & \mathbf{A}x + \mathbf{A}_0 x_0 = b, && (4.17) \\ & \bar{\mathbf{A}}\bar{x} + \bar{\mathbf{A}}_0 x_0 = \bar{b}_0, && (4.18) \\ & x, \bar{x}, x_0 \geq 0, \end{aligned}$$

where the constraints (4.17) and (4.18) correspond to the set $\mathcal{C}_A$ and $\mathcal{C}_I$, respectively.

Additionally, we define the corresponding reduced LP problem as

$$\begin{aligned} \min: \quad & c^\top x + c_0^\top x_0 \\ \text{s.t:} \quad & \mathbf{A}x + \mathbf{A}_0 x_0 = b, && (4.19) \\ & x, x_0 \geq 0, \end{aligned}$$

which only contains the scenarios given by the set $\mathcal{C}_A$. Note that the objectives of the reduced and full model are the same and they do not contain any contribution from the local variable $\bar{x}$ in the set $\mathcal{C}_I$, e.g. SCOPF problems can satisfy this restriction.

The idea is to use a sequence of reduced models, with an increasing set $\mathcal{C}_A$, to find the solution to the full model. We assume that both models start from the same initial point.

Firstly, let us focus on the full model, bearing in mind that this is the problem to which we really want to find the solution. When applying IPM to solve this full model, the common approach is to solve a sequence of barrier problems (2.48) (see Chapter 2). After dropping the iteration index, the KKT conditions of the barrier problem with parameter μ for the full model are

$$\begin{aligned} \mathbf{A}x + \mathbf{A}_0 x_0 &= b, && (4.20a) \\ \mathbf{A}^\top y + z &= c, && (4.20b) \\ \mathbf{Z}\mathbf{X}e &= \mu e, && (4.20c) \\ \bar{\mathbf{A}}\bar{x} + \bar{\mathbf{A}}_0 x_0 &= \bar{b}, && (4.20d) \\ \bar{\mathbf{A}}^\top \bar{y} + \bar{z} &= 0, && (4.20e) \\ \bar{\mathbf{Z}}\bar{\mathbf{X}}e &= \mu e, && (4.20f) \\ \mathbf{Z}_0\mathbf{X}_0 e &= \mu e, && (4.20g) \\ \mathbf{A}_0^\top y + \bar{\mathbf{A}}_0^\top \bar{y} + z_0 &= c_0. && (4.20h) \end{aligned}$$

The corresponding Newton system of equations (2.56) may then be written as

$$\left[\begin{array}{ccc|ccc|cc} \mathbf{A} & & & & & & \mathbf{A}_0 & \\ & \mathbf{A}^\top & \mathbf{I} & & & & & \\ \mathbf{Z} & & \mathbf{X} & & & & & \\ \hline & & & \bar{\mathbf{A}} & & & \bar{\mathbf{A}}_0 & \\ & & & & \bar{\mathbf{A}}^\top & \mathbf{I} & & \\ & & & \bar{\mathbf{Z}} & & \bar{\mathbf{X}} & & \\ \hline & & & & & & \mathbf{Z}_0 & \mathbf{X}_0 \\ & \mathbf{A}_0^\top & & & \bar{\mathbf{A}}_0^\top & & & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta y \\ \Delta z \\ \hline \Delta \bar{x} \\ \Delta \bar{y} \\ \Delta \bar{z} \\ \hline \Delta x_0 \\ \Delta \hat{z}_0 \end{array}\right] = \left[\begin{array}{c} \xi_p \\ \xi_d \\ \xi_\mu \\ \hline \bar{\xi}_p \\ \bar{\xi}_d \\ \bar{\xi}_\mu \\ \hline \xi_{\mu,0} \\ \hat{\xi}_{d,0} \end{array}\right], \tag{4.21}$$

where $\xi_p = b - \mathbf{A}x - \mathbf{A}_0 x_0$, $\xi_d = c - \mathbf{A}^\top y - z$, $\xi_\mu = \sigma\mu e - \mathbf{X}\mathbf{Z}e$, $\bar{\xi}_p = \bar{b} - \bar{\mathbf{A}}\bar{x} - \bar{\mathbf{A}}_0 x_0$, $\bar{\xi}_d = 0 - \bar{\mathbf{A}}^\top \bar{y} - \bar{z}$, $\bar{\xi}_\mu = \sigma\mu e - \bar{\mathbf{X}}\bar{\mathbf{Z}}e$, $\xi_{\mu,0} = \sigma\mu e - \mathbf{X}_0\mathbf{Z}_0 e$, and $\hat{\xi}_{d,0} = c_0 - \mathbf{A}_0^\top y - \bar{\mathbf{A}}_0^\top \bar{y} - z_0$ are the residuals at the current iteration. Generally, each IPM step requires the solution of one such linear system (4.21) in order to provide a descent direction $(\Delta x, \Delta y, \Delta z, \Delta \bar{x}, \Delta \bar{y}, \Delta \bar{z}, \Delta x_0, \Delta \hat{z}_0)$ for the full model.

On the other hand, in the first stage of the scenario generation approach, we try to focus on the behavior of the reduced problem (4.19). Similarly to the above process, we can derive the KKT conditions for the reduced model as

$$\mathbf{A}x + \mathbf{A}_0 x_0 = b, \tag{4.22a}$$

$$\mathbf{A}^\top y + z = c, \tag{4.22b}$$

$$\mathbf{X}\mathbf{Z}e = \mu e, \tag{4.22c}$$

$$\mathbf{Z}_0\mathbf{X}_0 e = \mu e, \tag{4.22d}$$

$$\mathbf{A}_0^\top y + z_0 = c_0. \tag{4.22e}$$

This KKT system is solved by Newton's method and the step direction $(\Delta x, \Delta y, \Delta z, \Delta x_0, \Delta z_0)$ is obtained by solving the following linear system of equations:

$$\left[\begin{array}{ccc|cc} \mathbf{A} & & & \mathbf{A}_0 & \\ & \mathbf{A}^\top & \mathbf{I} & & \\ \mathbf{Z} & & \mathbf{X} & & \\ \hline & & & \mathbf{Z}_0 & \mathbf{X}_0 \\ & \mathbf{A}_0^\top & & & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta y \\ \Delta z \\ \hline \Delta x_0 \\ \Delta z_0 \end{array}\right] = \left[\begin{array}{c} \xi_p \\ \xi_d \\ \xi_\mu \\ \hline \xi_{\mu,0} \\ \xi_{d,0} \end{array}\right], \tag{4.23}$$

where $\xi_p = b - \mathbf{A}x - \mathbf{A}_0 x_0$, $\xi_d = c - \mathbf{A}^\top y - z$, $\xi_\mu = \sigma\mu e - \mathbf{X}\mathbf{Z}e$, $\xi_{\mu,0} = \sigma\mu e - \mathbf{X}_0\mathbf{Z}_0 e$, and $\xi_{d,0} = c_0 - \mathbf{A}_0^\top y - z_0$.

As mentioned previously, we would like to use the solution to the reduced problem as an

approximation of the solution to the full model. However, due to the dimension reduction, the reduced model cannot decide the Newton direction $(\Delta\bar{x}, \Delta\bar{y}, \Delta\bar{z})$ for the local variables defined by $\mathcal{C}_I$.

One remedy for handling the additional dimensions is to solve an additional KKT-like system for $\mathcal{C}_I$:

$$\bar{\mathbf{A}}\bar{x} = \bar{b} - \bar{\mathbf{A}}_0(x_0 + \Delta x_0), \tag{4.24a}$$

$$\bar{\mathbf{A}}^\top \bar{y} + \bar{z} = 0, \tag{4.24b}$$

$$\bar{\mathbf{Z}}\bar{\mathbf{X}}e = \mu e, \tag{4.24c}$$

where the search direction Δx_0 is substituted by the obtained value from the linear system (4.23) for the reduced problem. This stage is considered as the second stage since the KKT-like system (4.24) can only be solved after the KKT system (4.22), since Δx_0 is required input data for (4.24).

This system is also solved by the Newton's method and we have

$$\begin{bmatrix} \bar{\mathbf{A}} & & \\ & \bar{\mathbf{A}}^\top & \mathbf{I} \\ \bar{\mathbf{Z}} & & \bar{\mathbf{X}} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} \bar{\xi}'_p \\ \bar{\xi}_d \\ \bar{\xi}_\mu \end{bmatrix}, \tag{4.25}$$

where $\bar{\xi}'_p = \bar{b} - \bar{\mathbf{A}}\bar{x} - \bar{\mathbf{A}}_0(x_0 + \Delta x_0) = \bar{\xi}_p - \bar{\mathbf{A}}_0\Delta x_0$, $\bar{\xi}_d = 0 - \bar{\mathbf{A}}^\top \bar{y} - \bar{z}$ and $\bar{\xi}_\mu = \sigma\mu e - \bar{\mathbf{X}}\bar{\mathbf{Z}}e$. It is clear that the solution to this linear system is unique.

As a result, solving the linear system (4.25) finds a descent direction for the appropriate local variables in each scenario in $\mathcal{C}_I$, and the solution to the Newton system (4.23) can give a descent direction for the reduced model which is built from all the scenarios in $\mathcal{C}_A$. Since $\mathcal{C}_A \bigcup \mathcal{C}_I = \mathcal{C}$, it is obvious that the sum of the solution dimensions of these two smaller systems is identical to that of the full model (4.21). Therefore, the solutions obtained from the KKT systems (4.22) and (4.24) can be used simultaneously as an approximation of the solution to the full system (4.20). They can also be used as a warm-start iterate in the IPM process when the reduced model is required to be updated with more scenarios. The warm-start issue will be further discussed in the next section.

Comparing the KKT system for the full problem (4.20) with the two smaller KKT systems (4.22) and (4.24), it is obvious that almost all the equalities are identical. The only difference is that the equality (4.22e) does not contain the term $\bar{\mathbf{A}}_0^\top \bar{y}$, whereas the equality (4.20h) does. Additionally, it is easy to see that $\hat{\xi}_{d,0} = \xi_{d,0} - \bar{\mathbf{A}}_0^\top \bar{y}$ by comparing the right hand sides of linear systems (4.21), (4.23) and (4.25).

With a given partition $\mathcal{C}_A$ and $\mathcal{C}_I$, if a full model containing all the scenarios in $\mathcal{C}_A$ and $\mathcal{C}_I$ is solved by an IPM algorithm, there are two possible outcomes at its optimal solution

$(x^*, y^*, z^*, \bar{x}^*, \bar{y}^*, \bar{z}^*, x_0^*, z_0^*)$:

- If all the scenarios in partition $\mathcal{C}_I$ are not active, it means that for all the scenario candidates in $\mathcal{C}_I$, all of their inequality constraints are not active and the reduced cost for all the local variables should be identically zero. This implies that $\bar{x}^* > 0$, $\bar{y}^* = 0$ and $\bar{z}^* = 0$ at the optimal solution to the full model.

  Recall that $\bar{x}^*$ does not contribute to the objective function, in this case, solution $(x^*, y^*, z^*, x_0^*, z_0^*)$ can be directly obtained by solving the reduced problem which only contains scenarios in $\mathcal{C}_A$.

- If some scenarios in $\mathcal{C}_I$ are active scenarios, it implies that they contain active boundary constraints and nonzero reduced costs for certain local variables $\bar{x}^*$. If this is the situation, the optimal solution to the reduced problem is different from the one to the full model. The solution of the reduced problem may lead to constraint violation for some scenarios in $\mathcal{C}_I$, and can never be used as the optimal solution to the full model.

In practice, the second case could be more general since it is difficult to predict the necessary scenarios in advance. A random partition $\mathcal{C}_A$ and $\mathcal{C}_I$ may achieve the goal of building the reduced model with no constraint violations, but it is also difficult to apply in practice.

In order to obtain the same solution from the reduced problem, our heuristic is to find the useful scenarios between IPM steps, by updating the partition $\mathcal{C}_A$ and the reduced model. An obvious question arising is how to find the active scenario candidates, and when to add these scenarios into the model. To answer this question, one heuristic idea is to analyze the primal step sizes of the new iterates after both linear systems (4.23) and (4.25) have been solved.

Before giving a necessary condition for finding an active scenario candidate during the IPM process using step sizes, we firstly need to introduce some notations and discuss some issues about the step size.

Similar to the notation used above, the term α denotes the primal step size for the iterates belonging to the active set $\mathcal{C}_A$, while $\bar{\alpha}$ is used for the primal step size in the inactive set $\mathcal{C}_I$. For a LP problem, these step sizes are calculated by the fraction-to-the-boundary rule (4.10):

$$\alpha = \max\{\ \alpha \in (0,1] \mid x + \alpha \Delta x \geq (1-\kappa)x,\ x_0 + \alpha \Delta x_0 \geq (1-\kappa)x_0\ \} \tag{4.26}$$

$$\bar{\alpha} = \max\{\ \bar{\alpha} \in (0,1] \mid \bar{x} + \bar{\alpha} \Delta \bar{x} \geq (1-\kappa)\bar{x}\ \}, \tag{4.27}$$

with constant $\kappa \in (0,1)$. Unlike solving the full problem by IPOPT algorithm directly, in which the primal step size is calculated as the minimum of (4.26) and (4.27), here we solve two linear systems (4.23) and (4.25) and compute two primal step sizes for each of them. Depending on whether $\alpha \leq \bar{\alpha}$ or $\alpha > \bar{\alpha}$, the new algorithm takes different actions:

- If $\alpha \leq \bar{\alpha}$: We can set $\bar{\alpha} := \alpha$, which means that a shorter feasible step in the direction

of the inactive subset should be taken. Note that this smaller step would not violate the boundary constraints $\bar{x} \geq 0$, and this is similar to the common IPM approach where the smallest step size is used as the actual one to update iterates.

- If $\alpha > \bar{\alpha}$: This means that some components of variables $\bar{x}$ can only accept a smaller step size than α, otherwise they would exceed their boundary limits. In other words, some constraints corresponding to the scenario set $\mathcal{C}_I$ appear more active than those constraints listed in the reduced problem. Therefore, we suggest moving the most violated scenario, which leads to the largest step size $\bar{\alpha}$, into the scenario set $\mathcal{C}_A$; and reject trial points $(x + \alpha\Delta x,\, x_0 + \alpha\Delta x_0)$ and $(x_0 + \alpha\Delta x_0,\, \bar{x} + \bar{\alpha}\Delta\bar{x})$, then re-compute this iterate from a new reduced problem defined by an updated $\mathcal{C}_A$.

The following theorem provides a necessary condition for the active scenarios.

Theorem 4.1. *Compute the step sizes of the Newton directions using* (4.26) *and* (4.27)*, respectively. If some scenarios in set* $\mathcal{C}_I$ *are active scenarios of the full problem containing all the scenarios, then there must exist one IPM iteration in which* $\alpha > \bar{\alpha}$*.*

Proof: Proof by contradiction. Assume for all the IPM iterations, the step size in the active set $\mathcal{C}_A$ is always less than or equal to the step size in the inactive set $\mathcal{C}_I$, i.e. $\alpha^i \leq \bar{\alpha}^i, \forall i$. Additionally, assume the reduced problem reaches convergence at the kth iteration.

Let us use $\xi(\alpha)$ and $\bar{\xi}(\alpha, \bar{\alpha})$ to denote the residuals of the KKT systems (4.22) and (4.24) at the new iterate $(x + \alpha\Delta x,\, x_0 + \alpha\Delta x_0)$ and $(x_0 + \alpha\Delta x_0, \bar{x} + \bar{\alpha}\Delta\bar{x})$, respectively.

According to (4.24a), the primal infeasibilities $\bar{\xi}_p$ for the KKT-like system (4.24a) can be represented as:

$$\bar{\xi}_p = \bar{b} - \bar{\mathbf{A}}\bar{x} - \bar{\mathbf{A}}_0 x_0, \tag{4.28}$$

and the residual at the new iterate $(x + \alpha\Delta x,\, x_0 + \alpha\Delta x_0,\, \bar{x} + \bar{\alpha}\Delta\bar{x})$ is

$$\begin{aligned}
\bar{\xi}_p(\alpha, \bar{\alpha}) = &\quad \bar{b} - \bar{\mathbf{A}}(\bar{x} + \bar{\alpha}\Delta\bar{x}) - \bar{\mathbf{A}}_0(x_0 + \alpha\Delta x_0) \\
= &\quad (\bar{b} - \bar{\mathbf{A}}\bar{x} - \bar{\mathbf{A}}_0 x_0) - \bar{\alpha}\bar{\mathbf{A}}\Delta\bar{x} - \alpha\bar{\mathbf{A}}_0\Delta x_0 \\
= &\quad \bar{\xi}_p - \alpha(\bar{\mathbf{A}}\Delta\bar{x} + \bar{\mathbf{A}}_0\Delta x_0) + \alpha\bar{\mathbf{A}}\Delta\bar{x} - \bar{\alpha}\bar{\mathbf{A}}\Delta\bar{x} \\
= &\quad (1 - \alpha)\bar{\xi}_p - (\bar{\alpha} - \alpha)\bar{\mathbf{A}}\Delta\bar{x},
\end{aligned} \tag{4.29}$$

where the last equality in (4.29) comes from the first Newton equation of (4.25) (it implies $\bar{\mathbf{A}}\Delta\bar{x} + \bar{\mathbf{A}}_0\Delta x_0 = \bar{\xi}_p$).

Since there is no IPM iteration which satisfies $\alpha > \bar{\alpha}$, we set $\bar{\alpha} := \alpha$ at each step and equation (4.29) now can be further evaluated as

$$\bar{\xi}_p(\alpha, \bar{\alpha}) = \bar{\xi}_p(\alpha, \alpha) = \quad (1 - \alpha)\bar{\xi}_p, \tag{4.30}$$

and at the kth IPM step, the primal residual for the inactive part becomes

$$\bar{\xi}_p^{k+1} = (1-\alpha^k)\bar{\xi}_p^k = \dots \tag{4.31}$$

$$= \prod_{i=0}^{k}(1-\alpha^i)\bar{\xi}_p^0, \tag{4.32}$$

where $\bar{\xi}_p^0$ is the initial primal residual corresponding to the set $\mathcal{C}_I$ at the initial point (x^{init}, x_0^{init}, $\bar{x}^{init}$).

Similarly, the primal residual at the new iterate for the reduced problem is

$$\xi_p(\alpha) = (1-\alpha)\xi_p. \tag{4.33}$$

According to the assumption of convergence, the primal residual of the reduced problem should satisfy the termination criteria at the kth IPM step:

$$\xi_p^{k+1} = (1-\alpha^k)\xi_p^k = \dots \tag{4.34}$$

$$= \prod_{i=0}^{k}(1-\alpha^i)\xi_p^0 \leq tol, \tag{4.35}$$

where tol is the given tolerance of convergence and ξ_p^0 is the initial primal residual for the reduced problem.

If $\bar{\xi}_p^0 < \xi_p^0$, which is easily achieved by scaling in the presolve process, from (4.35) and (4.32), we have

$$\bar{\xi}_p^{k+1} < tol, \tag{4.36}$$

which means that the primal infeasibility for the inactive part is less than the required tolerance.

Due to the assumption of convergence, we have an optimal solution (x^k, x_0^k) to the reduced problem at the kth step. On the other hand, according to the inequality (4.36), the iterate $\bar{x}^k = \bar{x}^{init} + \sum_{i=1}^{k} \bar{\alpha}^i \Delta\bar{x}^i = \bar{x}^{init} + \sum_{i=1}^{k} \alpha^i \Delta\bar{x}^i$ is also a primal-feasible solution to the scenarios in the set $\mathcal{C}_I$.

Therefore, the integrated solution (x^k, x_0^k, $\bar{x}^k$) gives an optimal solution to the reduced problem and it also satisfies other contingency constraints which are not included in the reduced model. Hence it is a feasible solution to the full model. Most important of all, since the optimal solution to the reduced problem gives a lower bound to the full problem, the feasible solution (x^k, x_0^k, $\bar{x}^k$) is also an optimal solution to the full model.

As a result, the above discussion implies that the scenarios in $\mathcal{C}_I$ can be eliminated from the full problem, and consequently, the set $\mathcal{C}_I$ does not contain any active scenarios of the full model. Hence a contradiction occurs. □

Following the above discussion, we are now in position to present our new algorithm, which is an extension of Algorithm 4.1. The principal framework of our scenario-generation-based IPM algorithm is summarized in Algorithm 4.3. Note that Step 7 and Step 11, where backtracking line search and filter techniques are applied, are specialized for solving NLP problems. They can be neglected when solving LP problems.

Algorithm 4.3: IPM with Scenario Generation

Step Given base scenario c_{init}

Step 1. Initialization. Set $\mathcal{C}_A = \{c_{init}\}$ and $\mathcal{C}_I = \mathcal{C} \setminus \mathcal{C}_A$. Let $k = 1$.

Step 2. Build the reduced problem based on the set $\mathcal{C}_A$.

Step 3. Check convergence of the reduced problem and the primal infeasibilities of other scenarios in the set $\mathcal{C}_I$. If it converges, algorithm TERMINATE.

Step 4. If necessary, update the barrier parameter and re-initialize the filter for the new barrier problem.

Step 5. Compute the search direction for the reduced problem.

Step 6. Fix x_0 and z_0 in the other scenarios. Compute a search direction for each local load flow problem. Choose the step size α_k and $\bar{\alpha}_k$, so that $x_k + \alpha_k \Delta x_k > 0$ and $\bar{x}_k + \bar{\alpha}_k \Delta \bar{x}_k > 0$.

Step 7. Backtracking line search for the reduced problem. Update the step size α_k for the reduced problem.

Step 8. If $\alpha_k \leq \bar{\alpha}_k$, set $\bar{\alpha}_k \leftarrow \alpha_k$ and continue with Step 10. Otherwise reject this step, go to Step 9.

Step 9. Find the most violated scenario, which corresponds to the largest component of $\bar{\alpha}_k$. Add this scenario into $\mathcal{C}_A$. Update $\mathcal{C}_I = \mathcal{C} \setminus \mathcal{C}_A$. Re-initialize the filter by (4.8). Continue with Step 2.

Step 10. Update the dual variables.

Step 11. Augment the filter if necessary.

Step 12. Set $k \leftarrow k + 1$ and go back to Step 2.

One thing worth mentioning is that this heuristic scenario-generation idea pays more attention to the primal space, as it observes the difference between the primal step sizes α and $\bar{\alpha}$ but not the dual step sizes. It also only needs to check the primal infeasibilities for the inactive part defined by the set $\mathcal{C}_I$. This heuristic idea is mainly due to the fact that the optimization problem only requires finding a feasible point for the scenarios in the inactive set $\mathcal{C}_I$, while the local variables do not affect the objective function.

Note that only the primal step sizes are compared to determine if a contingency in $\mathcal{C}_I$ should be moved to the active set $\mathcal{C}_A$. However, the steps in both the active and inactive part of the problem are calculated in both the primal and dual space. After obtaining an acceptable pair

of primal step sizes α and $\bar{\alpha}$ in Step 8 of the algorithm, the step sizes for the dual variables z and $\bar{z}$ are computed by the fraction-to-the-boundary rule (4.11), respectively, in Step 10. All the iterates are then updated by the appropriate direction and the step size. Hence all the primal-dual iterates are guaranteed to be inside its boundaries and eligible as an interior point for next iteration.

Theorem 4.1 provides a way to detect possible active scenario candidates during the IPM process. We then remove the most active candidate from the inactive set $\mathcal{C}_I$, which decides the step size in the inactive part, and add it into the active set $\mathcal{C}_A$. The reduced problem is augmented with this scenario and the model is updated with the additive dimensions corresponding to the revised $\mathcal{C}_A$. Following this methodology, the final $\mathcal{C}_A$ would contain all the active scenarios, i.e. it contains all the umbrella contingencies of the SCOPF problem.

Furthermore, we have the following corollary:

Corollary 4.1. *For those scenarios which always stay in the inactive set $\mathcal{C}_I$, their corresponding primal infeasibilities $\bar{\xi}_p$ can be decreased at the same rate as the infeasibilities ξ_p for the active set $\mathcal{C}_A$.*

Proof:

This follows directly from the proof in Theorem 4.1.

□

It is worth mentioning again that the convergence criteria in Step 3 of Algorithm 4.3 is made up of two parts: the first part consists of a convergence check of the reduced problem, which has a proper convergence criteria stated in [70], i.e. the residuals from the KKT system are smaller than the given tolerance; the second part is only applied to the primal infeasibilities in the inactive scenarios from the set $\mathcal{C}_I$. Corollary 4.1 shows that this mixed convergence criteria is meaningful. It also theoretically guarantees that the dead loop caused by non-decreasing primal infeasibilities for $\mathcal{C}_I$ never happens.

Last but not least, when Algorithm 4.3 is applied to solve a structured problem, it results in solving a sequence of reduced barrier problems with an increasing number of scenarios given by $\mathcal{C}_A$. As mentioned earlier, the major computational effort is now to solve the linear system of equations (4.23) defined by the reduced problem. Additionally, this new approach also requires solving one Newton's system of equations (4.25) for each scenario in the set $\mathcal{C}_I$. While the system of equations (4.23) is becoming larger due to the increasing number of active scenarios, the number of the systems (4.25) is decreasing. Since $\mathcal{C}_A$ and $\mathcal{C}_I$ are defined by a partition of $\mathcal{C}$, it means the sum of the number of scenarios in these two sets is a constant and identical to the total number of predetermined contingencies. Therefore, the factorizations of the appropriate matrices for each contingency are still required in each IPM step. The computational effort of applying factorizations should be the same as that in the original structure-exploiting IPM for NLP. However, the major computational saving from this scenario-generation-based algorithm

comes from forming the Schur complement matrix for the reduced problem. The reason for this is that, often in the SCOPF problem, only a few scenarios are active and required to be considered in the reduced model. Therefore, forming the Schur complement matrix becomes much easier and cheaper than doing it from a model containing all the scenarios.

4.2.2 Warm-start strategy

Note that the above two theoretical results, Theorem 4.1 and Corollary 4.1, mostly focus on how to detect active scenarios and how the primal infeasibilities behave. However, none of them shows whether incorporating the current reduced model with the detected scenario and using approximation iterates from previous approximate KKT system, is applicable or not. Moreover, another question is that whether or not doing this incorporating step would affect the convergence property of IPM. These questions are typical issues about the warm-starting technique.

The warm-starting technique used in [23] is in a similar context as Algorithm 4.3. In [23], Colombo and Grothey generate an initial point by decomposing the problem and using an approximate solution of the subproblems as the warm-starting point to solve the appropriate problem.

However, unlike generic warm-start IPM method where warm-started point is used as an initial approximation before restarting the next round of IPM, here warm-start technique is applied between IPM iterates. That is, we keep the current level of barrier parameter μ and use information from previous iterates as a warm-started point for the new reduced problem while increasing the problem dimension.

Now we are in position to describe and analyze the warm-start strategy used to re-set the reduced model with an additional scenario.

Firstly let us consider the Newton system of equations for the reduced problem (4.23) and the other one in (4.25), simultaneously. These two systems can be stated in one incorporated formulation as

$$\left[\begin{array}{ccc|ccc|cc} \mathbf{A} & & & & & & \mathbf{A}_0 & \\ & \mathbf{A}^\top & \mathbf{I} & & & & & \\ \mathbf{Z} & & \mathbf{X} & & & & & \\ \hline & & & \bar{\mathbf{A}} & & & \bar{\mathbf{A}}_0 & \\ & & & & \bar{\mathbf{A}}^\top & \mathbf{I} & & \\ & & & \bar{\mathbf{Z}} & & \bar{\mathbf{X}} & & \\ \hline & & & & & & \mathbf{Z}_0 & \mathbf{X}_0 \\ & \mathbf{A}_0^\top & & & & & & \mathbf{I} \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta y \\ \Delta z \\ \hline \Delta \bar{x} \\ \Delta \bar{y} \\ \Delta \bar{z} \\ \hline \Delta x_0 \\ \Delta \hat{z}_0 \end{array}\right] = \left[\begin{array}{c} \xi_p \\ \xi_d \\ \xi_\mu \\ \hline \bar{\xi}_p \\ \bar{\xi}_d \\ \bar{\xi}_\mu \\ \hline \xi_{\mu,0} \\ \xi_{d,0} \end{array}\right]. \tag{4.37}$$

Thus, solving linear system of equations (4.23) and (4.25) can be expressed by solving the

equivalent reformation (4.37).

If at the kth IPM step, i.e. the step used to seek new iterate $x_k + \alpha_k \Delta x_k$, a scenario in $\mathcal{C}_I$ is detected as active by the method mentioned in previous section, then this scenario is moved into the active set $\mathcal{C}_A$ and consequently the reduced model need to be updated with this additional scenario. For clarifying the notation, assume that all the scenarios in $\mathcal{C}_I$ are detected as active in this step. Then we can use the form of (4.21) to denote the updated reduced problem, where now the superscript '-' corresponds to the scenarios added in this step. In other words, from this kth step, the reduced problem is modified as a full problem since set $\mathcal{C}_I$ is empty.

After having a bigger reduced problem, the solution $(x^k,\ x_0^k,\ \bar{x}^k)$ obtained from previous step with system for equation (4.37) is used as a warm-start approximation in system (4.21). Obviously, this warm-start point is not feasible in (4.21) and it will generate some infeasibilities. Since the incorporated system and the new reduced system consist of identical dimension, the difference between the data of these two systems can be interpreted as a perturbation. According to [22] and [35], this kind of perturbation can be absorbed by Newton method under some conditions. Hence, we analyze the infeasibilities in our case and derive sufficient conditions to guarantee that this perturbation can be absorbed in one modified Newton's step.

Similar to the approach used in [22] and [35], a feasible interior point algorithm is applied to solve the incorporated system (4.37) and the new reduced system (4.21), which implies the residual ξ_p and ξ_d are equal to zero for all the previous iterates. Comparing the iterate matrices appearing in (4.37) and in (4.21), once the warm-start solution is used in the later system, the only additive infeasibility is the dual infeasibility caused by the difference in the last rows of the matrices, i.e. the term $\bar{\mathbf{A}}_0^\top$ is no longer zero.

In order to absorb the perturbation, the following system of equations is discussed:

$$\left[\begin{array}{ccc|ccc|cc} \mathbf{A} & & & & & & \mathbf{A}_0 & & \\ & \mathbf{A}^\top & \mathbf{I} & & & & & & \\ \mathbf{Z} & & \mathbf{X} & & & & & & \\ \hline & & & \bar{\mathbf{A}} & & & \bar{\mathbf{A}}_0 & & \\ & & & & \bar{\mathbf{A}}^\top & \mathbf{I} & & & \\ & & & \bar{\mathbf{Z}} & & \bar{\mathbf{X}} & & & \\ \hline & & & & & & \mathbf{Z}_0 & \mathbf{X}_0 & \\ & \mathbf{A}_0^\top & & & \bar{\mathbf{A}}_0^\top & & & \mathbf{I} & \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta y \\ \Delta z \\ \hline \Delta \bar{x} \\ \Delta \bar{y} \\ \Delta \bar{z} \\ \hline \Delta x_0 \\ \Delta \hat{z}_0 \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ -\bar{\mathbf{A}}_0^\top \bar{y} \end{array}\right]. \tag{4.38}$$

Before continuing, let us assume all the iterates of global variables are bounded. That is, $\exists\ B > 0 :\ x_0 \leq Be$, where $e = (1,\ \ldots,\ 1)$ in appropriate dimension. Additionally, to avoid overburdening the notation, we use superscript '$\sim$' to denote the incorporated vectors and matrices, e.g. $\tilde{x} = (x, \bar{x}, x_0)$ and $\tilde{\mathbf{X}} = \operatorname{diag} \tilde{x}$. Additionally, let $\tilde{\xi}_d$ denote the dual perturbation appearing in the right hand side of (4.38). It is also worth mentioning that the following

theoretical analysis is based on the symmetric central path neighborhood (2.54).

Now, let us state the following theorem, which is a sufficient condition for the dual infeasibility $-\bar{\mathbf{A}}_0^\top \bar{y}$ to guarantee a successful warm-start iterate. The proof follows a similar method in the Lemma 4 and Lemma 5 from [22].

Theorem 4.2. *Let $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathcal{N}_s(\gamma)$ be the warm-start iterate. If for $\beta < 1$, we have*

$$\|\bar{\boldsymbol{A}}_0^\top \bar{y}\|_\infty \leq \beta \frac{\gamma^2 \mu}{\sqrt{n} B} \tag{4.39}$$

then we can take a full Newton step (4.38) *from the warm-start iterate and the feasibilities can be completely restored in one step.*

Proof:

As shown in the Lemma 5 in [22], we can define $\mathbf{Q} = \mathbf{I} - \tilde{\mathbf{Z}}^{-1}\tilde{\mathbf{A}}^\top(\tilde{\mathbf{A}}\tilde{\mathbf{X}}\tilde{\mathbf{Z}}^{-1}\tilde{\mathbf{A}}^\top)^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{X}}$, and then derive $\tilde{\mathbf{X}}^{-1}\Delta\tilde{\mathbf{x}} = -\mathbf{Q}\tilde{\mathbf{Z}}^{-1}\tilde{\xi}_d = -\tilde{\mathbf{Z}}^{-1}\Delta\tilde{\mathbf{z}}$, since only dual infeasibilities exist (see [22]).

Additionally, according to the Lemma 4 in [22], matrix $\mathbf{Q}$ is bounded by

$$\|\mathbf{Q}\|_\infty \leq \frac{\sqrt{n}}{\gamma}, \tag{4.40}$$

where n is the number of primal variables.

Therefore, we can derive

$$\tilde{\mathbf{X}}^{-1}\Delta\tilde{\mathbf{x}} = -\mathbf{Q}\tilde{\mathbf{Z}}^{-1}\tilde{\xi}_d = -\mathbf{Q}\begin{pmatrix} 0 \\ 0 \\ \mathbf{Z}_0^{-1}\bar{\mathbf{A}}_0^\top \bar{y} \end{pmatrix} = -\tilde{\mathbf{Z}}^{-1}\Delta\tilde{\mathbf{z}}, \tag{4.41}$$

and the bound

$$\|\tilde{\mathbf{X}}^{-1}\Delta\tilde{\mathbf{x}}\|_\infty \leq \frac{\sqrt{n}}{\gamma}\|\mathbf{Z}_0^{-1}\bar{\mathbf{A}}_0^\top \bar{y}\|_\infty \leq \frac{\sqrt{n}}{\gamma}\|\mathbf{Z}_0^{-1}e\|_\infty \|\bar{\mathbf{A}}_0^\top \bar{y}\|_\infty, \tag{4.42}$$

where the first inequality in (4.42) comes from the result of Lemma 4 in [22], directly.

As $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathcal{N}_s(\gamma)$, we have

$$\gamma\mu \leq x_0 z_0 \leq \frac{\mu}{\gamma}. \tag{4.43}$$

Recalling the assumption $x_0 \leq B$, we can obtain

$$\frac{\gamma\mu}{B} \leq z_0, \tag{4.44}$$

$$\Rightarrow \|\mathbf{Z}_0^{-1}e\|_\infty \leq \frac{B}{\gamma\mu} \tag{4.45}$$

From (4.42), (4.45) and (4.39), we can have

$$\|\tilde{\mathbf{Z}}^{-1}\Delta\tilde{\mathbf{z}}\|_\infty = \|\tilde{\mathbf{X}}^{-1}\Delta\tilde{\mathbf{x}}\|_\infty \leq \frac{\sqrt{n}B}{\gamma^2\mu}\|\bar{\mathbf{A}}_0^\top \bar{y}\|_\infty \leq \beta < 1, \tag{4.46}$$

where the first equality comes from (4.41). Note that equation (4.46) implies that the full Newton step (4.38) from the warm-start iterate is acceptable, which completes the proof.

□

As a result, Theorem 4.2 gives a sufficient condition for a scenario being added into the reduced model. This good property gives us a flexibility to add scenarios into the reduced model while the final convergence may still be ensured.

Additionally, it also implies that an inactive scenario can be added into the reduced model even in the very late state of the IPM process, since its corresponding dual variable $\bar{y} \to 0$ always satisfies condition (4.39). On the other hand, if we consider about adding scenario at the first few iterates of the IPM algorithm, the condition for the dual infeasibility in (4.39) may be easier to be satisfied, since μ is larger. Last but not least, since condition (4.39) relates to the data of matrix $\mathbf{A}_0$, scaling the problem matrix in advance may also be a worth remedy.

Anyhow, the good news is that we observe that in practice, much larger infeasibility can be absorbed by this warm-start strategy. Moreover, despite that all the theoretical results are stated for the linear programming and feasible interior point method, it can also solve the nonlinear AC-SCOPF problems successfully, by improving the efficiency and robustness. We will confirm these issues by our implementation and the computational experiments in the following section.

4.2.3 Numerical Results

We have implemented the proposed warm-starting strategy inside the structure-exploiting interior point algorithm, mainly to solve the AC-SCOPF problems. To demonstrate the efficiency and accuracy of our new approach, we firstly build some AC-SCOPF model from the standard IEEE test dataset [80].

Note that the IEEE reliability test system [39] defines a 73-bus system, which is basically modified from three 24-bus systems with some additive interconnectors. We follow the same methodology and data file to make up a 96-bus system and 192-bus system by pasting the 24-bus system several times. Since our algorithm requires that the objective function can be obtained from any generated reduced problem, we start from the OPF condition as the initial scenario in our approach, and all the proper line outage contingencies are initially included in the inactive set $\mathcal{C}_I$. Besides the nonlinear AC-SCOPF problems, we also include some DC-SCOPF problem from previous chapter, to see how our new approach will perform for linear programming.

To run the tests, we have used a 2.33GHz Intel(R) Xeon(R) computer with 4GB RAM,

running Redhat Enterprise Linux in 32bit mode. The convergence tolerance of the algorithm is set to 10^{-5} or smaller.

		Original			Scenario Generation		
Prob	No.Sce	time(s)	iters	No.Act	time(s)	iters	No.ActSce
6	2	<0.1	13	2	<0.1	13	2
IEEE_24	2	0.3	36	2	0.3	36	2
IEEE_24	3	0.3	27	3	0.3	28	3
IEEE_24	38	5.7	41	6	3.9	30	6
IEEE_48	78	51.8	71	11	32.4	52	15
IEEE_73	117	204.1	97	16	156.7	92	25
IEEE_96	158	351.5	106	20	252.9	76	27
IEEE_118	178	???	??	42	1225.2	75	46
IEEE_192	318	2393.7	132	26	1586.0	92	40
L26	41	0.4	14	2	0.3	11	2
L200	371	264.3	53	7	56.4	25	7
L300	566	1153.1	88	17	196.3	22	20

Table 4.1: Scenario generation results

In Table 4.1, we summarize the numerical experiments from the original structure-exploiting NLP approach mentioned in section 4.1, and the scenario-generation-based approach. The first column denotes the problems we used in our numerical tests. All the SCOPF test problems are named by the following pattern: digit means the number of buses in the network; word 'IEEE' means it is a IEEE test problem; letter 'L' is used for DC-SCOPF case as a linearized model.

The second column denotes the number of the required scenarios for each problem. For the SCOPF problem, this is the number of line outage contingency cases plus one for the base-case condition. For some test problems, we do not include all the line outages in the SCOPF model due to the feasibility issue. We need to remove several contingency in order to have a feasible problem. For instance, the line from bus 7 to bus 8 in the 24-bus system is the only connection between bus 7 and the other parts of the power network. Therefore, bus 7 will be exactly isolated from the system once this line is removed. In order to have a feasible n-1 problem, we need to exclude this transmission line in the credible contingencies. The 5th column which is defined by 'No.Act' refers to the number of actual active scenarios in the model. This number is calculated by judging whether or not a scenario contains nonzero reduced cost at the optimum to the full model. We use 10^{-5} as the tolerance to check if those values are equal to zero. Note that for each problem, this number is the least number the set $\mathcal{C}_A$ should contain, in order to build a reduced problem having the same solution as the full problem. The last column named by 'No.ActSce' presents the actual number of scenarios in the set $\mathcal{C}_A$ at the convergence. It is equal to the number we detected by scenario generation algorithm plus one for the initial setting. These numbers are larger than or equal to the corresponding 'No.Act' values in the same row.

From Table 4.1, it is clear that we can solve all the test problems by the scenario gen-

eration approach. Using the original structure-exploiting method, we find that only problem 'IEEE_118' cannot converge within the limit of iterations because of some numerical difficulties. However, this problem can be solved by either the non-structure implementation or the scenario generation algorithm and achieve same solution.

In addition, the 5th column also indicates the numbers of essential scenarios are much smaller than the total number of scenarios which is presented in the 2nd column. That means we unnecessarily enlarge the optimization problem by adding too many useless scenarios into the full model. The good news is that our scenario generation algorithm can successfully pick up all the active scenarios, and what is more important, the number of detected scenarios is very close to the number of actual active scenarios. This can be confirmed by comparing the values in the 5th and last column.

For most of the test problems, the new approach requires less IPM iterations and also less computational effort in each step as we described previously. As a result, it can solve the problems much more efficiently. Generally, this new scenario-generation-based algorithm can successfully solve both linear and nonlinear SCOPF problem. The most interesting example may be the L300 case, as we can solve it within 22 iterations instead of 88 iterations when identifying 20 useful contingency scenarios out of the 566 scenarios.

It is worth mentioning that Table 4.1 is obtained under different setting of convergence tolerance. This is because we require that our scenario generation implementation can get exactly identical optimal objective (at least 8 significant digits). We found that if we set the convergence tolerance as 10^{-5}, the solution got from the scenario generation method may be a little bit different from the one got by the default structure-exploiting IPM. For instance, the optimal objective of the AC-SCOPF problem with 192 buses is 2226.26414 or 2226.26371, depending on which method is applied to solve the problem. However, once we require a tighter tolerance as 10^{-6}, the optimal objectives can become identical and have the same value of 2226.26334. The additional cost of achieving such objective is five extra steps with the original method, while only one extra step is required if scenario generation method is applied.

Generally, in our numerical tests, we observe that the numerical process becomes more stable once the scenario generation method is applied.

This may be due to the fact that the reduced problem contains fewer variables and contingencies, and consequently, the central path of the problems is not perturbed by the inactive scenarios. This is also the main point of view given by He and Tits [42], as inactive contingencies can be considered as redundant constraints which may affect the central path.

Three Small NLP Problems

In the end of this chapter, three small problems are presented as examples to show how the scenario generation IPM approach performs, and also demonstrates some additional advantages

of this approach.

Consider the following three small nonlinear problems:

Example 4.1.

$$A: \quad \min \ (x-1)^2 \quad st: \ x^2+y=100, \qquad\qquad 0 \le x, y \le 100 \tag{4.47}$$

$$B: \quad \min \ (x-1)^2 \quad st: \ x^2+y=100, \qquad x^2+z=100, \quad 0 \le x, y, z \le 100 \tag{4.48}$$

$$C: \quad \min \ (x-1)^2 \quad st: \ x^2+y=100, \qquad x^2+z=104, \quad 0 \le x, y, z \le 100 \tag{4.49}$$

These three test problems are nonlinear problems with the same objective function. Obviously, all of them have constraint $x^2 + y = 100$. The optimal objectives of the first two problems is 0, when $x = 1$. The first two small problems do not contain any active scenario at the optimum, since there is no nonzero reduced cost. On the other hand, the first constraint in problem (4.49) is an active constraint since it really changes the optimal solution. At the optimal solution, $x = 2,\ z = 100$, variable z reaches its upper bound and the corresponding reduced cost is nonzero. Additionally, we can consider that problem (4.48) and (4.49) contain two contingencies where each contingency is defined by one constraint.

In the scenario generation IPM implementation, the first constraint in problems (4.47) and (4.48) is used as the base case, which is included in the reduced model initially, and variable x is set as the first stage variable. For the third problem (4.49), we test two different schemes depending on which constraint is considered as the base case.

The numerical experiments for these small NLP problems are summarized in Table 4.2, where problem $C1$ means that the first constraint in (4.49) is set as the base case, while $C2$ uses the second constraint as the base case.

		Original		Scenario Generation	
Prob	No.Sce	Iters	No.Act	Iters	No.ActSce
A	1	9	0	9	1
B	2	22	0	9	1
C1	2	10	1	10	2
C2	2	10	1	9	1

Table 4.2: Results for the three small test problems.

As shown in Table 4.2, problem (4.47) and (4.48) require 9 and 22 IPM steps, respectively, if scenario generation method is not applied. However, both of them can be solved with the same number of interior point step with the scenario generation method. This indicates that our scenario generation strategy can successfully deny the access of a repeated scenario.

Although problem $C1$ and $C2$ refer to the same NLP problem (4.49), they need different number of IPM iterations if scenario generation IPM is applied. This is mainly due to the fact that the second constraint in (4.49) is an active scenario but the first one is not. If the second

constraint is given as the base case in the new approach, IPM does not require adding the other one into consideration and hence it approaches the optimal solution faster. From these small examples, we can find that this new approach can move the iterates towards the optimal solution without being disturbed by the inactive constraints at termination.

Chapter 5

Contingencies Selecting Techniques

Despite the fact that both AC-SCOPF and DC-SCOPF problems have been solved successfully by structure-exploiting IPM, one common weakness of the structure-exploiting interior point algorithms presented in the thesis is that they need initially add all the required contingencies into the model.

Even though all the active contingencies could be successfully found by the new scenario generation interior point algorithm described in the previous chapter, this new approach still cannot avoid taking the local load flow problems defined by other discarded contingencies into consideration in every IPM iteration. As illustrated in the linear system (4.37), all the contingencies are required to be included in the mathematical model. Most important of all, the new approach still requires iterating appropriate solutions of all the inactive contingencies in the IPM process. Although the solution techniques are able to focus on the reduced SCOPF model with fewer contingencies, it is still inevitable that additional memory and solution time are required to analyze the unnecessary contingencies. A considerable amount of memory and excessive time may be spent on these unnecessary ones, especially when many contingencies are required in a SCOPF problem.

Conversely, if those active contingencies can be predicted in advance, the full AC-SCOPF model with credible contingencies can be replaced by a reduced SCOPF model with only active contingencies. The reduced model would produce an identical solution to the one from the full model, and most important of all, the dimension of its mathematical model can be much smaller than that of the original model. Then the SCOPF problem can be solved much more efficiently with less storage demands. In other words, only one additive iteration would suffice to find the optimal solution to the full AC-SCOPF model, if the generic iterative solution techniques (shown in Figure 2.2) are applied with given active contingencies. As a result, this will improve

the computational efficiency by avoiding spending extra time on the trial iterations and their corresponding contingency analysis steps.

In spite of everything, obtaining the aforementioned information about the active contingencies seems to benefit the AC-SCOPF solution process later. However, detecting active contingencies is difficult in advance since this can only be obtained explicitly by checking the nonzero reduced cost at the optimal solution to the full SCOPF model.

Although the major concern is to get an operation plan from solving AC-SCOPF problems in the real world, the AC-OPF and DC-SCOPF problems are much easier to be achieved. Therefore, some arising generic questions are whether or not the information of active contingencies can be obtained by solving a DC-SCOPF problem as an approximation; and if there is any hint of such information that can be found from the solution to an AC-OPF problem.

This chapter focuses on how to identify the active contingencies by solving some load flow problems. The modified IEEE reliability 24-bus test system [39] will be the main sample system. Firstly, a brief summary of this 24-bus system is given, accompanied by a discussion of its AC-SCOPF problem. The second part of this chapter goes to the different heuristic approaches used to identify the active contingencies. Those heuristic approaches can mainly be divided into two groups. The first group of approaches is mainly based on solving the OPF problem, which is presented in section 5.2; the other group consists of the different ways to exploit the use of DC-SCOPF approximation, which is discussed in section 5.3.

5.1 Summary of the 24-Bus System

The entire grid of the IEEE reliability 24-bus system is briefly illustrated in Figure 5.1, which contains 38 transmission lines and 11 PV buses.

From Figure 5.1, one can identify that line 38, which connects bus 7 and bus 8, is the only link between bus 7 and the other parts of the network. In order to have a feasible AC-SCOPF problem, line 38 cannot be included as a n-1 security contingency. Bear in mind that only the outage of a certain line can be regarded as a contingency in the thesis, the maximum number of contingencies for this particular 24-bus system is 37, which is the number of transmission lines minus one. This implies that the full AC-SCOPF problem for this 24-bus system contains 38 scenarios, where the additional one corresponds to the base case.

Additionally, a contingency is considered active if it contains nonzero reduced cost, at the optimal solution to the AC-SCOPF problem. In the numerical experiments, a tolerance level of 10^{-5} is used as the criteria to check whether or not a reduced cost for a particular variable is nonzero, and then it is able to conclude if a contingency is active or not.

After solving the 24-bus AC-OPF problem and its AC-SCOPF problem with full contingencies, the numerical results are summarized in Table 5.1. The first row presents the results from the AC-OPF problem, while the second row shows the results from the corresponding SCOPF

problem. Additionally, the 4th column in Table 5.1 denotes the indices of the active scenarios at optimum, where term '0' denotes the base-case scenario and other numbers denote the index of the missing line, respectively, e.g. number 9 means the contingency caused by lack of line 9. The last column gives the indices of buses where their attached real power generators need to be switched on. In other words, these buses are PV buses where the real power outputs are nonzero. The interpretation of the other columns are in accordance with the previous chapter.

Bus	No.Sce	No.Act	ActSce.ID	GenPV.ID
24	1	1	0	13, 15, 16, 18, 21, 22, 23
24	38	5	0, 9, 22, 23, 36	1, 2, 7, 13, 15, 16, 18, 21, 22, 23

Table 5.1: Active scenarios and buses with nonzero real generation in the 24-bus system. Test problems used in previous chapters.

If the operation plan strictly follows the optimal solution of the OPF problem, only the real power generators on PV buses 13, 15, 16, 18, 21, 22 and 23 are in operation. Obviously, all of them are presented schematically in the top half of Figure 5.1. As a result, the top half forms a power surplus region and the bottom half forms a power deficit region. Any line outage in the bottom can easily block the transmission system. For instance, under the OPF operating condition, real power flow transmits from bus 10 to bus 6 and then transfers to bus 2. Once line 9 is blocked off, power must go through bus 2 to bus 6 to satisfy the load demand at bus 6. However, there is not enough power that can be transferred into bus 2 and then be passed to bus 6, which leads to an infeasible optimization problem. Therefore, if the contingency defined by the outage of line 9, which connects bus 10 and bus 6, is added to the security model, the optimal solution must be different from the operation plan given by OPF.

In fact, it can be deduced from the solution to the full SCOPF model, which is also presented in Table 5.1. At the optimal solution to the SCOPF problem, the real generator at bus 2 is switched on to supply real power to bus 6. Similarly, real generator at bus 7 should also be kept in operation to meet the load demands at bus 7 and bus 8. Note that bus 1 is also switched on, since it is the reference bus which is mainly used to fill in the energy loss occurred in each contingency case.

Actually, after solving AC-SCOPF with full contingencies for this 24-bus system, there is surprisingly no active line flow constraint (2.13) in contingencies. Thus, all the line flows can be transferred under their own thermal limits in all the contingencies. This is mainly caused by the larger flow limits used in the contingency case while comparatively small limits are used in the base case. For this particular 24-bus system, the flow limits in a contingency case is roughly 1.25 times of that of the base case. As a result, power flows are bounded by the base case limits and they cannot reach the contingency limits in all the given contingencies. It is also worth mentioning that the limits of the voltage levels are quite tight. The test SCOPF problems only accept a slight perturbation for the norminal voltage level, which is $\pm 6\%$ for the

base case and ±10% for the contingencies.

Note that all the active contingencies in the AC-SCOPF model are identified mainly due to the nonzero reduced costs of reactive power generations (2.11). Furthermore, active contingencies 0, 9 and 22 also contain some voltage variables (2.12) whose reduced costs are nonzero.

According to [55], since both reactive power and voltage level are typical AC issues in a SCOPF problem, DC-SCOPF model may not find out all the necessary binding contingencies for the AC model. Therefore, when active contingencies are caused by the reactive power issues or voltage issues, it is impossible to identify such active contingencies from the DC approximation. This is mainly because that the DC model is a simplified variant of the AC model, neglecting all the reactive power in the model and assuming the power system network is lossless. Moreover, in order to linearize Kirchoff's laws, DC-OPF requires setting the voltage levels at all the buses to be identical, i.e. the DC-OPF model does not treat voltage level as a variable. Conversely, voltage level is also a critical issue in the AC model.

Since reactive power generations do not show up in the objective function (2.1) of a SCOPF problem, it is interesting for research purposes to relax the reactive power generation limits (2.11). Consequently, the result model can be used to exam how far a DC approximation can detect the active contingencies if only line flow variables have nonzero reduced cost.

Unfortunately, in the AC-SCOPF model, finding a feasible test problem with only flow limit constraints showing active is difficult, even with abandoning the reactive power generation limits. Alternatively, if the reactive power generations are abandoned and the contingency flow limits are reduced to the base-case level, the modified data also gives a 24-bus system, which slightly changes the optimal solution of the full SCOPF problem. However, in this modified model, there exists one active contingency caused by nonzero reduced cost of the flow variable only.

More specifically, this modified model defines a 24-bus system containing 7 active contingencies at the optimal solution to its full SCOPF problem. Similar to Table 5.1, the summary of the OPF problem and SCOPF problem associated with the modified 24-bus system is demonstrated in Table 5.2.

Bus	No.Sce	No.Act	ActSce.ID	GenPV.ID
24	1	1	0	13, 15, 16, 18, 21, 22, 23
24	38	7	0, 3, 7, 9, 12, 22, 36	1, 2, 7, 13, 15, 16, 18, 21, 22, 23

Table 5.2: Active scenarios and buses with nonzero real generation in the 24-bus system. Test problems used in this chapter.

In this modified 24-bus AC-SCOPF problem, active contingencies 3, 7 and 12 contain active flow limit constraints. While active contingencies 7 and 12 also involve active voltage limit constraints, contingency 3 is active only because the thermal flow in line 1 reached its limit. All other active contingencies are detected by containing nonzero reduced cost of voltage level.

Last but not least, it is worth mentioning that the problems illustrated in Table 5.1 is the one used in previous chapters. Contrarily, due to all the reasons mentioned above, the problems illustrated in Table 5.2 are used in this chapter only. The main reason of applying modified model is to force the derived AC-OPF model and AC-SCOPF model focusing on the effects caused by the power flow issues, rather than the power generation issues.

5.2 Finding Active Contingencies from AC-OPF

Since OPF problem is the basis of the SCOPF problem, many SCOPF solution techniques mainly start from solving the corresponding OPF problem [8, 17, 19, 27]. To cover the gap between the OPF and SCOPF, the original iterative methodology presented in [8] adds all contingencies in which constraint violations occur to the security model. However, this may result in detecting quite a lot of contingencies containing constraint violations, and then all these contingencies are required to be added to the OPF model. As a result, in the next step, the iterative method needs to solve a very large optimization problem.

To avoid bringing too many contingencies into the security model and to improve the computational efficiency, several researches [17, 19, 27] try to reduce the number of contingencies added by using different contingencies filtering techniques.

One critical issue is that all these measures use a trial and error framework to test the optimal solution to the current reduced security model. If this solution does not produce any violated contingencies, this solution is accepted as the solution to the full SCOPF model. Otherwise it requires adding violated contingencies into the security model and restart the solution process for the updated security model. However, there is no guarantee that no more constraint violations would occur at the solution of the new security model. Moreover, some other contingencies, which can be satisfied in the previous security models, may become violated at the new solution. This is due to the fact that the above approaches only add violated contingencies detected from current security model, but some contingencies whose constraints are very close to their limits are ignored. This kind of contingencies may also be added into the model in the later iterations and they are also potential candidates to become active contingencies.

On the other hand, if the restriction of only adding violated contingencies can be remitted, the security model may become more flexible to add contingencies and may solve the problem more efficiently once all the active contingencies have been added into the security model.

Hence, it encourages us to see if all the active contingencies can be found directly from some simple contingency analysis steps from the OPF model, and how efficiently this can be achieved. If there is a successful method to find a comparatively small set of binding contingencies , which includes all the candidates of active contingencies, the iterative framework can be discarded by solving just one reduced security model with all the potential contingencies.

Previous works mainly focus on the violated contingencies, such as the filtering techniques,

while no relevant work tries to analyze how easily the active contingencies can be found from the AC-OPF problem directly. Note that an active contingency may not be a violated contingency at the OPF solution, which will be shown in the later numerical test, but it becomes violated after some contingencies are included. Additionally, all the active contingencies together can overwrite the effects caused by other contingencies.

In the following section, several heuristic methods of contingencies analysis are investigated to see how far they can find out the active contingencies from AC-OPF problem.

5.2.1 Contingency Analysis from AC-OPF

Generally, after solving an AC-OPF problem, the contingency analysis step is set up from the current optimal solution. That is, it keeps the operation plan by fixing the real power generation and the voltage level for each PV bus. Also it applies simulation techniques, or solve load flow problems by solving equations, to evaluate if all the required contingencies are feasible or not.

Since a load flow problem or a simulation method can only determine if a given solution is feasible or not, it is not an optimization problem, making it difficult to rank all the contingency violation. Instead of seeking a feasible solution to the load flow problem for each contingency, we formulate two possible improvements of the load flow problem, each of which is a proper optimization problem.

Both optimization load flow models contain all the contingency constraints (2.11)-(2.18) except for the line thermal constraints (2.13). Depending on the different ways to evaluate the load flow, the line thermal constraints (2.13) can be replaced by one of the following two schemes.

- **Scheme.1:** Evaluate load flow by the ratio. This can be achieved by adding the following constraint (5.1),

$$\frac{\left(f^P_{c,(i,j)}\right)^2 + \left(f^Q_{c,(i,j)}\right)^2}{\left(f^+_{c,(i,j)}\right)^2} \leq S_c, \quad \forall (i,j) \in \mathcal{L}, \tag{5.1}$$

  where $f^+_{c,(i,j)}$ is the flow limit for line (i,j) in contingency c, and S_c is a slack variable.

- **Scheme.2:** Evaluate load flow by the explicit value. Similarly to the Scheme.1, it can be achieved by adding the following constraints (5.2),

$$\left(f^P_{c,(i,j)}\right)^2 + \left(f^Q_{c,(i,j)}\right)^2 - \left(f^+_{c,(i,j)}\right)^2 \leq S_c, \quad \forall (i,j) \in \mathcal{L}, \tag{5.2}$$

The main aim of these optimization models is to find a local load flow setting which implies that the most violated line flow in its corresponding contingency is minimized. Both of these

models can use the same objective function (5.3):

$$\min \ S_c. \tag{5.3}$$

If Scheme.1 is applied, the objective minimizes the largest proportion of the thermal flow and its limit. Otherwise it minimizes the largest explicit gap between the thermal flow and its limit, once Scheme.2 is applied. Note that in Scheme.2, a negative objective means this specific contingency is feasible since no line flow exceeds its flow limit.

No matter which scheme is applied to build the load flow problem, it provides a way to measure the flow capacities in a contingency. Hence, after all the optimization load flow problems have been solved, it gives an opportunity to rank the flow constraint violations and then find the potential violated contingencies.

Two different ranking methods of the constraint violations are taken into account. The first ranking method is applied to evaluate the most violated contingency, by ranking the optimal objective from each optimization load flow problem. Since only one line outage is considered in the security model, each transmission line can also define a contingency case. Therefore, the second ranking method comes from observing the largest possible thermal flow carrying in a certain transmission line throughout all the optimization load flow problems.

These three steps: solving OPF problem, solving optimization load flow problem for each contingency and ranking the line flow violations are coded in one merged model in the AMPL environment [31]. The numerical results from different ranking methods are illustrated in Table 5.4 and Table 5.3, respectively.

In Table 5.3 and Table 5.4, the results in red indicate the actual active contingencies which can also be found in Table 5.2. Contingencies 1 and 12, have infeasible load flow problems, as no local setting can be found under the given OPF operation plan. Hence they are listed at the top of Table 5.3.

Generally, from these two tables, it seems ranking by contingencies performs better than ranking by transmission lines. The former one succeeds in ranking all the six active contingencies higher than the later ranking method, for both Scheme.1 and Scheme.2 of measuring flow violations.

Focusing on Table 5.3, the outcome from Scheme.1 is similar to the one from Scheme.2. If ranking by contingencies is applied, both schemes can rank all the six active contingencies in the top 14 out of 37 contingencies in total. However, this also means that the reduced security model needs to include roughly half of the contingencies in order to achieve an identical solution to the full model.

It is also worth mentioning that if the thermal flow limits are restored, there are three infeasible load flow problems. As shown in Table 5.3, two of them are caused by contingency 1 and 12, respectively. The additional one is caused by the outage of line 11 since the load flow

	Scheme.1		Scheme.2	
Rank	Obj(%)	Cont_ID	Obj	Cont_ID
1	Inf	1	Inf	1
2	Inf	12	Inf	12
3	219.89	11	5.8027	11
4	80.8297	37	-0.9273	37
5	76.0301	36	-1.1593	36
6	66.6721	5	-1.2563	5
7	57.6041	7	-1.6958	7
8	56.203	35	-1.8059	22
9	55.1642	9	-1.8082	33
10	54.9517	18	-1.9575	9
11	54.7962	33	-2.0648	35
12	54.7954	22	-2.1696	18
13	52.7283	16	-2.2415	3
14	52.484	3	-2.2570	2
15	52.1929	14	-2.2745	8
16	51.7482	24	-2.2776	34
17	48.5998	6	-2.2824	16
18	48.514	8	-2.3091	14
		⋮		
36	50.8106	6	-2.3981	13
37	49.5836	8	-2.4593	6

Table 5.3: Contingency analysis from different schemes for the 24-bus system. Ranked by contingencies.

problem of this particular contingency has an optimal objective larger than 100% in Scheme.1 or larger than 0 in Scheme.2. This implies the flow constraint violations.

It is clear that all these three critical lines are in the bottom of the network displayed in Figure 5.1. As mentioned before, the bottom half of the grid is a power deficit region. Additionally, the further a power flow is transferred, the more energy may be lost in it. In this 24-bus example, even though the flow limits have been omitted, once a failure caused by losing one of these three lines occurs, the problem becomes infeasible due to the lack of real power generation (note that the reactive power generation limits have been omitted). This is because in a contingency case, only the reference bus is allowed to change its output, which may not be enough to cover the power loss. For instance, if the real power generation limit at the reference bus can be abandoned, all the load flow problems mentioned above can be feasible.

If contingency analysis is applied to each contingency from AC-OPF solution, it may produce some infeasible load flow problems for certain contingencies, even though the flow limits have been abandoned. Unfortunately, the full security model requires including all contingencies of this kind, and hence the solution to a security model may be quite different from the OPF solution. This is mainly caused by switching on some power generators. This great change between the behavior of an OPF and the behavior of a SCOPF becomes the bottleneck for using the OPF solution to identify the active contingencies for a SCOPF problem.

	Scheme.1		Scheme.2	
Rank	MaxFlow(%)	Line_ID	MaxFlow	Line_ID
1	148.287	12	5.80265	12
2	148.067	1	-0.913113	38
3	90.0744	38	-0.927253	1
4	89.9053	11	-0.927253	11
5	81.653	9	-1.25627	9
6	74.1918	7	-2.21026	6
7	74.0245	18	-2.27446	19
8	73.6987	6	-3.1116	8
9	68.3782	33	-3.12987	7
10	68.1985	23	-3.58913	2
11	67.391	37	-3.72659	5
12	65.7695	36	-3.74552	3
13	64.262	20	-4.2725	4
14	64.262	21	-15.4953	18
15	62.6993	16	-19.6432	37
		⋮		
17	62.6495	28	-20.4281	36
		⋮		
26	52.3069	22	52.3069	22
		⋮		
28	33.7501	3	26.9125	35
		⋮		
38	16.2565	30	-37.9061	30

Table 5.4: Contingency analysis from different schemes for the 24-bus system. Ranked by transmission lines.

Additionally, contingencies that lead to large change of the operation plan may be the candidates for active contingencies, e.g. contingency 12 in the 24-bus system. This scenario results in an interesting question – whether the analysis from reduced SCOPF with just one line contingency will imply more useful information for the full security model?

5.2.2 Contingency Analysis from Reduced SCOPF with One Contingency

Instead of applying load flow problems to check if contingencies are acceptable, a group of reduced security problems is solved to find the changes of generators. Each reduced SCOPF problem only contains the base case and one line outage contingency, hence this reduced SCOPF model can be announced as 'Base&1' problem for short. Note that the solution to each problem corresponds to a power operating plan. From solving 'n' Base&1 problems for all the given contingencies, and obtaining 'n' operation plans, an outline of the full security problem and a clue to the active contingencies is expected.

After solving 'n' Base&1 problems, the operating plans of real power generators at PV

buses are summarized in Table 5.5. In the table,'Y' denotes the nonzero real output from the corresponding bus, while 'N' denotes that the generators on that bus are not in service. Additionally, the ranking of all objectives is utilized to measure how far the operation plan of an Base&1 problem is from the solution to the OPF problem or SCOPF problem. The ranking results are also presented in Table 5.5.

			PVBUS			
Rank	Cont_ID	Obj	b1	b2	b7	b13
0	SCOPF	296.325	Y	Y	Y	Y
1	7	291.399	Y	Y	Y	Y
2	2	290.698	Y	Y	Y	Y
3	9	286.485	Y	Y	Y	Y
4	8	286.275	Y	Y	Y	Y
5	22	280.638	N	Y	Y	Y
6	11	274.293	Y	Y	Y	Y
7	5	273.240	N	Y	Y	Y
8	1	271.185	N	Y	Y	Y
9	37	271.062	N	Y	Y	Y
10	36	271.036	N	Y	Y	Y
11	33	268.123	N	Y	Y	Y
12	12	267.030	N	N	Y	Y
13	4	266.682	N	N	Y	Y
14	18	265.390	N	N	Y	Y
15	35	265.022	N	N	Y	Y
16	3	264.957	N	N	N	Y
		⋮				
37	14	264.957	N	N	N	Y
38	OPF	264.957	N	N	N	Y

Table 5.5: Ranking objectives of n Base&1 models and summary of the working real power generators.

Note that for all Base&1, OPF and SCOPF problems, the real power generators attached at PV bus 15, 16, 18, 21, 22 and 23 are in operation, hence they are ignored in the table for clarification purpose. Additionally, results from rank 16 to 37 are neglected since all of them can be covered by the base case, and they have exactly the same solution as the solution from OPF problem. Note that the contingency caused by losing line 3 is active at the full SCOPF problem, but it cannot play any role if it is the only required contingency.

Unfortunately, even though solving 'n' Base&1 problems can tighten the lower bound of the objective to the full SCOPF problem, it can hardly find all the active contingencies due to the discussion above. In the 24-bus example, shown in Table 5.4, the last active contingency can never be detected from this approach, although there are three, out of six, active contingencies ranked in the top 5.

In other words, from the AC-OPF problem, it is difficult to conclude which contingencies can become active ones. For this 24-bus system, even though two contingencies and three contingencies can be listed in the top 5 in Table 5.3 and Table 5.5, respectively, it is still

impossible to identify the last active contingency given by the outage of line 3.

5.3 Finding Active Contingencies from DC-SCOPF

Besides exploiting the use of AC-OPF to solve the corresponding SCOPF problem, the use of DC-SCOPF approximation may also throw light on improving the iterative solution technique. In [50], Marano-Marcolini et.al suggest using DC-SCOPF approach to detect the potentially binding contingencies that could lead to a thermal flow loading of more than 98%. They have presented some superb results in [50], but also mentioned that the weakness of such approximation is the use of reactive power in the AC model.

Similarly to the approaches described in Section 5.2.1, two schemes can be applied to measure the contingencies after solving the DC-SCOPF with full binding contingencies. These two approaches can be summarized as follows:

- **Scheme.1:** Evaluate load flow by the ratio. This measurement calculates the ratio between the absolute value of the contingency flow and its limit, namely: $\|f_{c,l}\|/f_l^+$.

  Note that this method is also adopted in [50].

- **Scheme.2:** Evaluate load flow by the explicit value. This measurement calculates the difference between the absolute value of the contingency flow and its limit, namely: $\|f_{c,l}\| - f_l^+$.

Obtained by solving the DC-SCOPF with full contingencies for the IEEE 24-bus system, Table 5.6 presents the ranks of the maximum line loading at each binding contingency. The second and third columns indicate the results from ranking the maximal line flow loading in ratio for each contingency, which refer to Scheme.1. The last two columns indicate the results from ranking the difference between load flow and its limit, which refer to Scheme.2. Additionally, the active contingencies for the AC-SCOPF problem are the base case and also the contingencies 3, 7, 9, 12, 22 and 36, as emphasized in red in Table 5.6.

Similar to the conclusion described earlier, these two ranking schemes still cannot list all the active contingencies in the top. However, ranking the difference seems a better choice than ranking the line loading by ratio, as it ranks all the active contingencies in the top 11 out of 38.

This is due to the fact that some particular transmission lines have much smaller loading limits. For instance, the flow limit of line 9 is 1.75 units, while it is 5 units for transmission line 21. If the line flow flexibilities is judged by ratio, load of line 9 has a quite low ratio as 73%, while the load of line 21 is roughly 85% of its upper limit. However, compared to line 21, line 9 is more dangerous as it can hardly cover the impact from line flow surge caused by losing other transmission lines, since its absolute capacity is smaller than that of line 21.

Although DC approximation is not successful in detecting all the active contingencies, it is worth mentioning that it can find two important contingencies, contingency 11 and 12, where

	Scheme.1		Scheme.2	
Rank	Loading(%)	Cont_ID	Loading	Cont_ID
1	100	11	0	11
2	100	12	0	12
3	91.2584	22	-0.4371	22
4	91.2584	33	-0.4371	33
5	85.9353	18	-0.4399	37
6	84.8964	20	-0.458	9
7	84.8964	21	-0.458	5
8	84.1791	24	-0.4835	36
9	83.3218	17	-0.5626	18
10	81.0878	16	-0.6971	3
11	78.365	23	-0.7022	7
12	76.9226	19	-0.7345	14
13	76.3709	13	-0.7357	2
14	74.8654	37	-0.7382	16
15	74.5225	10	-0.7391	24
16	73.8286	5	-0.7436	17
17	73.8286	9	-0.7485	23
18	72.3719	36	-0.7501	19
		⋮		
23	68.5583	3	-0.7552	21
		⋮		
30	67.4856	7	-0.7579	10
		⋮		
38	65.2934	34	-0.823442	34

Table 5.6: Ranking maximal flow loadings after solving full DC-SCOPF model for IEEE 24-bus system.

both of them contain transmission lines working in full capacity in its corresponding contingency case. These two contingencies are also critical in the AC case, as their load flow problems are infeasible if solution from AC-OPF is applied, which is shown in Table 5.3.

For this particular 24-bus problem, DC-SCOPF provides a poor approximation to the AC-SCOPF problem and it cannot identify potentially active contingencies for AC model. Although this 24-bus problem has been modified in order to focus on the flow limit, it is still difficult to find a problem with only active flow limits at the optimal solution to the full SCOPF problem. This is mainly due to the power of Kirchoff's laws, which link voltage issue with the power flow. Additionally, the reactive power also plays a very important role to maintain the power system in the AC model. The modified 24-bus problem relaxes the bounds of the reactive generation, which implies some reactive flow can be quite large in the AC model. Contrarily, the reactive power is neglected in the DC approximation. Hence, it is not surprising that DC-SCOPF approximation cannot provide a sensible idea of the active contingencies for this particular 24-bus AC-SCOPF problem.

In conclusion, it seems the use of DC-SCOPF approximation is able to identify certain

potential violated contingencies for the AC model, as it can detect contingencies 11 and 12 for the 24-bus system. However, the major bottleneck of the DC model is that it cannot identify all the active contingencies efficiently.

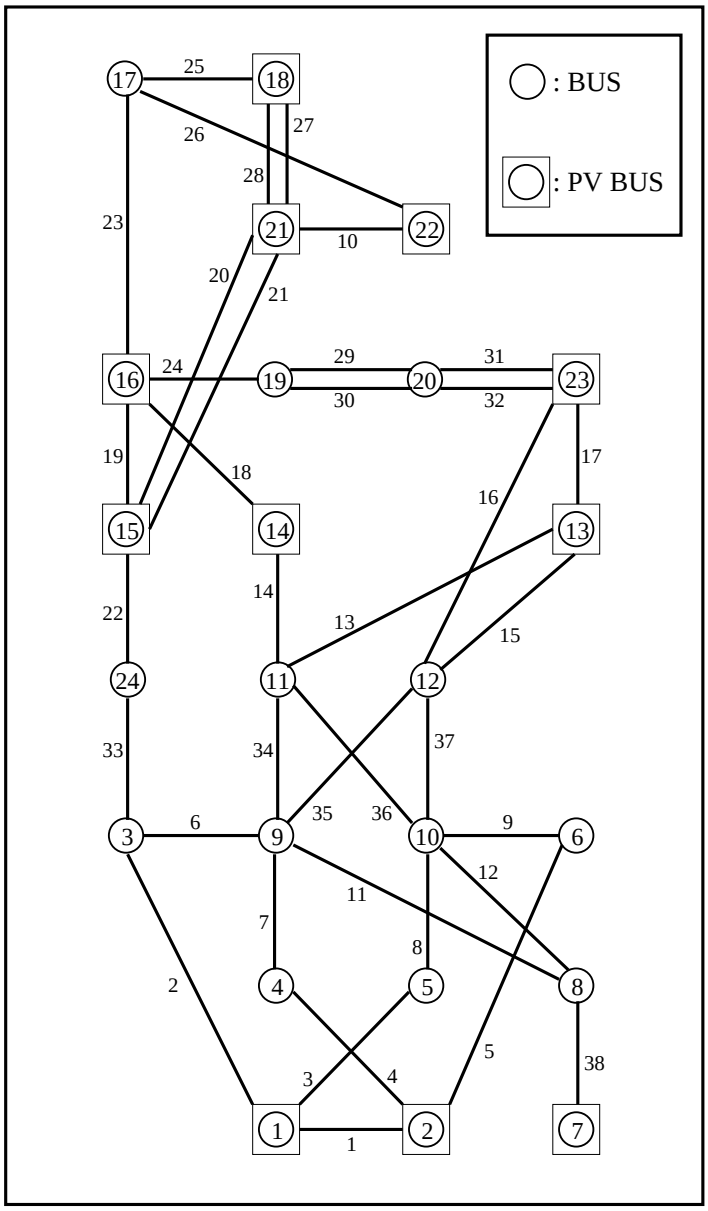

Figure 5.1: IEEE 24-bus system.

Chapter 6

Interface to Other Software

6.1 Introduction

Nowadays, with the fast growing software industry, designing an optimization algorithm is much easier, especially in a large project. This is because quite a lot of useful tools, such as the algebraic modeling languages and the solver for the linear systems, are available. These tools have been widely tested and improved. Hence they are widely accepted in the optimization area, due to their robustness and efficiency.

This chapter describes the software which has been applied in our research. Firstly, a summary of the modeling language AMPL [31], which is used to build all the mathematical models in the thesis, is presented in Section 6.2. Secondly, we demonstrate how the structured information is exchanged between AMPL and the IPM solver in Section 6.2.1. The last section of this chapter gives a brief introduction to the linear system solvers.

6.2 Modeling Language: AMPL

There are many available algebraic modeling languages, such as AMPL, GAMS [81], XPRESS [77], AIMMS [82] and so on. Each of these modeling languages offers a friendly environment to build the optimization model. They provide a platform for people to construct a model in a simple format similar to its standard arithmetic expressions and hence logically ease the model building. Additionally, through this way of building model, modifying and examining the models become easier for users, compared to building models in a proper programming language such as C++. Another advantage of algebraic modeling languages is that they can separate the model from data, which means the model does not require further modification and remains concise.

Among these modeling languages, AMPL is chosen as the default tool to communicate the structure-information in our implementation due to the following reasons:

- AMPL can be applied to both linear and nonlinear problems, as it supports the evaluations of first and second order derivatives, computed by automatic differentiation for NLP models. In addition, each information can be easily obtained with a single statement in the programming language.

- AMPL provides its solver library (ASL) as an open source. This allows a programming language to read the .nl files and to access AMPL's automatic differentiation facilities.

- AMPL is widely used and an AMPL interface to the NEOS Server [83] is available. Hence it offers more opportunities for people to run benchmarks and find the best solver corresponding to their particular problems, e.g. nonlinear solvers FILTER [30], LANCELOT [25] and LOQO [66, 12, 67]. All these solvers can be accessed from AMPL model via NEOS Server.

- AMPL can define vectors and matrices in a sparse form. That is, instead of calling all the data in matrices and vectors, it only requires the nonzero elements and their corresponding row/column indices.

- It is possible to use user-defined functions within AMPL. This makes AMPL a more flexible modeling language, e.g. the specific nested structure in the model can be achieved by Structured Modeling Language [24], which is a parallelizable implementation of a structure-conveying extension to AMPL modeling language.

As far as the author is concerned, other modeling language products might have some of the above advantages, but only AMPL has all the listed strengths.

6.2.1 Interface to AMPL

In order to solve a nonlinear problem with the structure-exploiting IPM implementation mentioned in Chapter 4, the first order differentiation information, Jacobian, and the second order differentiation information, Hessian, are necessary to build the KKT system. Unlike the linear case where the Jacobian is a constant matrix and Hessian is null, this information needs to be evaluated at current point in the nonlinear system, i.e. Jacobian $\mathbf{J}(x)$ and Hessian $\mathbf{H}(x, y)$ are functions of point x or (x, y). Therefore, they must be updated iteration by iteration in the IPM process.

Additionally, the scenario generation variant of IPM mentioned in Chapter 4 requires AMPL to supply the second order information twice in one IPM iteration. This is because the Hessian matrices of the active scenarios and the inactive scenarios cannot be computed in one single statement of AMPL.

For the above reasons, in the structure-exploiting method, one robust and efficient interface to AMPL is a must to set the structure and recall the information later.

References [32] and [40] have presented some ways of communicating information between AMPL and a solver. Briefly, for a nonlinear problem, AMPL tries to communicate with a solver by writing and reading a so-called .nl file. That is, after a model is built in AMPL, all the necessary information such as the constraints and objectives functions (it supports multi-objectives) can be saved in a .nl file. This file contains a complete problem description of the problem and most importantly, it would not introduce new rounding errors caused by producing the .nl file [1].

Once a .nl file is generated by calling command *write* in AMPL, it can be passed to the AMPL solver library ASL which can be linked to the solver. Then ASL will generate a workspace for this .nl file and its containing problem information. A solver can use AMPL routines from its library to access this workspace and consequently compute the essential information, i.e. constraints, objectives and first/second order derivatives.

Hence, all one requires is an interface between the solver and AMPL, which can convert the information from ASL format into the solver format and vice versa. With the use of such an interface, ASL routines can evaluate the functions as required and send the outcome back to the solver in its readable format. The process of this communication can be represented in Figure 6.1.

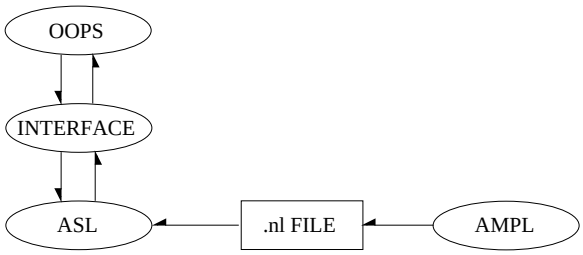

Figure 6.1: Solver and AMPL interaction

In a structure-exploiting IPM solver, such as OOPS [37], all the information stemming from ASL need to be deconstructed and re-assembled into a user-defined order. The re-assembled information should have a correct structure for the use of the structure-exploiting solver, e.g. the reordered data should correspond to the correct node structure in OOPS, in order to be allocated to the correct processor.

In order to achieve this correct structure setting, the solver should ask for additional row/-column information and a user-defined structure file for the particular problem. Fortunately, AMPL can generate a so-called .row file and a .col file at the same time when getting the .nl file for a problem. AMPL saves the names of all the constraints in ASL format in a .row file, while the names and the order of variables are saved in a .col file. Note that these orders are different from the orders used in the solver OOPS, and this structure is only used in evaluating

[1] AMPL can generate an MPS file for LP problem, but it adds new rounding error due to converting the numbers into 12-column fields [32].

function by AMPL solver library. However, these two files provide useful information when the interface converts the format.

The user-defined structure file, which is named as a .str file, requires additional explanation as follows. Firstly, this .str file defines how many subproblems this problem should have, and then it decomposes the entire problem into the subproblems by splitting variables and constraints. The .str file gives a description of which variables/constraints belong to which block, as well as the way these variables/constraints should be ordered in each block. The latter is extremely important in the implementation in Chapter 3, as it can show that the matrices given by the sub-block can have a augmented-system form after reordering.

To clarify the use of a .str file, the following Example 6.1 is used to demonstrate how a .str file looks like. This example is the .str file corresponding to the problem (4.48).

Example 6.1.

```
Matrix A DblBordDiag 2

  Border
    col
      x
    row
  End Border

  SubMatrix Sparse
    col
      y
    row
      Constraint_1
  End SubMatrix

  SubMatrix Sparse
    col
      z
    row
      Constraint_2
  End SubMatrix

End Matrix
```

Example 6.1: Small example of .str file.

As shown in Example 4.48, variable x is allocated as a border of the whole problem, while there are two sub-blocks given by y and the first constant, z and the other constant, respectively. Note that this example is quite easy, but briefly interprets .str file. It can be much more

sophisticated in practice, such as defining the saddle point structure for the DC-SCOPF problem in Chapter 3.

Generally, a stochastic problem can be decomposed by its scenarios, and a deterministic case can be decomposed by its specific structures, e.g. the SCOPF problems shown in the thesis. It is worth mentioning that each subproblem would correspond to a diagonal block appearing in (3.17).

The full process with these three additional files can be illustrated in the following figure:

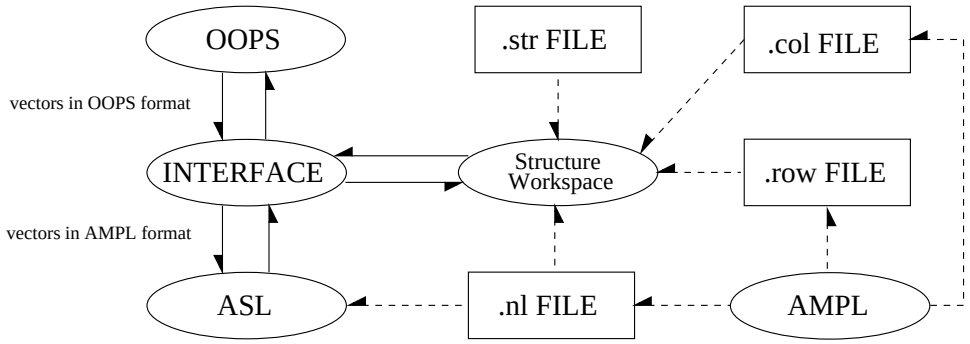

Figure 6.2: Solver and AMPL interaction.

The diagram in Figure (6.2) indicates the role of the interface; it should convert the information in OOPS format into ASL format and vice versa. In transactions, the solid line denotes the information needed to be recalculated in every IPM iteration. OOPS updates the current point x in its own format and sends it to the interface. In the interface, the programme would ask for the structure information, which is determined by the .str, .row and .col files, to check how to reorder the information into ASL format and then does the conversion. After having a readable file for ASL, ASL can evaluate the functions as user required. Finally, the backward process in the interface is to convert the AMPL evaluations into OOPS readable format, using necessary information from the structure workspace.

However, this process can be very expensive if it repeats in each call of the evaluation of function. Each time it needs to go through three files to find the correct structure. In order to overcome this difficulty, a remedy is to save the 'switching condition' of the structure information in its workspace. For instance, for each matrix in a NLP problem, such as the Jacobian or Hessian at each iterates, its nonzero pattern is unknown for OOPS before the routine of ASL evaluates the corresponding function. After the first call of a function evaluation in ASL, the nonzeros are returned in a sparse format of ASL. Then, loops are applied to find the corresponding place in OOPS format for these nonzeros, i.e. which block it belongs to and its local row and column indices.

Additionally, for each nonzero in OOPS format, it is given a unique index to record its place in the nonzero pattern in ASL format. These numbers should have one-to-one correspondence since the numbers of nonzeros are identical in both formats. Then, from the later call of ASL

evaluation for the same function, the loop mentioned above can be abandoned. Instead, the saved unique indices can be exploited to find the correct value in ASL format much more efficiently.

As a result, the dashed lines in Figure 6.2 only need to be applied once and the unique one-to-one index gives a fast way to switch the formats between OOPS and ASL.

Last but not least, the only problem left is that the scenario-generation-based IPM Algorithm 4.3 requires computing the Hessian of the Lagrangian function (2.32) twice in one IPM iteration. This is due to the fact that the last diagonal block matrix $\mathbf{W}_G = \bigtriangledown^2_{x_G x_G} L$ in (4.5), which refers to the first stage variables and global constraints, would be used in building the Schur complement matrix for the reduced problem. Hence it should be computed from the reduced problem with active scenarios only. Contrarily, ASL would return the Hessian for the full problem where $y_i \neq 0$, for all $i \in \mathcal{C}_I$, which leads to a different block $\mathbf{W}_G$.

In order to prevent this effect, the last block of Hessian needs to be updated via a second evaluation of Hessian for the reduced problem. This can be easily achieved by setting y_i to zero, for all $i \in \mathcal{C}_I$. Note that other blocks do not require the second evaluation, since local variables from a scenario are independent from the ones in other scenario. Therefore, in the interface between OOPS and ASL, only the last block of Hessian is updated after the second evaluation in ASL, although the routine in ASL would compute the full Hessian again (with $y_i = 0, i \in \mathcal{C}_I$).

This process can be illustrated in Figure 6.3.

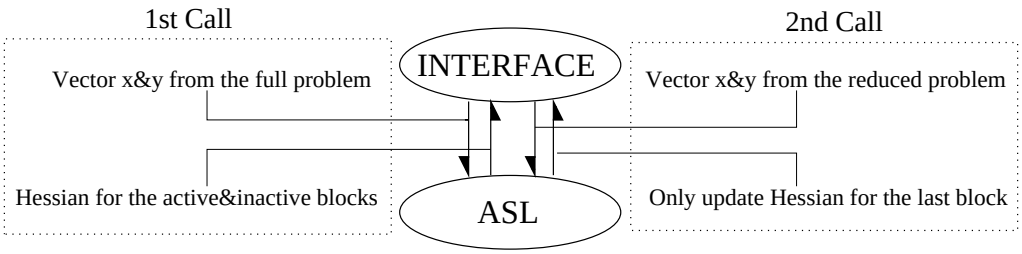

Figure 6.3: Solver and AMPL interaction

6.3 Introduction to a Solver for Linear Systems

There are many different solvers for linear systems, whose aim is to solve different kind of linear equations $\mathbf{A}x = \mathbf{b}$, corresponding to the features of the system, e.g. if matrix $\mathbf{A}$ is symmetric or not. Among all these solvers, MA27 and MA57, which come from HSL Archive [85], are chosen to solve the incurred sparse symmetric indefinite linear systems at each IPM iteration in our structured implementation. It is worth mentioning that MA27 is the solver used in IPOPT [70].

Both of these two open sources, MA27 and MA57, solve the linear system (2.59) incurred

at each interior point iteration by $\mathbf{LDL}^\top$ factorization. The main differences between these two packages, is that MA57 is the next generation of MA27 and it supports solving the system $\mathbf{AX} = \mathbf{B}$, where $\mathbf{B}$ is a matrix contains several right-hand-side vector $\mathbf{b}$. It can find the solution $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$ for the matrix system in one single statement, while MA27 needs to solve it column by column via a loop of statement.

Most of all, MA57 supports the partial solve function of the $\mathbf{LDL}^\top$ factorization. That is, one can have the access to the partial solution $\mathbf{L}^{-1}\mathbf{b}$ and $\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{b}$. This is very important in the structure-exploiting IPM implementation in OOPS, since it suggests using partial solutions to build the contribution of the Schur complement matrix $\mathbf{C}$ via (3.33). More details about the structured IPM implementation can be found in Chapter 3 and [37].

Generally, using these packages is a double-edged sword in our implementation. The advantages are that both codes support 2×2 block pivots, and as a result the factorization should be more stable than when only 1×1 pivots are applied. Moreover, both codes are widely tested and hence are robust enough. However, this may also be a weakness as well, since users are not able to apply dynamic regularization suggested by Altman and Gondzio in [9]. This may cause some numerical problems in the structure-exploiting algorithm, such as the wrong inertia problem mentioned in Chapter 4. Even it can be fixed easily by adding the regularization iteratively similar to what IPOPT [70] suggests, the trial-and-error procedure can be quite inefficient in the structure-exploiting algorithm. Once the problem happens at the Schur complement matrix, algorithm must forgo all the factorizations and the re-assembling of the Schur complement matrix.

In comparison with the interface to AMPL, the interface between OOPS and MA27/MA57 is much easier to be implemented. The main problem of MA27/MA57 is that they only support the decision matrix $\mathbf{A}$ in a sparse form but not the corresponding solution vector $\mathbf{x}$ and right-hand-side $\mathbf{b}$. Therefore, in the interface, we need to convert the sparse vector $\mathbf{b}$ into a dense vector and send it into the linear system solver, and convert the outcome solution $\mathbf{x}$ to the sparse form. Another problem is that it could not apply dynamic regularization as mentioned in the previous paragraph. Hence the interface and other codes must be modified to meet the requirements.

6.4 Conclusion

Generally, in order to solve the nonlinear problems, OOPS has been enhanced to interface with AMPL and other solvers of linear systems. The new interface for setting up nonlinear problems is a more robust method compared to the previous interface which only supports linear problems. It works well throughout the research and its performance is satisfactory. Additionally, the use of MA27/MA57 gives us an alternative to see how the algorithm works. Although they may be slower than the default OOPS routines, as described earlier, they are

useful tools for debugging and benchmarking issues.

Chapter 7

Conclusions

In this chapter, all the works presented in the thesis are summarized. Additionally, some possible research extensions are discussed in the end of this thesis.

7.1 Research Summary

The main focus of this thesis is how to solve the security constrained optimal power flow problem by exploiting its structure in interior point methods. In the second chapter, it was shown that this problem involves a base-case optimal power system operation plan and also requires the whole system to survive under certain predetermined security contingencies.

Accordingly, there are several ways to describe SCOPF problems and we apply a deterministic approach to build the model, which means there is no probability involved in the model. This deterministic SCOPF problem is a nonlinear programming problem, where real power generation and voltages levels at PV buses are the main controllable variables. It requires all the flow limits and Kirchhoff's laws to be satisfied under all the given contingency scenarios and the objective of the problem is to find the base-case operation plan so that the real power generation cost is minimized. Based on how large the size of the transmission network is and how many predetermined contingencies are required, the corresponding SCOPF problem with n-1 security criteria can result in an incredibly large mathematical model. In fact, a SCOPF problem in the real world can be very extensive and complicated, e.g Figure 7.1 (from GENI website [84]) indicates the full European transmission network.

Like many other structured problems, the SCOPF model consists of a small core structure that is repeated many times. Each small core is defined by a given contingency which is required to be included in the model to maintain the power system. It can be very expensive to solve such a large-scale model if no further techniques have been applied. The use of a structure-exploiting interior point method is suggested to solve this kind of problems.

The first approach suggested in this thesis is specialized for the DC-SCOPF model, which

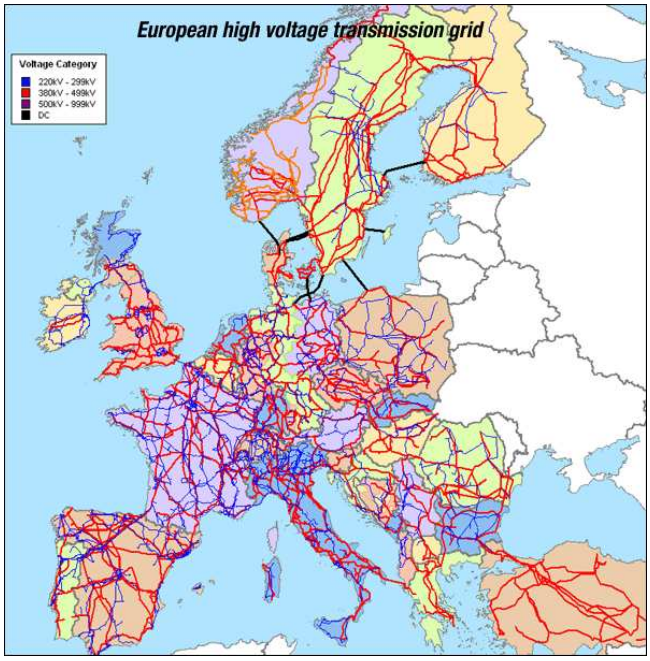

Figure 7.1: European high voltage transmission grid

is a linearized variant of the standard AC-SCOPF model, given several assumptions. In this DC-SCOPF model, most of the matrices that need to be factorized are constant between IPM steps. As a result, most of the factorizations and the corresponding back-solve operations are only required to be performed once throughout the whole IPM process. The numerical results state that this approach can successfully speed up the entire solution process. However, one major disadvantage is that this approach demands much more storage, in order to store partial computation results for later use.

Observing that the major computational effort consists of forming the Schur complement products in the structure exploiting interior point algorithm, a preconditioned iterative method is applied to overcome this difficulty. Two different schemes are suggested to pick a good and robust preconditioner for DC-SCOPF problems, based on how to add active contingencies into the preconditioner. Numerical results of these two approaches indicate that they are much faster than previous structure exploiting methods, even with less memory consumption.

The next main contribution of this thesis goes to the structure exploiting IPM for nonlinear programming, in order to solve the AC-SCOPF problems. A line search interior point algorithm is implemented to solve NLP problems. This line search method consists of a two-dimensional filter which is used to guarantee that the new iterates can deliver a satisfactory reduction on either the objective or the constraint violations.

Although the nonlinear AC-SCOPF problems with all required contingencies can be solved by such a filter line search IPM, the numerical experiments demonstrate that only a few contingency scenarios contain active constraints at the optimal solution which is also known as the operation plan for the power system. These dominant contingencies are considered as active, as

they can decide the solution of the whole problem. In other words, the solution to the reduced SCOPF problem defined by all these critical scenarios is exactly the same as that of the problem including all scenarios. The greatest advantage of using such a reduced SCOPF problem defined by all active contingencies is that the solution process can be boosted due to the much smaller problem size. However, the major difficulty is how to find such active contingencies.

Generally, an iterative SCOPF solution technique is applied to solve the SCOPF problems. It detects the violated contingencies from the current solution and then restarts with the updated SCOPF model by adding the violated contingencies. Unlike the work of others, we merge the iterative solution techniques into the IPM process. The contingency analysis step which is used to find violated contingencies is now applied between IPM iterates. We observe that active scenarios would have a bigger effect on determining the step size of the Newton's direction. The step size generated by an active scenario can easily block others' step since its corresponding step size is much smaller than others'. In other words, the corresponding components of the direction for the active scenario make it easier to reach the boundary and consequently its constraints appear tighter than others. Therefore, a necessary condition, based on the observation of the step sizes for the Newton's step, is presented to indicate how to find an active contingency during the IPM process.

This scenario generation interior point approach starts from solving the AC-OPF model and then generates all the potentially active contingencies dynamically when needed, hence it results in calling the optimization solver only once throughout the whole process. This new approach is related to the warm-start issue of the interior point method, which is discussed in the corresponding chapter with some theoretical support. The numerical results from this scenario generation approach also show its better performance on solving the AC-SCOPF problem, compared with solving the full SCOPF model by default structure-exploiting NLP IPM. It can also successfully identify all the essential contingencies for the AC-SCOPF problems, and additionally, the number of detected contingencies is relatively small. Most important of all, the reduced problem built with fewer scenarios has a more stable numerical behavior, and it can solve a test problem which cannot be solved by the default structure-exploiting NLP IPM.

The final contribution of this research endeavors to identify these active contingencies before starting to solve the AC-SCOPF problem. If all the active contingencies can be obtained in advance, there is no need to apply contingency analysis steps in solving the AC-SCOPF problem. That is, the additive contingency analysis step between IPM iterates or between the iterative solution techniques can be discarded by solving the reduced SCOPF problems with all the active contingencies.

To achieve this goal, we attempt to get such information for the AC-SCOPF problem from the solution to its corresponding AC-OPF problem or DC-SCOPF problem. Some heuristic algorithms, which exploit the use of AC-OPF or DC-SCOPF approximation, are designed to rank all the contingency candidates by different criteria. Hence we can rank all the potential

contingencies and produce a measure to figure out all the active contingencies.

However, all the numerical results indicate that none of these heuristic approaches can successfully identify all the active contingencies for the IEEE 24-bus system. Despite the fact that some of them can rank all the necessary contingencies in the top half; it seems that the ranking schemes can hardly distinguish the useful contingencies from others in the middle of the ranking. In other words, the difference between the ranked objectives is too small to decide whether a contingency is potentially active or not. Besides all these disadvantages mentioned above, the numerical experiments and corresponding analysis also demonstrate the bottlenecks of the use of AC-OPF and DC-SCOPF approximation. That is, the solution to the AC-OPF cannot handle the switches of the power generators in certain contingency cases, while DC-SCOPF approximations cannot identify the constraint violations caused by the reactive power and voltage issue.

7.2 Future Work

The main purpose of this PhD research is to seek a powerful tool to solve the n-1 AC-SCOPF problems. In general, the SCOPF problems with n-1 security requirement are composed by numerous repetition of small structures. This kind of structured problems is always the case for the large-scale problems, especially for the stochastic programming problems with considerable numbers of scenarios.

Although the new scenario-generation interior point algorithm is designed to solve the non-linear deterministic AC-SCOPF problem, there is no doubt that this approach can be applied to other problems with the same structure. One interesting extension is how this algorithm can be modified to solve a generic stochastic programming problem. The greatest difficulty of this extension is that it can hardly be achieved since all the scenarios may become active. However, applying similar method to find the useful scenarios to build the preconditioner for iterative methods, such as the DC-SCOPF solution approach presented in Chapter 3, is worth a try and it can be applied to stochastic programming definitely. As a result, we can include wind generation as a stochastically changing generation into the SCOPF framework. This is also a widely discussed topic in recent decades and it seems this research topic will remain open and popular for a while to come.

The numerical experiments indicate that this new scenario-generation-based approach can successfully solve the problem with fewer scenarios, and most important of all, the reduced problem displays a more stable numerical behavior. However, the theoretical analysis requires further improvement to cover the convergence condition and the computational complexity of the algorithm.

Finally, one piece of good news is that exascale computing facilities will be implemented in the near future, and consequently, people will have much more powerful high-performance

computation tools to deal with the large or extremely large problems. However, a good parallel support always plays a very important role in high-performance computing. The main challenge of the parallel structure-exploiting IPM algorithm is how to assemble and solve the Schur complement system efficiently. As mentioned in Chapter 3 and Chapter 4, all the new approaches presented in this thesis mainly speed up the solution process by forming the Schur complement matrix more efficiently. It is worth seeking a more efficient way to deal with the Schur complement system.

Additionally, it is worth mentioning that the framework of structure exploiting IPM itself supplies a very friendly parallelizable environment and the linear IPM algorithm presented in this thesis is a parallel implementation. Conversely, the nonlinear IPM implementation has not been optimized under the parallel condition. It is obvious that the scenario generation IPM implementation would destroy the synchronous behavior since it is a two-stage algorithm: solve the reduced model and then test other unused scenarios. One quick idea for future research regarding this issue is the use of an asynchronous parallel framework. That is, keep solving the reduced problem on particular processors while testing the feasibilities of other scenarios. Once a violated contingency is identified, the algorithm can be restored with the barrier parameter μ corresponding to the violated contingency. This prospective research direction may also be a very interesting topic.

Appendix A

First Appendix

GMRES Algorithms

Algorithm A.1: GMRES Algorithm

Step 1. Initialization:
Given an initial guess u_{init}, compute $r_{init} = s - \mathcal{H}u_{init}$, $\beta := \|r_{init}\|_2$, $v_1 := r_{init}/\beta$.

Step 2. Main Loop:
For $j = 1, 2, \ldots,$ do:
$w_j := \mathcal{H}v_j$
For $i = 1, 2, \ldots, j,$ do
$h_{i,j} := w_j^\top v_i$
$w_j := w_j - h_{i,j}v_i$
end-do
$h_{j+1,j} := \|w_j\|_2$, if $h_{j+1,j} = 0$, set $m := j$ and go to **Step 3**
$v_{j+1} := w_j/h_{j+1,j}$
end-do

Step 3. Find Solution:
Define $\mathbf{V}_m := (v_1, \ldots, v_m)$, $e_1 := (1, 0, \ldots, 0)^\top$ and $\bar{H}_m := \{h_{i,j}\}_{1 \le i \le m+1, 1 \le j \le m}$
Find $u_m := u_{init} + \mathbf{V}_m y_m$, where y_m minimizes $\|\beta e_1 - \bar{H}_m y\|_2$

Algorithm A.2: GMRES(m) Algorithm

Step 1. Initialization:
Given an initial guess u_{init}, compute $r_{init} = s - \mathcal{H}u_{init}$, $\beta := \|r_{init}\|_2$, $v_1 := r_{init}/\beta$.

Step 2. Main Loop:
For $j = 1, 2, \ldots, m$, do:
$w_j := \mathcal{H}v_j$
For $i = 1, 2, \ldots, j$, do
$h_{i,j} := w_j^\top v_i$
$w_j := w_j - h_{i,j}v_i$
end-do
$h_{j+1,j} := \|w_j\|_2$, if $h_{j+1,j} = 0$, set $m := j$ and go to **Step 3**
$v_{j+1} := w_j/h_{j+1,j}$
end-do

Step 3. Find Solution:
Define $\mathbf{V}_m := (v_1, \ldots, v_m)$, $e_1 := (1, 0, \ldots, 0)^\top$ and $\bar{H}_m := \{h_{i,j}\}_{1 \le i \le m+1, 1 \le j \le m}$
Find $u_m := u_{init} + \mathbf{V}_m y_m$, where y_m minimizes $\|\beta e_1 - \bar{H}_m y\|_2$

Step 4. Restart:
Compute $r_m := s - \mathcal{H}u_m$, if $\|r_m\|_2 < tol$ **STOP**
else set $r_{init} := r_m$, $\beta := \|r_m\|_2$, $v_1 := r_m/\beta$ and **Restart** go to **Step 2**

Bibliography

[1] Fundamental review: Update and consultation report. 23 April 2010, available at: http://www.nationalgrid.com/uk/Electricity/Codes/gbsqsscode/LiveAmendments/.

[2] National electricity transmission system seven year statement. May 2011, available at: http://www.nationalgrid.com/uk/Electricity/SYS/current/.

[3] Operational security. 24 June 2004, available at: https://www.entsoe.eu/publications/system-operations-reports/operation-handbook/.

[4] White paper. optimal power flow. basic requirements for real-life. problems and their solutions. July 2012, available at: http://www.ieee.hr/_download/repository/Stott-Alsac-OPF-White-Paper.pdf.

[5] Working group report 1: International benchmarking report. 23 April 2010, available at: http://www.nationalgrid.com/uk/Electricity/Codes/gbsqsscode/LiveAmendments/.

[6] G. Al-Jeiroudi. *On Inexact Newton Directions in Interior Point Methods for Linear Optimization*. PhD thesis, University of Edinburgh Graduate School of Mathematics, 2008.

[7] O. Alsac, J. Bright, M. Prais, and null. Further developments in LP-based optimal power flow. *Power Systems, IEEE Transactions on*, 5(3):697–711, Aug.

[8] O. Alsac and B. Stott. Optimal load flow with steady-state security. *Power Apparatus and Systems, IEEE Transactions on*, PAS-93(3):745 –751, May 1974.

[9] A. Altman and J. Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. *Optim. Methods Softw.*, 11/12(1-4):275–302, 1999. Interior point methods.

[10] M. Balandin, O. Chernyshev, and E. Shurina. Analysis of methods for solving large-scale non-symmetric linear systems with sparse matrices. In V. Malyshkin, editor, *Parallel*

Computing Technologies, volume 1277 of *Lecture Notes in Computer Science*, pages 336–343. Springer Berlin / Heidelberg, 1997.

[11] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[12] H. Y. Benson, R. J. Vanderbei, and D. F. Shanno. Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions. *Comput. Optim. Appl.*, 23(2):257–272, 2002.

[13] F. Bouffard, F. D. Galiana, and J. M. Arroyo. Umbrella contingencies in security constrained optimal power flow. In *15th Power Systems Computation Conference (PSCC 05*, 2005.

[14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[15] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM J. Optim.*, 9(4):877–900 (electronic), 1999. Dedicated to John E. Dennis, Jr., on his 60th birthday.

[16] R. H. Byrd, J. Nocedal, and R. A. Waltz. KNITRO: An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization*, volume 83 of *Nonconvex Optim. Appl.*, pages 35–59. Springer, New York, 2006.

[17] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel. Contingency filtering techniques for preventive security-constrained optimal power flow. *Power Systems, IEEE Transactions on*, 22(4):1690 –1697, Nov 2007.

[18] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel. Interior-point based algorithms for the solution of optimal power flow problems. *Electric Power Systems Research*, 77(5-6):508 – 517, 2007.

[19] F. Capitanescu and L. Wehenkel. A new iterative approach to the corrective security-constrained optimal power flow problem. *Power Systems, IEEE Transactions on*, 23(4):1533 –1541, Nov 2008.

[20] J. Carpentier. Contribution à l'Étude du dispatching Économique. *Bulletin de la Société Françise des Électricit*, 3:431–447, 1962.

[21] G. Chen, M. Erin, and O. Dominique. Bounds on eigenvalues of matrices arising from interior-point methods. Technical report. Cahier du GERAD G-2012-42, GERAD, Montreal, Canada, 2012.

[22] M. Colombo, J. Gondzio, and A. Grothey. A warm-start approach for large-scale stochastic linear programs. *Mathematical Programming*, 127(2):371 – 397, 2011.

[23] M. Colombo and A. Grothey. A decomposition-based crash-start for stochastic programming. *Computational Optimization and Applications*, pages 1–30, 2013.

[24] M. Colombo, A. Grothey, J. Hogg, K. Woodsend, and J. Gondzio. A structure-conveying modelling language for mathematical and stochastic programming. *Mathematical Programming Computation*, 1:223–247, 2009. 10.1007/s12532-009-0008-2.

[25] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Springer Verlag, Heidelberg, New York, 1992.

[26] F. E. Curtis, O. Schenk, and A. Wächter. An interior-point algorithm for large-scale nonlinear optimization with inexact step computations. *SIAM Journal on Scientific Computing*, 32(6):3447 – 3475, 2010.

[27] C. Dent, L. Ochoa, G. Harrison, and J. Bialek. Efficient secure AC OPF for network generation capacity assessment. *Power Systems, IEEE Transactions on*, 25(1):575 –583, Feb 2010.

[28] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff. A set of level 3 basic linear algebra subprograms. *ACM Trans. Math. Softw.*, 16(1):1–17, Mar 1990.

[29] M. E. El-Hawary and G. S. Christensen. *Optimal economic operation of electric power systems.* Mathematics in science and engineering: v. 142. New York ; London : Academic Press, 1979.

[30] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Numerical Analysis Report NA/171, Dundee University, 1997.

[31] R. Fourer, D. M.Gay, and B. W.Kernighan. *AMPL: A Modeling Language for Mathematical Programming, 2nd Ed.* Curt Hinrichs, Canada, 2003.

[32] D. M. Gay. Hooking your solver to AMPL. Technical Report 97-4-06, Computing Sciences Research Center, Bell Laboratories, 1997.

[33] J. Gondzio. Hopdm (version 2.12) a fast lp solver based on a primal-dual interior point method. *European Journal of Operational Research*, 85(1):221 – 225, 1995.

[34] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587 – 601, 2012.

[35] J. Gondzio and P. González-Brevis. A new warmstarting strategy for the primal-dual column generation method. Technical Report ERGO 12-007, University of Edinburgh, 2012.

[36] J. Gondzio and A. Grothey. Parallel interior-point solver for structured quadratic programs: application to financial planning problems. *Ann. Oper. Res.*, 152:319–339, 2007.

[37] J. Gondzio and A. Grothey. Exploiting structure in parallel implementation of interior point methods for optimization. *Comput. Manag. Sci.*, 6(2):135–160, 2009.

[38] N. I. M. Gould and D. Orban. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. Technical report, ACM Transactions on Mathematical Software, 2001.

[39] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh. The IEEE reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee. *Power Systems, IEEE Transactions on*, 14(3):1010 –1020, Aug 1999.

[40] A. Grothey. *Decomposition Methods for Nonlinear Nonconvex Optimization Problems*. PhD thesis, University of Edinburgh Graduate School of Mathematics, 2001.

[41] J. A. Harrison. *The essence of electric power systems*. Essence of engineering. London : Prentice Hall, 1996., 1996.

[42] M. Y. He and A. L. Tits. Infeasible constraint-reduced interior-point methods for linear optimization. *Optimization Methods & Software*, 27(4/5):801 – 825, 2012.

[43] K. Karoui, L. Platbrood, H. Crisciu, and R. Waltz. New large-scale security constrained optimal power flow program using a new interior point algorithm. In *Electricity Market, 2008. EEM 2008. 5th International Conference on European*, pages 1 –6, May 2008.

[44] G. Karypis and V. Kumar. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, Sept 1998.

[45] C. T. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, Philadelphia, PA,, 1995.

[46] D. S. Kirschen and G. Strbac. *Fundamentals of power system economics / Daniel Kirschen, Goran Strbac.* Chichester, West Sussex, England ; Hoboken, NJ : John Wiley & Sons, c2004., 2004.

[47] J. Korzak. Convergence analysis of inexact infeasible-interior-point algorithms for solving linear programming problems. *SIAM Journal on Optimization*, 11(1):133–148, 2000.

[48] L. Liu, X. Wang, X. Ding, and H. Chen. A robust approach to optimal power flow with discrete variables. *Power Systems, IEEE Transactions on*, 24(3):1182 –1190, Aug. 2009.

[49] C.-N. Lu and M. Unum. Network constrained security control using an interior point algorithm. *Power Systems, IEEE Transactions on*, 8(3):1068 –1076, Aug 1993.

[50] A. Marano-Marcolini, F. Capitanescu, J. Martinez-Ramos, and L. Wehenkel. Exploiting the use of DC SCOPF approximation to improve iterative AC SCOPF algorithms. *Power Systems, IEEE Transactions on*, 27(3):1459 –1466, Aug 2012.

[51] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[52] J. Momoh, R. Koessler, M. S. Bond, B. Stott, D. Sun, A. Papalexopoulos, and P. Ristanovic. Challenges to optimal power flow. *Power Systems, IEEE Transactions on*, 12(1):444–455, 1997.

[53] A. Monticelli, M. V. F. Pereira, and S. Granville. Security-constrained optimal power flow with post-contingency corrective rescheduling. *Power Engineering Review, IEEE*, PER-7(2):43–44, 1987.

[54] J. Nocedal and S. J.Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[55] T. Overbye, X. Cheng, and Y. Sun. A comparison of the ac and dc power flow models for lmp calculations. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, page 9 pp., Jan 2004.

[56] C. Petra and M. Anitescu. A preconditioning technique for schur complement systems arising in stochastic optimization. *Computational Optimization and Applications*, 52:315–344, 2012.

[57] W. Qiu, A. Flueck, and F. Tu. A new parallel algorithm for security constrained optimal power flow with a nonlinear interior point method. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 447 – 453 Vol. 1, June 2005.

[58] V. Quintana, G. Torres, and J. Medina-Palomo. Interior-point methods and their applications to power systems: a classification of publications and software codes. *Power Systems, IEEE Transactions on*, 15(1):170 –176, Feb 2000.

[59] V. Radziukynas and I. Radziukyniene. Optimization methods application to optimal power flow in electric power systems. In J. Kallrath, P. Pardalos, S. Rebennack, and M. Scheidt, editors, *Optimization in the Energy Industry*, Energy Systems, pages 409–436. Springer Berlin Heidelberg, 2009.

[60] M. Rider, C. Castro, M. Bedrinana, and A. Garcia. Towards a fast and robust interior point method for power system applications. *Generation, Transmission and Distribution, IEE Proceedings-*, 151(5):575 – 581, Sept 2004.

[61] H. M. Ryan. *High voltage engineering and testing.* IEE power series: 17. Stevenage : Peregrinus on behalf of the Institution of Electrical Engineers, c1994., 1994.

[62] Y. Saad. *Iterative methods for sparse linear systems.* SIAM, Philadelphia, PA,, 2003.

[63] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[64] B. Stott, O. Alsac, and A. Monticelli. Security analysis and optimization. *Proceedings of the IEEE*, 75(12):1623 – 1644, Dec 1987.

[65] B. Stott, J. Jardim, and O. Alsac. DC power flow revisited. *Power Systems, IEEE Transactions on*, 24(3):1290–1300, Aug.

[66] R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optim. Methods Softw.*, 11/12(1-4):451–484, 1999. Interior point methods.

[67] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appl.*, 13(1-3):231–252, 1999. Computational optimization—a tribute to Olvi Mangasarian, Part II.

[68] L. Vargas, V. Quintana, and A. Vannelli. A tutorial description of an interior point method and its applications to security-constrained economic dispatch. *Power Systems, IEEE Transactions on*, 8(3):1315 –1324, Aug 1993.

[69] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1 – 31, 2005.

[70] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1, Ser. A):25–57, 2006.

[71] D. S. Watkins. *The matrix eigenvalue problem: GR and Krylov subspace methods.* SIAM, Philadelphia, PA, 2007.

[72] H. Wei, H. Sasaki, and R. Yokoyama. An application of interior point quadratic programming algorithm to power system optimization problems. *Power Systems, IEEE Transactions on*, 11(1):260 –266, Feb 1996.

[73] A. J. Wood and B. F. Wollenberg. *Power generation, operation, and control.* New York ; Chichester : J. Wiley & Sons, c1996., 1996.

[74] S. J. Wright. *Primal-dual Interior-Point Methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1997.

[75] L. Xie and H.-D. Chiang. A enhanced multiple predictor-corrector interior point method for optimal power flow. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1 –8, July 2010.

[76] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[77] http://www.fico.com/en/Products/DMTools/.

[78] http://pages.cs.wisc.edu/~swright/ooqp/.

[79] http://www.maths.ed.ac.uk/~gondzio/parallel/solver.html.

[80] http://www.ee.washington.edu/research/pstca/.

[81] http://www.gams.org.

[82] http://www.aimms.com/aimms.

[83] http://www.neos-server.org/neos/.

[84] http://www.geni.org/globalenergy/library/national_energy_grid/europe/europeannationalelectricitygrid.shtml.

[85] HSL(2011). A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk.