March 2021

Power System Optimization Methods: Convex Relaxation and Benders Decomposition

Minyue Ma
University of South Florida

Power System Optimization Methods: Convex Relaxation and Benders Decomposition

by

Minyue Ma

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Lingling Fan, Ph.D.
Zhixin Miao, Ph.D.
Nasir Ghani, Ph.D.
Tapas K. Das, Ph.D.
Bo Zeng, Ph.D.

Date of Approval:
Novmber 13, 2020

Keywords: alternating current optimal power flow, security constrained optimal power flow,
relaxation exactness, model predict control

Dedication

I dedicate this dissertation to my wife Danxu, I could not have done this without all of your love.

Acknowledgments

I would like to specially thank the following people, without whom I would not have achieved this goal. To my committee chair, Dr. Lingling Fan, who has tirelessly supported my endeavors. Thank you for always checking in on me, helping me out when I cannot find solutions, and teaching me to be a qualified independent researcher. I deeply appreciate you nurturing the potential that you saw in me. Dr. Zhixin Miao, thank you for inspiring and mentoring me. You have been my professional role model, and it is you who opened my eyes to the amazing possibilities of research in science. Dr. Zeng Bo, thank you for working with me to overcome difficult challenges. Your comments and feedback were extremely helpful with my dissertation studies. Thank you to Dr. Nasir Ghani, Dr. Tapas Das, and Dr. Yicheng Tu for their careful review of the dissertation and constructive comments.

Thank you to my Ph.D. cohort at the University of South Florida Yin Li, Yangku Xu, Yi Zhou, Miao Zhang, Li Bao, Zhengyu Wang, Ibrahim Alsaleh, and Abdullah Alassaf for inspiring me to be a better student and bring joy in the Smart Grid Power Systems Laboratory.

Last but not least, I am grateful to my family members who have supported me throughout this process. To my wife Danxu Ma, thank you for your love and steadfast support. I was not alone because I always have you by my side. To my son Max, you are my little laughter. The day you were born was the greatest moment of my life. I am highly in debt to my parents for supporting me with their best wishes. To my father, "It's like that with love, we may be close, we may be far, but your love still surrounds me... wherever you are."

Table of Contents

List of Tables

List of Figures

Abstract

Power system optimization methods are wildly used to solve power system problems. Engineers adopt different methods to keep the reliability and efficiency of the power system operation, planning and control. This dissertation focuses on the application and implementation of two optimization methods: Convex relaxation and Benders' decomposition.

The first part of the dissertation focuses on the application of convex relaxation to solve Alternating Current Optimal Power Flow (ACOPF) problems. In the completed work, a 3-node cycle based sparse convex relaxation is proposed to solve ACOPF problems. This method adds virtual lines in minimal chordless cycles to decompose each of them into 3-node cycles. By enforcing the submatrices related to 3-node cycles Positive Semi-Definite (PSD), the resulting convex relaxation has a tight gap. For the majority of the test instances, the resulting gap is as tight as that of a semi-definite programming (SDP) relaxation, yet the computing efficiency is much higher. Furthermore, to achieve the exactness of the convex relaxation, two algorithms are designed to decrease the relaxation gap. The first method is based on the convex iteration technique. It could help the proposed convex relaxation to achieve the exactness by enforcing all submatrices corresponding lines and virtual lines rank-1. The second method is based on the nonlinear programming formulation of ACOPF with the PSD matrix as the decision variable. In this method, the rank-1 PSD matrix constraint is reformulated to equality constraints: all 2×2 minors of the PSD matrix are zeros. The graph decomposition-based approach is implemented to reduce the computation burden.

In the second part of the work, the application of Benders' decomposition is investigated through two problems. The first problem is the Model Predict Control (MPC) problem for Modular Multilevel Converter (MMC). The objective of the MPC is to determine the best switching sequences for the submodules in the MMC to track the phase current references for multiple time horizons.

The MPC is formulated as a nonlinear mixed-integer programming (MIP) problem with the on/off status of submodules as binary decision variables and MMC dynamic states such as phase currents, circulating currents and submodule capacitor voltages as continuous decision variables. With a large number of submodules and a large number of time horizons, the dimension of the nonlinear MIP problem is difficult to handle. Our contribution is to formulate this problem and solve this problem using Benders' decomposition. In the second problem, an efficient Benders' decomposition strategy is designed to solve the Security Constrained DCOPF (DC-SCOPF) with generator response constraints. The major difficulties to solve such SCOPF are the large number of contingencies and non-convexity of the generator response constraints. In this work, Benders' decomposition strategies were investigated to seek an efficient computing. We formulate the generator response constraints via bilinear expressing, and adopt Benders' decomposition to decompose the problem into a master problem with multiple sub-problems, each associated with a contingency. Based on the case study results, the proposed method has faster computing speed compares with the traditional big-M based mixed-integer linear programming method.

This dissertation has led to three journal papers and one conference paper.

Chapter 1: Overview

1.1 General Introduction

As an essential part of the running of model society, the power system which is the network to generate, transmit, and use the electric power1.1, has been studied and developed over one hundred years. Nowadays, the power system has been one of the largest engineering systems in the world. It made hundreds of billions dollars revenue per year for electrical industry in united state. One of the main challenges in power system operation is ensure the reliability and security of the system. Meanwhile, the efficient power system operation is an important concern for the power system engineers, because it can contributes to decrease the resource consumption, ensuring sustainability with better planning, and increasing the economy benefits. To operate the power system reliably and efficiently, optimization methods are wildly used to solve power system problems.

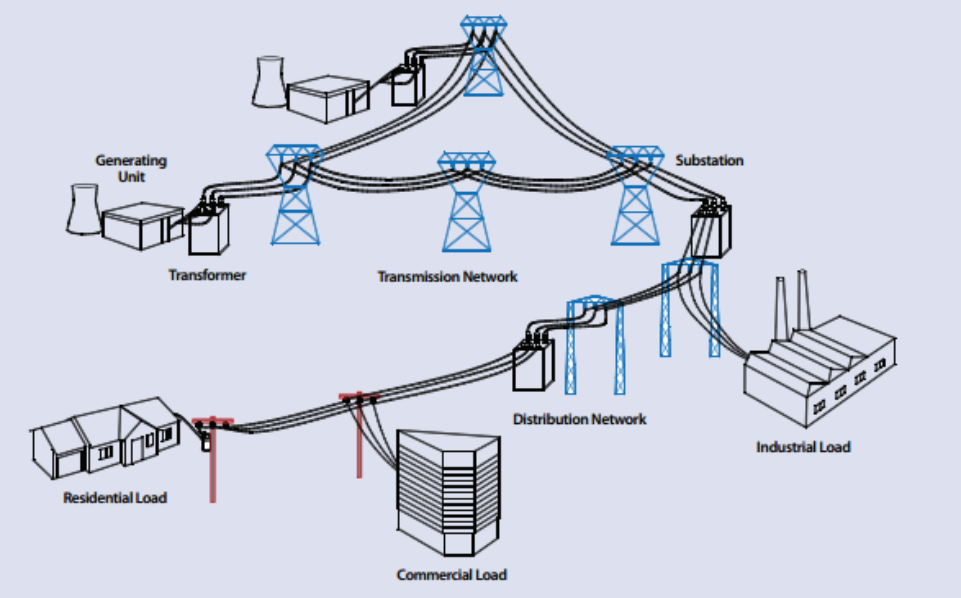


Figure 1.1: Power system structure. Reprinted from [1]. Permission is included in Appendix A.

Optimization methods have been used over the years for many power systems planning, operation, and control problems [2]. An power system optimization problem is a mathematical model which is proposed to minimize or maximize a objective function, and satisfy some constraints based on the physical requirements or decision making mechanism. In general, to solve real world power system problems, the mathematical formulations of problems have to be derived based on some assumptions, such as the Direct Current Optimal Power Flow (DCOPF) problem. However, even under these assumptions to simplify problems, it is still not easy to obtain solutions for large size problems. In general, real world power systems are large size, and complex because it includes many different units and operation requirements. To solve the power system optimization problem accurately and efficiently, several optimization methods have been used to deal with different formulations of problems, such as linear programming (LP), interior point (IP) method, quadratic programming (QP), decomposition technical, mixed integer programming (MIP) and so on. There are two methods attract a lot of interests in power system optimization researches. They are convex relaxation and Benders' decomposition.

In general, convex relaxation is implemented through relaxing some of the non-convex constraints and meanwhile extending the feasible region of the original problem, and the optimal value of the relaxation problem is a lower bound of the optimal value of the original problem [3]. The major application of convex relaxation is in Alternating Current Optimal Power Flow (ACOPF) problems.

As the power flow equations are nonlinear, ACOPF problem is non-convex. Traditionally, nonlinear programming methods have been applied to solve the problem [4]. The nonlinear programming methods essentially find a local optimal solution in the feasible region that satisfies the first-order optimality condition [5]. Examples presented in [5] indicate that local optima could occur due to disconnected feasible region, loop flow, an excess of real or reactive power, or large difference in voltage angles across lines. Thus, different initial point selections can result in different solutions. As the convex relaxation is guarantee to converge and find the lower bound of original problem, some new approaches to solve ACOPF can be developed based on it [6].

Benders' decomposition is proposed by J.F.Benders [7] in 1962. The major objective of this method is to simplify problems with complicating variables. Its fundamental idea is to separate the problem into a master problem and a subproblem. In the subproblem, complicated variables are considered as fixed value. Thus the subproblem could be solved easier. Iteratively, the dual variables which are solved from the subproblem that will be used to generate Benders' cuts and add them to the master problem. And then, the solution of the complicated variables is returned to the subproblem. This iteration process is repeated until the stop criteria is met.

When implementing Benders' decomposition in power system problems, it is hard to directly apply the classic Benders' decomposition formulation which is presented in [7]. In general, we need to choose the appropriate extended formulation of Benders' decomposition to accurately and effectively solve the problems [8]. Another important concern when using Benders' decomposition is how to formulate the problem models. Based on the research in [9], the problem formulation can directly impact the performance of Benders' decomposition.

This dissertation will cover topics about convex relaxation and Benders' decomposition. For convex relaxation, more efficient formulation for ACOPF problem is proposed, and algorithms to improve the exactness for ACOPF convex relaxation is investigated. For Benders' decomposition, two problems are considered: model predict control problem for modular multi-level converter and security constrained optimal power flow problem. In next sections, some backgrounds will be provided for motivation of research.

1.2 Alternative Current Optimal Power Flow and Convex Relaxation

The first problem that will be discussed in this dissertation is the ACOPF problem. This problem is first introduced by Carpentier in 1962 [10]. The objective function is to minimize generation cost or power loss. The exact AC power flow equations are considered in the problem. Full constraints are related to power grid physical characteristics, component limits, and network operation limits. Depending on the practical requirements, some extra constraints may be included, such as security constraints or stability constraints. As the power flow equations are nonlinear, ACOPF is non-convex. Traditionally, nonlinear optimization solving methods, such as Newton-type method [11] and interior point method [12], have been applied to solve the problem. These

methods essentially find a local optimal solution in the feasible region that satisfies the first-order optimality condition. These local optima could occur because of disconnected feasible region, loop flow, an excess of real or reactive power, or large difference in voltage angles across lines.

Conventionally, to avoid the major disadvantages which is brought by the nonlinear AC power flow equations, the ACOPF will be simplified to directly current optimal power flow problem (DCOPF). In DCOPF, the exact AC power flow constraints will be linearized to DC power flow constraints. This simplification is based on three major assumptions: 1) all bus voltage magnitude is 1 per unit; 2) all bus voltage angle is very small; 3) the transmission line resistor is ignorable. DCOPF has better computational efficiency than ACOPF, but it ignored some important propriety in the real power flow such as the reactive power transmission and might lead big error solutions for stressed system.

In recent years, convex relaxation has attracted a lot attentions on solving ACOPF problem, because it is capable to find provable lower bound of the solution to the original ACOPF problem. The two major relaxation techniques are semi-definite programming (SDP) relaxation and second-order cone programming (SOCP) relaxation. SDP relaxation was first applied to solve ACOPF in Bai et al [13]. SOCP was proposed in Jabr for radial networks [14]. SDP relaxation of ACOPF has shown to be a very strong convex relaxation to be original non-linear formulation. Nevertheless, the disadvantage of SDP is its expensive computational cost. Comparing with the SDP relaxation, SOCP has better computational efficiency. But its relaxation gap tend to be higher, especially for the mesh network.

Find a faster and more accurate method to solve ACOPF problem is important, because it could efficiently reduce the cost in power system. Based on the study in [15], even a 5% computation efficiency improvement could lead billions dollars saving in each year. Thus, the first part of this dissertation will focus on increasing the computation efficiency of the convex relaxation method, and develop the computational strategy to decreasing the relaxation gap, so that leads to high quality, near global optimal solutions.

1.3 Exactness of Convex Relaxation

Though it has been studied that SDP relaxation can give global optimum for many IEEE test systems while the solutions are feasible to the original ACOPF problems (termed as "SDP exact") in [16], in some other cases, SDP relaxation leads to solutions not feasible or SDP inexact [5,17,18]. Thus, research efforts have been devoted to achieve SDP exactness, e.g., [19,20].

The exactness conditions for SDP and SOCP relaxations are presented in [6]. Research has been conducted to achieve exactness for convex relaxation through exploiting the exactness conditions. In [19,20], objective functions are modified to include penalty related to the rank-1 constraint. [21] treats an ACOPF problem as an SDP relaxation problem and a non-convex rank-1 feasible region mapping problem. Alternating direction method of multipliers (ADMM) iterative procedure is then applied. In [22], the exactness constraint is reformulated as minor constraints and approximated by convex constraints. A strengthened SOCP relaxation of ACOPF is then solved. [23] proposes an convex iteration algorithm to solve a convex problem with a regularization term related to the maximal eigenvalue of the full PSD matrix. With the regularization term achieving zero, the solution achieves global minimum. In [24], the non-convex OPF branch flow equation is decomposed into SOCP constraint and a non-convex constraint related to the difference of two convex functions. The concave term is then approximated by linear functions and updated in each iteration. A sequential convex optimization method is implemented to carry out the iteration.

The approaches above lead to exact solutions in many cases. However, large gaps are still observed for special cases [20].

1.4 Model Predict Control for Power Electronic Application

As an advanced control method, MPC is very successful on its application for the control of power converters [25]. Its basic principle is to generate a system dynamic model based minimizing optimization problem, and provide the solutions to the controller for driving the system to reach the control target (Generally will be formulated as the objective function in the MPC optimization problem). For MPC for power electronics, the switching signals are normally chosen as the decision variables of the optimization problem. Conventionally, the gate signals are normally generated by pulse width modulation (PWM). The PWM gate signal generator compares the reference signal

with a high frequency carrier waveform. The frequency of the resulting switching signal will be close to the carrier waveform, which is not necessary at all the time. Furthermore, in real applications, the power electronics gates are not ideal gate that has zero resistance for turning-on mode, infinite resistance for turning-off mode, and zero reacting time for switching the gate status. So the higher switching frequency will results higher switching loss, and the power loss is normally turned into heat generation, which is unwanted for most of the applications.

The concept of MPC is to predict the switching actions for the next N horizons and apply the switching action of the current step to the switch gates. With time evolving, the microprocessor calibrates the actions for the future horizons and sends the control signals related to the current time step to the switch gates. Therefore the MPC for power electronics should be able to provide a switching signal with a variable switching frequency. The gates should only switch its status when necessary. Another difference between MPC and PWM is the input of the modulator. For conventional PWM method, the input of the modulator is the voltage reference. The current tracking is achieved by a feedback control loop with vector control. While the MPC control works as a current regulator in particular. With MPC, the switching signal for the power electronics is directly generated by the MPC controller with current reference and measurement from the system [26] [27]. The major difficulty to solve the MPC for power electronic devices is that the system dynamic model of power electronic devices could be nonlinear with binary terms, such as the Modular Multi-level Converter (MMC). The dynamic model of the MMC includes a lot integer variables for each module's switch status, and bilinear terms about the output current. It means the MPC problem of MMC is a nonlinear MIP problem which generally hard to obtain the solution and high computing burden. Thus, it is worth to design a more efficient method to solve the MPC problem.

1.5 Security Constrained Optimal Power Flow (SCOPF) Problem

Security constrained OPF (SCOPF) is an extension of OPF. Its purpose is to find an operation point to optimize an objective function at base case, while satisfying all pre-contingency (base case) and post-contingency constraints. There are two major approaches to formulate the SCOPF problem, which are preventive approach and corrective approach. In the preventive approach,

any undesirable operation conditions will be prevented from the beginning if the contingency happens. Thus, the re-dispatches are not allowed for the control variables, except those automatically response to contingencies, such as the primary and secondary frequency response [28]. In the corrective approach, the constraints violations which are caused by the contingency can be removed within a certain time limits by the predefined corrective actions, such as switching the transmission lines or generators [29]. It means the SCOPF with corrective approach is easier to get more optimal solution, but has to consider more variables and constraints which will make the system more difficult to be solved. This dissertation focuses on the preventive-SCOPF (PSCOPF), because in the industrial application, preventive-SCOPF (PSCOPF) is dominated [30].

The major challenge to solve PSCOPF is the large size of the problem. Even we only consider the "N-1" criterion, the computation cost of the SCOPF with all contingencies considered could be too high. To address this issue, the decomposition technique is implemented to reduce the decomposition cost. There are two major techniques are implemented in the SCOPF problem, Benders' decomposition and alternating directions methods of multiplier [31]. In Benders' decomposition, the SCOPF problem is decomposed into one master problem and multiple subproblems, each associated to a contingency. The problem will be solved based on the iteration, and feasibility cuts generated by subproblems will be added to the master problem in each iteration. The problem will be considered as solved until the there is no feasibility cut need to be added. In ADMM, the SCOPF will be decomposed into a number of smaller subproblems related to each contingency. All subproblems can be solved parallel. ADMM also need to be solved based on the iteration. But different with Benders' decomposition, the feasible regions of subproblems have not been changed during iterations.

Another issue in PSCOPF is about how to formulate the online generator responses for the generator outage contingencies. A generator's post-contingency response is illustrated in Fig 1.2, where $P_{gi}^{\max}$ and $P_{gi}^{\min}$ are the upper and lower limits of the ith generator's active power output respectively; $P_{gi}^{(0)}$ is the power in pre-contingency state; $P_{gi}^{(k)}$ is the power in post-contingency state; $\Delta^{(k)}$ is the active power imbalance in the system right after the contingency before AGC; $\alpha_i^{(k)}$ is the participation factor corresponding to the slope. For each generator, the participation factor is

the ratio of the output power response of this generator in the total power deviation. Three feasible regions are denoted in the Fig. 1.2.

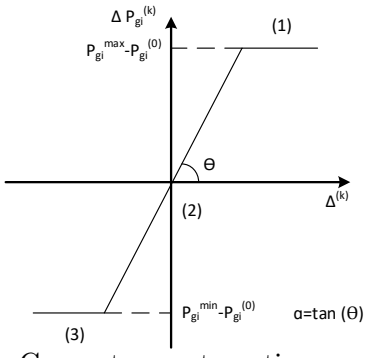


Figure 1.2: Generator post-contingency response.

Considering the generator response is PSCOPF is mathematically challenging because the response has to be modeled as an nonlinear, non-convex constraints. In conventional formulation, the generator outages are ignored in the PSCOPF [32] [33] [34], or the power output of generators are considered as freezing when the generator outage contingencies happen. However, in practical situations, generator outage is common and the remaining online generators will try to compensate the power loss instantaneously because it is governed by the automatic generator control (AGC) setting.

Modeling the non-convex generator response characteristics has been carried out in the literature. In [35] and [36], the authors assume that the generator responses in post-contingency always follow their predefined participation factors, i.e. only region (2) is considered. This formulation could simplify the problem as the generator responses are defined by a set of linear constraints. However, due to the inaccurate representation of the feasible region, the solutions are not correct. Alternatively, mixed integer linear programming (MILP) formulation has been designed in [37] and [38]. The generator response constraint is formulated as a set of MILP constraints based on big-M technique. The major disadvantage of this method is the well-known disadvantage related to big-M formulation , i.e., the difficulty of finding a suitable value M.

1.6 Research Objectives

The major research purpose of this dissertation is to study the application for power system optimization methods. In particular, the dissertation objectives include four major parts:

- Find a more efficient convex relaxation methods to solve the alternative current optimal power flow problem
- Investigate new algorithms to improve the exactness of the convex relaxation formation for the ACOPF problem.
- Build the MPC problem model for MMC, and implement Benders' decomposition method to solve the problem.
- Develop a new method to solve security constrained optimal power flow problem while considering the generator response constraints.

The first two objectives plan to enhance the performance of convex relaxation and improve its exactness, objective 3 and 4 aim to build appropriate problem models for Benders' decomposition and improve the computation efficiency for the problems.

1.7 Outline of the Dissertation

The dissertation is organized as follow:

- Chapter 1 Introduces the importance of power system optimization methods, and presents a brief literature reviews for the problems tackled in this dissertation.
- The proposed convex relaxation method to solve ACOPF will be presented in the Chapter 2. This chapter first introduces the classic formulation of the ACOPF, and the SOCP and SDP relaxation formulation of ACOPF problem in section 2.2 and 2.3 respectively. In section 2.4 presents the maximal clique- and cycle-based convex relaxation formulations. Finally, numerical results are presented in section 2.5.
- Chapter 3 investigate two methods to improve the exactness of convex relaxations. The first method is based on the convex iteration. The second method is based on the Rank-1 PSD

Matrix-Based Nonlinear Programming. Section 3.3 presents the principle of the convex iteration and its implementation on the convex relaxed ACOPF problem. The formulation of the ACOPF which is based on rank-1 PSD Matrix-Based Nonlinear Programming is introduced in section 3.4. The performance of these two methods on improving the exactness of the convex relaxation is discussed in the section 3.5.

- Chapter 4 introduce the Benders' decomposition based MPC for MMC. In section 4.2, the dynamic model for MMC is derived. Section 4.3 gives the details about the formulation of the MPC as a nonlinear MIP problem. Section 4.4 gives the Benders' decomposition algorithm. The comparison of the performance for nonlinear MIP and Benders' decomposition is compared through case study in the section 4.5.

- Chapter 5 presents the proposed method to solve SCOPF with generator response constraints. In section 5.2, we introduce SCOPF and the MILP formulation of the generator response constraints. The proposed bilinear formulation and tow Benders' decomposition strategies are described in section 5.3. And case study is presented in the section 5.4.

- Chapter 6 concludes the dissertation and proposes future works by extending the research to more complex and practical power system. And the further study to implement distributed computation algorithm to solve SCOPF problem.

Chapter 2: A Sparse Convex ACOPF Solver Based on 3-node Cycles

2.1 Introduction

[1]Alternating current optimal power flow (ACOPF) is a classic optimization problem in power systems [40]. The objective is to minimize generation cost or power loss. Constraints are related to power grid physical characteristics (e.g., power flow equations), component limits (e.g., generator capacity limits, transmission line limits) as well as network operation limits (e.g., voltage limits). Depending on the practical requirements, some extra constraints may be included, such as security constraint [29], or stability constraint [41, 42].

As the power flow equations are nonlinear, ACOPF is non-convex. Traditionally, nonlinear optimization solving methods, e.g., Newton's method and interior point method , have been applied to solve the problem [4]. These methods essentially find a local optimal solution in the feasible region that satisfies the first-order optimality condition [5]. Examples presented in [5] indicate that local optima could occur due to disconnected feasible regions, loop flow, an excess of real or reactive power, or large difference in voltage angles across lines.

Global optimum means guaranteed least cost. In recent years, applying convex relaxation techniques to solve ACOPF problem and find global minimum has been carried out and a tutorial can be found in [6, 43]. Relaxation problems find the lower bound of the solution to ACOPF. The two major relaxation techniques are SDP relaxation, and SOCP relaxation. SDP relaxation was first applied to solve ACOPF in [13]. SOCP relaxation was proposed in [14] for radial networks. In radial networks, SOCP relaxation and SDP relaxation are equivalent [6].

SDP relaxation of ACOPF has shown to be a very strong convex relaxation to the original non-convex formulation [16, 44]. Nevertheless, the disadvantage of SDP relaxation is its expensive computational cost.

[1]This chapter was published in Electric Power Systems Research [39]. Permission is included in the Appendix A.

For that reason, sparse technique has been exploited for SDP relaxation [19, 45, 46]. The main theorem is the PSD matrix completion theorem [47], which states that if every submatrix related to every maximal clique in a chordal graph is PSD, then the partial symmetric matrix X_{ch} corresponding to the chordal graph can be completed as a full PSD matrix $X \succeq 0$. The graph related to the topology of a power grid is usually not a chordal graph. To obtain a chordal graph, Cholesky factorization has been used to find chordal extension [45]. Detailed implementation procedure of Cholesky factorization and sparse SDP relaxation can be found in [46]. Instead of finding maximal cliques and further a clique tree through Cholesky factorization, tree width decomposition can also be used to find a clique tree [19]. This method has been implemented in a software package for SDP relaxation of OPF [48].

The aforementioned researches focus on sparse SDP relaxation. On the other hand, there is a category of research focusing on strengthening SOCP relaxation [49, 50]. Compared to SDP relaxation, SOCP relaxation is computationally more efficient. Nevertheless, the feasible region of SOCP relaxation is less tight. Strengthening SOCP relaxation has been studied in [49, 50] by implementing cutting plane algorithms, i.e., iteratively adding valid inequalities, including SDP based ones. The principle of the methods in [49, 50] is based on the fact that for a PSD matrix, its submatrices corresponding to cycles in a cycle basis are PSD. If a submatrix of the solution of the SOCP is not PSD, a valid inequality can be constructed to reduce the search region. The constraint can be constructed using duality concept in [49] and shortest Euclidean distance technique in [50].

In this research, we explore an alternative computationally friendly method that can strengthen SOCP relaxation. Instead of iteratively solving and strengthening the SOCP relaxation by cutting plane algorithms, we propose to directly add maximal clique-based and cycle-based SDP feasibility constraints in the SOCP relaxation. Those added constraints enforce the submatrices related to maximal cliques and cycles to be PSD.

Further, we conduct chordal relaxation for chordless cycles. A chordless cycle of size n can be decomposed into 3-node cycles or cliques by adding $(n-3)$ virtual edges. Adding virtual lines has also been adopted by other researchers. For example, [51] proposed to add virtual lines between

the reference bus and all its non-adjacent buses. By enforcing all submatrices related to the virtual lines PSD, the resulting convex relaxation in [51] is stronger than the original SOCP.

Compared to [51], our method of virtual line addition based on chrodless cycle 3-node decomposition results in less virtual lines and thus is more efficient. Overall, this method results in a stronger convex relaxation compare to SOCP relaxation. All 2×2 principal submatrices of the full matrix are guaranteed to be PSD. Further, all maximal cliques with size greater than 2 in the original power grid graph, and the 3-node cycles constructed from chordless cycles are PSD. The computing efficiency has been compared with sparse SDP methods [46] [19] and is found to be higher. Though the graph after chordal extension is not guaranteed to be a chordal graph, the resulting formulations in many cases are as strong as SDP relaxation.

The major contributions in this chapter is twofold. First, we answer a question naturally arise from the research results from [49] and [50]: Will a PSD solution be found if its SOCP solution's submatrices related to cycles in a cycle basis are PSD? We demonstrate that chordal relaxation for every cycle in a cycle basis cannot result in a chordal graph. Hence, there is no guarantee that the strengthened SOCP in [49] and [50] can lead to SDP solution eventually. Second, we propose a stronger convex relaxation compared to SOCP by enforcing minimal cycles of a cycle basis PSD. This enforcement can be further replaced by 3-node cycle PSD enforcement. The proposed solver is implemented in CVX platform and shows higher computing efficiency compared with sparse SDP methods [46] [19].

2.2 ACOPF Problem

First we describe the original formulation of ACOPF. Considering a power network, we denote the buses as $i \in \mathcal{B}$, the transmission line as $(i, j) \in \mathcal{L}$ and the generators as $i \in \mathcal{G}$. The admittance matrix is defined as Y where $Y = G + jB$, G and B are the conductance matrix and the susceptance

matrix, respectively. The classic ACOPF problem is formulated as follows.

$$\min_{V,\theta,P_g,Q_g} \quad \sum_{k\in\mathcal{G}} C_{2k}P_{gk}^2 + C_{1k}P_{gk} + C_{0k} \tag{2.1a}$$

$$P_i^g - P_i^d = \sum_{j\in Adj_i}^{n} V_iV_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}), i \in \mathcal{N} \tag{2.1b}$$

$$Q_i^g - Q_i^d = \sum_{j\in Adj_i} V_iV_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}), i \in \mathcal{N} \tag{2.1c}$$

$$|S_{ij}(V,\theta)| \leq S_{ij}^{\max}, \quad (i,j) \in \mathcal{L} \tag{2.1d}$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}, \quad i \in \mathcal{G} \tag{2.1e}$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, \quad i \in \mathcal{G} \tag{2.1f}$$

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \quad i \in \mathcal{N} \tag{2.1g}$$

where C_{2k}, C_{1k} and C_{0k} are the coefficients of the quadratic cost function for the generator k, P_i^g, Q_i^g are the total generated active and reactive powers from the generators connected at Bus i, P_i^d, Q_i^d are total demanded active and reactive powers at Bus i, Adj_i is the set of the buses that have direct connection with Bus i, $V \in \mathbb{R}^{|\mathcal{N}|}$ and $\theta \in \mathbb{R}^{|\mathcal{N}|}$ are the voltage magnitude vector and angle vector, respectively, S_{ij} is the complex power flow on the transmission line from Bus i to Bus j. $|.|$ notates the cardinality of a set. The decision variables are $\{P_g, Q_g, V, \theta\}$.

2.3 SOCP and SDP Relaxations of ACOPF

The ACOPF formulation is a non-convex optimization problem. This can be shown by the power injection equality constraints(2.1b) and (2.1c). Note that the equality constraints of power injections are nonlinear in terms of V and θ. This yields the ACOPF problem non-convex. Relaxations have been developed in the literature to have a convex feasible region. These methods deal with new sets of decision variables to replace V and θ.

In SOCP relaxation [14], a new set of variables c_{ij} and s_{ij} is used to replace the voltage phasors $V_i \angle \theta_i, i \in \mathcal{B}$.

$$
\begin{aligned}
c_{ii} &= V_i^2, \ i \in \mathcal{B} \\
c_{ij} &= V_i V_j \cos(\theta_i - \theta_j), \ (i,j) \in \mathcal{L} \\
s_{ij} &= -V_i V_j \sin(\theta_i - \theta_j), \ (i,j) \in \mathcal{L}
\end{aligned}
$$

where $c_{ij} = c_{ji}$ and $s_{ij} = -s_{ji}$.

It is easy to find the following relationship:

$$
c_{ij}^2 + s_{ij}^2 = V_i^2 V_j^2 = c_{ii} c_{jj}. \tag{2.2}
$$

There will be $|\mathcal{L}|$ number c_{ij} and s_{ij}. If there is no direct connection between Bus i and Bus j, the power injection equations will not contain c_{ij} nor s_{ij}. The decision variables V and θ are replaced by $c_{ii}, i \in \mathcal{B}$, and $c_{ij}, s_{ij}, (i,j) \in \mathcal{L}$. The dimension of the new set of the variables is $2|\mathcal{G}|+|\mathcal{B}|+2|\mathcal{L}|$.

With the new set of variables, power injection equations (2.1b) and (2.1c) are now linear. The line flow limit constraints become second-order cone constraints. The only constraint that makes the program non-convex is (2.2). This constraint can be relaxed as a second-order cone:

$$
c_{ij}^2 + s_{ij}^2 \leq c_{ii} c_{jj}, \quad (i,j) \in \mathcal{L} \tag{2.3}
$$

This relaxation was first proposed in [14] for ACOPF and named as SOCP relaxation.

In SDP relaxation proposed by [13], the decision variables related to voltage phasors are replaced by a matrix $X = \overline{V V}^H$, where $X_{ij} = \overline{V}_i \overline{V}_j^* = c_{ij} - j s_{ij}$, superscript H denotes Hermitian transpose for a vector and superscript $*$ means complex conjugate for a scalar.

(2.3) can be rewritten as

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = X_{ii}X_{jj} - X_{ij}X_{ji} \geq 0. \tag{2.4}$$

Based on the definition $X = \overline{V}V^H$, it is obvious that X is PSD and rank-1.

$$X = X^H, \quad X \succeq 0, \text{and } \operatorname{rank}(X) = 1 \tag{2.5}$$

where $X \succeq 0$ means that this matrix is PSD.

The power injection constraints are linear with the elements of X. With the rank-1 constraint relaxed, the problem is a convex problem: SDP relaxation of ACOPF. For tree networks, SOCP relaxation and SDP relaxation are equivalent [44]. For meshed network, the SOCP constraint (2.3) or (2.4) enforces only 2×2 principal submatrices related to lines PSD. For cliques with sizes greater than 2 and cycles, SOCP relaxation does not guarantee the related submatrices PSD.

2.4 Prposed Sparse Convex Relaxation Formulation

Instead of dealing with a full matrix X for the entire grid, for each maximal clique and each cycle in the cycle basis of the network, we impose the PSD constraint for the corresponding submatrix $\widetilde{X}^{(i)}$. This is equivalent to first conduct chordal extension to make a chordless cycle of size n a clique and then enforce the related $n \times n$ submatrix PSD. On the other hand, there is a more efficient way of chordal extension for a chordless cycle: A cycle can be decomposed into 3-node cycles. This approach can save computing cost due to the reduction of the size of the PSD matrices (all 3×3). We further examine if such chordal relaxation can lead to a chordal graph for the entire power grid. If so, we have a sparse SDP relaxation. If not, we have a stronger convex relaxation compared to SOCP.

In this section, we first review a few graph theory techniques that will be used to identify maximal cliques and chordless cycles. We then examine the proposed chordal extension for various examples and check if they can lead to a chordal graph. Finally, we give the proposed sparse convex relaxation formulation.

2.4.1 Maximal Cliques Identification

Given a graph's boolean adjacency matrix, all maximal cliques can be identified using Bron-Kerbosch algorithm [52]. In this project, a MATLAB toolbox [53] based on Bron-Kerbosch algorithm has been used to identify maximal cliques. Table 2.1 presents the size of the largest maximal cliques in test instances. We may observe that all grids have largest maximal cliques with size 3 or less, except IEEE 118-bus system. This system has a maximal clique of size 4.

After identifying the maximal cliques in a power grid, the next step is to identify chordless cycles.

Table 2.1: Size of the largest maximal cliques

Test case	size	Test case	size
nesta_case3_lmbd	3	nesta_case4_gs	2
nesta_case5_pjm	3	nesta_case14_ieee	3
nesta_case30_ieee	3	nesta_case57_ieee	3
nesta_case118_ieee	4	nesta_case300_ieee	3
nesta_case1354_pegase	3	nesta_case2383wp	3
nesta_case2736sp_mp	3	nesta_case2737sop_mp	3
nesta_case2746wop_mp	3	nesta_case2746wp_mp	3
nesta_case3012wp_mp	3	nesta_case3120sp_mp	3

2.4.2 Minimal Cycles in a Cycle Basis

Cycle basis identification algorithm in [54] is used to identify the cycle basis. A related MATLAB toolbox is also available [55]. The procedure of cycle basis identification is to first build a spanning tree. The edges that are not in the spanning tree are identified as the back edges. The number of back edges is the number of the cycles in a cycle basis. The back edges are added back to the spanning tree one by one. If a back edge is added, one cycle is identified. The resulting cycles are not necessarily minimal cycles. In this project, we aim to find minimal cycles. The justification of minimal cycles is given by the following example.

Fig. 2.1 presents an example graph to illustrate the chordal graph construction and why minimal chordless cycles are desired. Fig. 2.1(a) presents the original topology of a graph. The definition of a chordal graph is that all cycles of four or more vertices have a chord. This original graph is not a chordal graph since there is no chord for cycle $\{2, 3, 4, 5\}$.

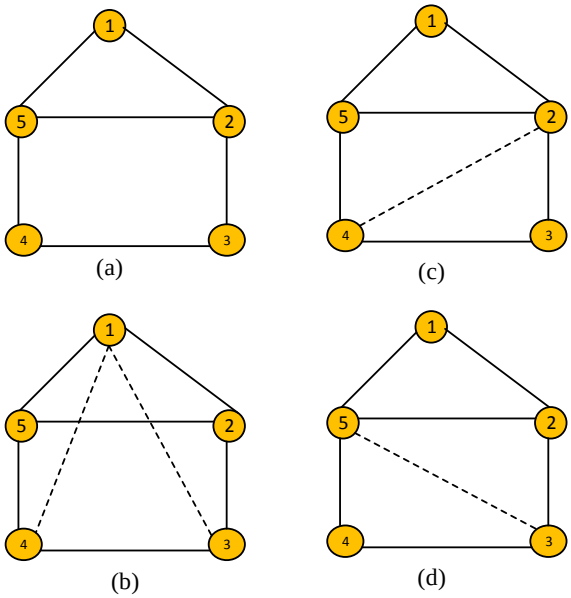


Figure 2.1: Chordal graph construction explanation.

As cycle basis identification algorithm does not guarantee minimal chordless cycles, we may end up with two cycles identified for Fig. 2.1(a): $\{1, 2, 5\}$ and $\{1, 2, 3, 4, 5\}$. Suppose that for the second cycle identified, two lines: $1 - 4$ and $1 - 5$ are added. The resulting graph shown in Fig. 2.1(b) is not a chordal graph since there is no chord in cycle $\{2, 3, 4, 5\}$.

On the other hand, if we are able to identify the two minimal chordless cycles as $\{1, 2, 5\}$ and $\{2, 3, 4, 5\}$, we may add a chord in the 4-node cycle (line $2 - 4$ or line $3 - 5$). The resulting graphs shown in Fig. 2.1(c)(d) are two chordal graphs.

To find minimal chordless cycle in a cycle basis, we use shortest path algorithm. For every back edge, first, the entire graph will have this back edge removed. The two nodes of the back edge are defined as the start node and the destination node. Shortest path in the modified graph from the start node to the destination node can be found using MATLAB 2017's graph toolbox.

2.4.3 Chordal Extension

A further graph decomposition strategy is employed to have virtual lines added and have any minimal chordless cycles with size greater than 3 to be decomposed into cycles with 3 nodes. The number of virtual lines added is $(n - 3)$ where n is the size.

The graph after chordal extension is not necessarily a chordal graph. Though these cycles can be extended into chordal graphs, the entire grid may still have other cycles with size greater than 3. If a graph is chordal, then there exists a permutation to make the Cholesky factorization with zero filling. On the other hand, if there exists Cholesky factorization with non-zero filling, then the graph is not a chordal graph. In this project, we adopt MATPOWER's SDP toolbox [56] function "maxcardsearch" written by Dan Molzahn to conduct the check.

IEEE 14-bus system and IEEE 30-bus system are used as two examples in Fig. 2.2 to demonstrate minimal cycles and 3-node decomposition by adding virtual lines (dotted lines). If the resulting graph after 3-node decomposition is not a chordal graph, Cholesky factorization is then conducted. The additional virtual lines will be added as solid magenta lines. We can see that the 14-bus system after 3-node decomposition is a chordal graph while the 30-bus system after 3-node decomposition is not a chordal graph. Additional lines should be added to achieve a chordal graph.

Two cycles with size greater than 4 are identified for the 14-bus system. They are $\{4, 5, 6, 13, 14, 9\}$ and $\{4, 5, 6, 11, 10, 9\}$. Node 6 is used as the starting node to add virtual lines for both cycles. Total there are 4 virtual lines added to decompose the two cycles into 3-node cycles. The resulting graph is a chordal graph.

Four cycles with size greater than 4 are identified for the 30-bus system. They are $\{16, 12, 4, 6, 10, 17\}$, $\{25, 27, 28, 6, 10, 22, 24\}$, $\{18, 15, 12, 4, 6, 10, 20, 19\}$, and $\{23, 15, 12, 4, 6, 10, 22, 24\}$. For the first cycle, 3 virtual lines are added starting from node 16: $16 - 4$, $16 - 6$, and $16 - 10$. Similarly, virtual lines are also added. The resulting graph, however, is not a choral graph. Cholesky factorization is then conducted and 5 virtual lines are found: $25 - 23$, $6 - 12$, $16 - 23$, $16 - 18$, $23 - 18$.

We have also conducted chordal extension to make every minimal cycle a clique. The resulting graphs for systems with size more than 57 are found as not chordal graphs.

This study answers a question naturally arise from the research on cycle-based SDP feasibility enforcement presented in [49] and [50]: Will a PSD solution be found if its SOCP solution's submatrices related to cycles in a cycle basis are PSD? We demonstrate that chordal relaxation for

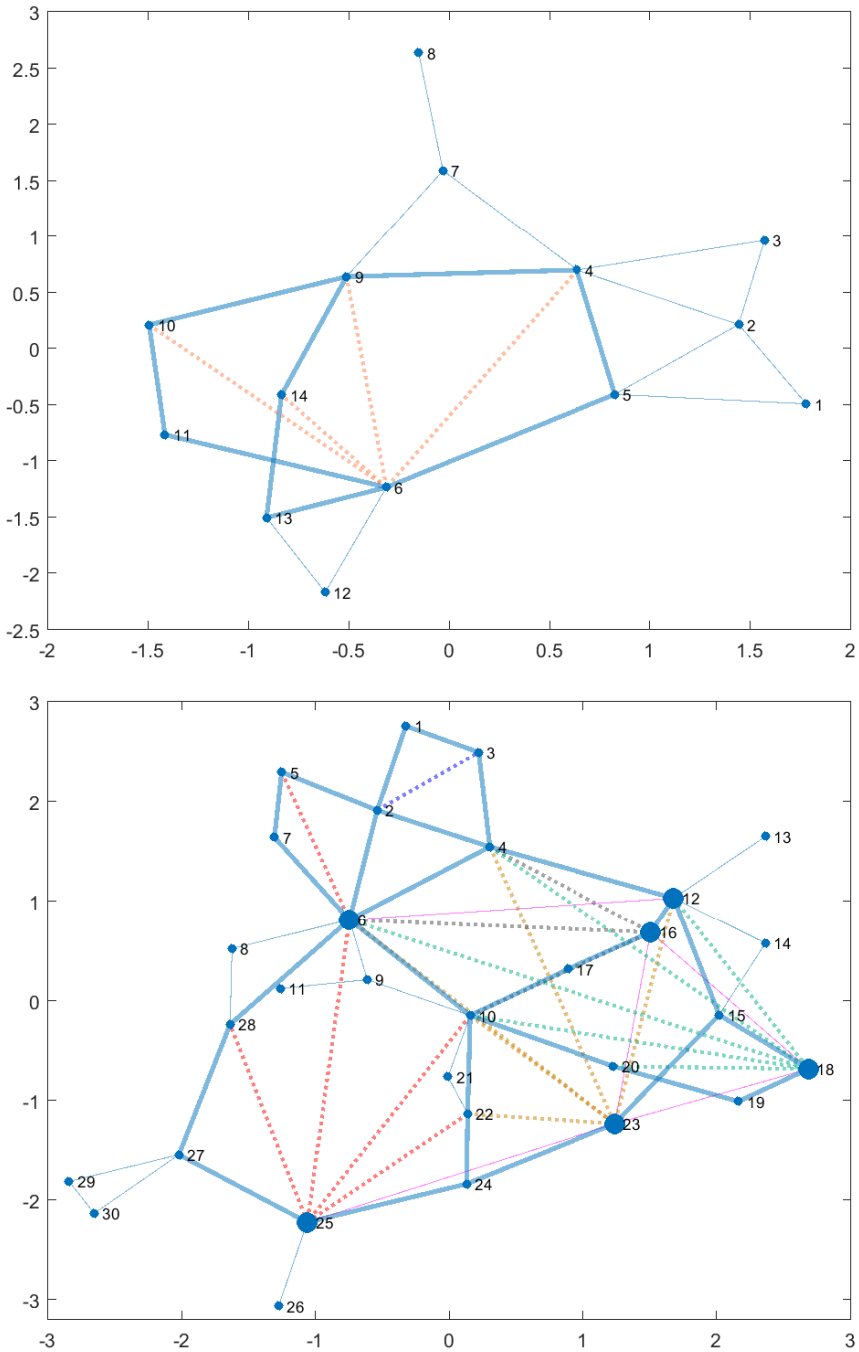


Figure 2.2: Topologies of IEEE 14-bus case and IEEE 30-bus. Blue solid lines represent the lines of the power grid. Highlighted blue lines notate the edges related to cycles of more than 3 nodes. Dotted lines are the virtual lines added to decompose a cycle into 3-node cycles. The solid magenta lines in the 30-bus case notates additional lines added to make the graph chordal.

each cycle of a cycle basis cannot result in a chordal graph. Hence, there is no guarantee that the strengthened SOCP in [49] and [50] can lead to SDP solution eventually.

2.4.4 Proposed Sparse Convex Relaxation Formulation

With no guarantee of a chordal graph, the 3-node decomposition leads to a sparse convex relaxation. The decision variables of the proposed convex relaxation include c_{ii} ($i \in \mathcal{B}$) and c_{ij}, s_{ij} ($(i,j) \in \mathcal{L} \cup \mathcal{V}$). $\mathcal{V}$ notates that the set of virtual lines for 3-node cycles decomposition. Note that compared to SOCP formulation whose decision variables include c_{ij} and s_{ij} for every line, the proposed convex relaxation has additional decision variables related to virtual lines.

Sparse matrix technique is employed in the proposed convex relaxation formulation. A sparse matrix X is defined to have its diagonal elements and elements related to lines and virtual lines non zero. The rest elements are all zeros.

$$X_{ii} = c_{ii}, \qquad i \in \mathcal{B} \tag{2.6a}$$

$$X_{ij} = c_{ij} - j s_{ij}, \qquad (i,j) \in \mathcal{L} \cup \mathcal{V} \tag{2.6b}$$

$$X_{ji} = c_{ij} + j s_{ij}, \qquad (i,j) \in \mathcal{L} \cup \mathcal{V} \tag{2.6c}$$

The sparse convex relaxation enforces all submatrices related to maximal cliques PSD. The maximal cliques include the maximal cliques with size greater than 2 from the original graph, 3-node cycles resulting from decomposition, and rest lines. The formulation is presented in (2.7).

In the formulation (2.7), $\mathcal{S}_{MC}$ notates the set of maximal cliques and $\widetilde{X}^i$ notates the submatrix of X related to i^{th} maximal clique. $g_{ij} = \text{real}(y_{ij})$ and $b_{ij} = \text{imag}(y_{ij})$ and y_{ij} is a branch (between Bus i and Bus j)'s admittance. (2.7b) and (2.7c) represent the net power injection constraints at each bus. (2.7f) is the line flow limit constraint. (2.7g) is the voltage limit constraint. (2.7h) are the generator power limit constraints. (2.7i) enforces PSD for submatrices related to maximal cliques.

Formulation (2.7) is a general SOCP/SDP ACOPF solver employing sparse matrix technique. If chordal extension of a power grid can result in a chordal graph, the above solver is a SDP solver. On the other hand, if the graph is not a chordal graph, the above solver is a stronger convex

relaxation solver than SOCP. For comparison, we have employed Cholesky factorization method to find a chordal graph (Solver B in section 2.5).

$$\min \quad f(P_g) \tag{2.7a}$$

$$s.t. \quad P_i^g - P_i^d = \sum_{j\in\delta_i}(G_{ij}c_{ij} - B_{ij}s_{ij}), \quad i \in \mathcal{B} \tag{2.7b}$$

$$Q_i^g - Q_i^d = \sum_{j\in\delta_i}(-G_{ij}s_{ij} - B_{ij}c_{ij}), \quad i \in \mathcal{B} \tag{2.7c}$$

$$P_{ij} = g_{ij}(c_{ii} - c_{ij}) - b_{ij}s_{ij}, \quad (i,j) \in \mathcal{L} \tag{2.7d}$$

$$Q_{ij} = -b_{ij}(c_{ii} - c_{ij}) - g_{ij}s_{ij}, \quad (i,j) \in \mathcal{L} \tag{2.7e}$$

$$\sqrt{P_{ij}^2 + Q_{ij}^2} - S^{\max} \leq 0, \quad (i,j) \in \mathcal{L} \tag{2.7f}$$

$$(V_i^{\min})^2 \leq c_{ii} \leq (V_i^{\max})^2, \quad i \in \mathcal{B} \tag{2.7g}$$

$$P_g^{\min} \leq P_g \leq P_g^{\max}, \quad Q_g^{\min} \leq Q_g \leq Q_g^{\max} \tag{2.7h}$$

Constraints (2.6)

For all cliques MC,

$$\widetilde{X}^{(i)} \succeq 0, \quad i \in \mathcal{S}_{MC} \tag{2.7i}$$

2.5 Case Study

Instances from the NICTA test archive [57] are tested using the proposed formulation. Additional instances with large gaps (case9mod, case39mod1, case300mod) are from [5]. We also implemented the method in [45] and [46] and developed a sparse SDP solver based on a chordal graph using Cholesky factorization. Cholesky factorization of a Hermitian semi-definite matrix A is defined as follows. $P_\sigma A P_\sigma^T = LL^T$, where P_σ is a permutation of the elements in A; L is a lower triangular matrix which is called Cholesky factor of A. The sparsity pattern for the chordal extension of A is the same with $L + L^T$ [45]. Moreover, to obtain minimal number of virtual lines, P_σ will permute the indexes of A based on the minimal degree ordering. Using Cholesy factoriza-

tion, virtual lines of a power grid graph are found and added to achieve a chordal graph. Maximal cliques of the chordal graph are then found and the submatrices related to the maximal cliques are enforced to be PSD.

The cases were first solved by MATPOWER [56] to obtain feasible solutions as upper bounds. In addition, the cases were solved by the SDP solver developed by Lavaei's group [48] (Solver A), the sparse SDP solver based on Cholesky factorization (Solver B), and the proposed solver (Solver C).

We compare the computing time and solutions of the three solvers to demonstrate that 1) the gaps from the proposed solver is as tight as those from other sparse SDP solvers; 2) the computing efficiency is higher compared with the two sparse SDP solvers. The number of virtual lines required for the three solvers, sizes of maximal cliques, and ranks of submatrices generated are all compared. To show the proposed relaxation solver is stronger than the SOCP solver, we compared the optimality gap of the proposed relaxation with one strengthen SOCP solver [50]. Finally, we select a few instances with nonzero gaps to demonstrate that convex iteration based on 3-node cycles can give rank-1 PSD solutions in those instances.

In all tests, the gap is defined as: $Gap = \frac{UB-LB}{UB} \times 100\%$, where UB is the upper bound which is calculated through MATPOWER; LB is the lower bound of the objective value. In Table 2.2 and 2.3, LB is calculated by the relaxation solvers; in Table 3.1, LB is calculated by the convex iteration solver.

2.5.1 Proposed Convex Relaxation

Our numerical experiments are conducted on an Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30 GHZ (2 processors) computer. All solvers are implemented in MATLAB 2017a-based CVX platform [58]. Mosek 7.1.0.12 solver is invoked. To achieve the balance between the stability and accuracy, we adopt the Mosek setting of Solver A(tuned by Lavaei's group). Although this setting may decrease the accuracy of the solution for large size cases, it provides the best stability for the Mosek solver (Mosek with default setting may fail to solve the cases which are larger than 2736 buses). In Table 2.3, we compared the proposed relaxation solver with the strengthen SOCP solver [50]. Because the test cases of reference [50] also comes from the NICTA test archive, we cited the results of [50]

in Table 2.3. The numerical results from two SDP solvers and the proposed solver are listed in Table 2.2.

Table 2.2: Results comparison

Case	UB	Gap(%)			Solver Time			Max_cliqueSize			Max_Rank			N_vline		Decomp_Time	
		A	B	C	A	B	C	A	B	C	A	B	C	B	C	B	C
nesta_case3_lmbd	5812.64	0.41	0.39	0.39	0.58	**0.30**	0.48	3	3	3	2	2	2	0	0	0.02	0.02
nesta_case4_gs	156.43	0.00	0.00	0.00	0.56	0.39	**0.39**	3	3	3	1	1	1	1	1	0.01	0.03
nesta_case5_pjm	17551.89	5.22	5.23	5.22	0.90	**0.37**	0.45	3	3	3	2	2	2	1	1	0.02	0.02
nesta_case14_ieee	244.05	0.00	0.00	0.00	0.59	**0.42**	0.51	3	3	3	1	1	2	4	4	0.01	0.02
nesta_case30_ieee	204.97	0.00	0.00	0.00	0.58	0.42	**0.41**	4	4	3	1	1	1	14	14	0.03	0.03
nesta_case57_ieee	1143.28	0.00	0.00	0.00	0.81	0.95	**0.59**	6	6	3	2	1	1	59	55	0.05	0.06
nesta_case118_ieee	3689.92	0.07	0.07	0.09	1.34	1.78	**1.34**	5	5	4	2	2	3	87	73	0.12	0.33
nesta_case300_ieee	16891.28	0.08	0.08	0.09	6.92	4.93	**3.51**	7	7	3	2	2	3	250	193	0.55	0.55
nesta_case1354_pegase	74064.11	0.56	0.50	1.20	26.42	20.10	**11.53**	13	13	3	6	6	3	1020	698	1.13	3.79
nesta_case2383wp_mp	1870807.81	0.96	1.38	1.02	100.04	86.92	**35.05**	24	25	3	6	6	3	3269	2225	4.23	7.55
nesta_case2736sp_mp	1307961.70	28.01	27.77	27.94	36.53	37.00	**11.54**	24	25	3	6	6	3	3878	2810	5.79	8.26
nesta_case2737sop_mp	777668.88	11.84	11.37	11.37	23.40	25.07	**18.21**	24	24	3	6	6	3	3853	2814	5.91	8.22
nesta_case2746wop_mp	1208281.08	15.42	15.44	15.68	46.92	31.98	**11.15**	24	26	3	6	6	3	4103	2819	6.46	9.33
nesta_case2746wp_mp	1631868.17	28.89	28.92	29.37	47.82	23.07	**9.91**	25	26	3	6	6	3	3973	2800	6.04	8.57
nesta_case3012wp_mp	2600842.77	0.23	0.27	0.80	124.16	115.29	**73.40**	26	28	3	6	6	3	4407	3065	7.51	9.58
nesta_case3120sp_mp	2145965.33	0.33	0.86	0.46	172.45	118.98	**81.20**	29	27	3	6	6	3	4527	3153	7.91	9.74
nesta_case30_fsr_api	372.14	11.08	11.09	11.62	0.53	0.52	**0.11**	4	4	3	2	2	2	14	14	0.02	0.02
nesta_case118_ieee_api	6383.57	5.28	5.29	5.56	1.54	1.23	**0.06**	5	5	4	2	2	3	87	73	0.04	0.34
case9mod.m	4267.07	35.48	35.48	35.48	0.48	**0.45**	0.48	3	3	3	2	2	2	3	3	0.01	0.01
case39mod1	11221.00	3.72	3.72	3.72	0.45	0.53	**0.04**	4	4	3	2	2	2	21	21	0.02	0.06
case300mod	378540.50	0.14	0.14	0.14	5.27	4.37	**0.06**	7	7	3	3	3	3	250	193	0.12	0.66

In Table 2.2, columns A, B, C represent the three solvers; Max_cliqueSize is the size of the largest clique; Max_Rank means the maximum rank of submatrices; Solver Time is the optimizer terminate time of Mosek; N_vline means the numbers of the virtual lines; Decomp_Time is time cost on the cliques decomposition.

According to Table 2.2, from nesta_case3_lmbd to nesta_case300_ieee, Solver C obtains the same results as both or one of the two SDP solvers for small and median size cases. For large size cases, Solver C has a gap slightly larger but a much higher computing efficiency. This is due to the following two facts. 1) The proposed method deals only 3-node cycles while the two SDP solvers deal with cliques with larger sizes. For example, for case nesta_case3120sp_mp, the size of the cliques reach 29 and 27 for Solver A and Solver B. 2) On the other hand, the proposed method adds less virtual lines compared to SDP solver B. For case nesta_case3120sp_mp, the number of virtual lines is 4527 for Solver B while it is 3153 for Solver C.

The proposed relaxation solver is compared with the strengthen SOCP method [50] in Table 2.3. The table shows the optimality gaps for two solvers. We can see performance of the proposed relaxation solver is better than the strengthen SOCP solver.

Table 2.3: Comparison with one strengthened SOCP solver

Case	UB	Gap(%)	
		Proposed Solver	SOCP_SDP [50]
nesta_case3_lmbd	5812.64	0.39	1.27
nesta_case4_gs	156.43	0.00	0.00
nesta_case5_pjm	17551.89	5.22	9.08
nesta_case14_ieee	244.05	0.00	0.00
nesta_case30_ieee	204.97	0.00	0.29
nesta_case57_ieee	1143.28	0.00	0.00
nesta_case118_ieee	3689.92	0.09	1.51
nesta_case300_ieee	16891.28	0.09	0.64

We note that in Table 2.2, there are some cases showing different relaxation gaps between two SDP solvers, and Solver C showing tighter gaps than one of the SDP solvers. According [44], since both sparse SDP solver A and B are based on chordal graphs, their solutions are SDP OPF solutions and should be the same. Moreover, Solver C should have gaps greater than or equal to those from SDP. In our experiments, the reason of the numerical inconsistency is due to the configuration of Mosek. We tested some cases by CVX with SDPT3 in default setting, and listed the results in the Table 2.4. The results show SDP solver A and B, and solver C achieve the same gaps. However, as SDPT3 is much slower than Mosek, we use Mosek for all case studies.

Table 2.4: SDPT3 results

Case	UB	Gap		
		A	B	C
nesta_case300_ieee	16891.28	0.08%	0.08%	0.08 %
nesta_case1354_pegase	74064.11	0.01%	0.01%	0.01 %
nesta_case2736sp_mp	1307961.70	0.00%	0.00%	0.00%
nesta_case2737sop_mp	777668.88	0.00%	0.00%	0.00%

The proposed convex relaxation solver achieves almost the same tightness of SDP solvers with a much higher computing efficiency. The computing time decreases at least 27% with an average of 49%. Our method solves the dilemma mentioned in [44] that decreasing the size of submatrices results in increased virtual lines for sparse SDP. The proposed sparse convex relaxation solver can achieve almost the same tightness of SDP solvers with much higher computing efficiency.

2.6 Conclusion

In this chapter, we proposed a 3-node cycle decomposition based sparse convex relaxation for ACOPF. We have shown that the 3-node cycle decomposition can not guarantee that the resulting graph is a chordal graph. However, the proposed relaxation can achieve the close tightness as SDP OPF solvers. On the other hand, our method has a clearly higher computing efficiency.

Chapter 3: Exactness of the Convex Relaxation

3.1 Introduction

[2]Though it has been studied that SDP relaxation can give global optimum for many IEEE test systems while the solutions are feasible to the original ACOPF problems (termed as "SDP exact") in [16], in some other cases, SDP relaxation leads to inexact solutions for the original problem [5,17,18]. Thus, research efforts have been devoted to achieve SDP exactness, e.g., [19,20].

The exactness conditions for SDP and SOCP relaxations are presented in [6]. Some researches have been conducted to achieve exactness for convex relaxation through exploiting the exactness conditions. In [19, 20], objective functions are modified to include penalty related to the rank-1 constraint. [21] treats an ACOPF problem as an SDP relaxation problem and a non-convex rank-1 feasible region mapping problem. Alternating direction method of multipliers (ADMM) iterative procedure is then applied. In [22,60], the exactness constraints are reformulated as quadratic minor constraints. The minor constraints are approximated as convex constraints in [22]. A strengthened SOCP relaxation of ACOPF is then solved. In [60], the convex-hull descriptions of the minor constraints are examined and valid inequalities are added for SDP relaxation. [23] proposes a convex iteration algorithm to solve a convex problem with a regularization term related to the maximal eigenvalue of the full PSD matrix. With the regularization term achieving zero, the solution achieves global minimum. In [24], the non-convex OPF branch flow equation is decomposed into SOCP constraint and a non-convex constraint related to the difference of two convex functions. The concave term is then approximated by linear functions and updated in each iteration. A sequential convex optimization method is implemented to carry out the iteration. The aforementioned approaches rely on convex relaxation formulations. In many cases, exact solutions can be found after dealing

[2]The majority of this chapter was published in Electric Power Systems Research [39] and International Transactions on Electrical Energy Systems [59]. Permissions are included in the Appendix A.

the exactness condition. However, large gaps are still observed for special cases [20]. In this chapter, we propose two algorithms to achieve the SDP exactness conditions in ACOPF problem. To improve the computation efficiency, both of the algorithms are implemented on the sparse convex ACOPF formulation which is proposed in the Chapter 2.

In the first algorithm, convex iteration is carried out based on 3-node cycles. Convex iteration based SDP OPF has been implemented in [61] [62]. SOCP exactness condition [63] states that the exactness condition is for the submatrices related to two nodes of a line PSD and rank-1, and cycle constraints (sum of the angles across a cycle is zero) satisfied. With every cycle in a cycle basis has been decomposed into 3-node cycles, the cycle constraints can be replaced by the cycle constraints of 3-node cycles. The exactness condition thus requires that every submatrix corresponding to every 3-node cycle is PSD and rank-1. This requirement is further translated to an equivalent requirement: all 3×3 submarices corresponding to 3-node cycles should be PSD and every 2×2 submatrix corresponding to lines and virtual lines should be rank-1. Convex iteration is then implemented to enforce those 2×2 submatrices rank-1. The resulting solution should be a feasible solution.

In the second algorithm, we rely on nonlinear programming formulation with a PSD matrix as the decision variable. We reformulate the rank-1 constraint as a set of quadratic minor constraints. The idea of minor constraints has been mentioned in [22] and [60]. The research in [22, 60] obtains convex constraints to be used to tighten the respective convex relaxation formulations. Different from the aforementioned research, in this work, we will directly deal with all 2×2 minors and come up with a nonlinear programming formulation. Therefore, we aim to use the same decision variables of SOCP or SDP relaxation, but to solve a nonlinear optimization problem with exactness constraints imposed. With the solution from SOCP or SDP relaxation as the starting point, nonlinear programming solvers may find a feasible solution.

Our contribution is three-fold. First, based on the 3-node cycles sparse convex ACOPF formulation, we implement convex iteration to enforce the submatrices related to the 3-node cycles PSD and rank-1. An even more efficient rank-1 enforcement is then derived. With the proposed sparse solver, enforcing all 2×2 submatrices related to lines and virtual lines rank-1 will lead to

a feasible solution. Second, we formulate a nonlinear programming problem of ACOPF based on a new set of decision variables instead of voltage phasors. The new set of decision variables align with those in SDP/SOCP relaxation. In our formulation, rank-1 constraints are replaced by a set of quadratic equality constraints representing all 2×2 minors equal to zeros. The challenge of the formulation is that the number of those minors are very large. For a N-bus power grid, there are a total $C_N^2 C_N^2$ minors. Thus, our third contribution is to employ graph decomposition technique to significantly reduce computational burden. We first decompose a power network into lines and 3-node cycles. Instead of considering all minors, only those minors related to lines and 3-node cycles are considered. As a result, an alternative ACOPF formulation are the final outcome.

3.2 Exactness Condition of SDP and SOCP Relaxation

For a relaxation formulation, if its solution is feasible to the original ACOPF problem, then the solution is exact. The exact conditions of SDP and SOCP have been thoroughly discussed in [6] and are presented as follows.

The exactness condition for SDP is the rank-1 constraint shown in (3.1).

$$X \succeq 0, \quad \text{rank}(X) = 1 \tag{3.1}$$

For SOCP, the exactness conditions are:

$$R_{ij}^2 + I_{ij}^2 = R_{ii} R_{jj}, \text{or} \begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = 0, \text{ for } (i,j) \in \mathcal{L} \tag{3.2a}$$

$$\sum_{(i,j) \in c} \angle X_{ij} = 0, \quad c \in \Psi \tag{3.2b}$$

where Ψ is the set of cycles in the power network.

Note that that the two exactness conditions (3.1) and (3.2) are exchangeable.

3.3 Convex Iteration

3.3.1 Exactness Based on 3-node Cycles

Based on the sparse convex relaxation ACOPF formulation that we proposed in the Chapter 2, the 3-node cycle decomposition makes computing more efficient. In this section, we claim that if

the submatrices related to the 3-node cycles inside each cycle in a cycle basis are PSD and rank-1, then the full matrix is an exact solution.

As the lower limit of bus voltage is larger than zero in general, the constraint (2.7g) can ensure X_{ii} is positive for any $i \in \mathcal{B}$. Thus, the sufficient and necessary condition for a solution from SOCP relaxation being feasible or exact is as follows [44].

$$\begin{vmatrix} X_{ii} & X_{ij} \\ X_{ji} & X_{jj} \end{vmatrix} = 0 \tag{3.3a}$$

$$\sum_{(i,j)\in c} \angle X_{ij} = 0, \quad c \in \mathcal{C} \tag{3.3b}$$

where $\mathcal{C}$ is the set of cycles in the power network.

(3.3a) guarantees that the submatrix related to two nodes i and j related to a line is rank-1. Besides (3.3a), (3.3b) guarantees the submatrix related to a cycle is PSD and rank-1.

For any chordless cycle of size n, we may add $(n-3)$ virtual lines to decompose the cycle into $(n-2)$ 3-node cycles. The cycle constraint (3.3b) can then be replaced by the cycle constraints related to every 3-node cycle.

3.3.2 Mathematical Proof of 3-node Cycles Based Exactness Condition

The mathematical background can be demonstrated by a lemma and a theorem. First, the lemma can be described as : with every cycle in a cycle basis of a graph decomposed into 3-node cycles, if all submatrices corresponding to 3-node cycles are PSD and rank-1, the full matrix related to the entire graph is PSD and rank-1.

This lemma is easy to be proved based on the fundamental knowledge, so we will not discuss it in this subsection. For convenient, we name this lemma as Lemma 1.

And then, we define the theorem: for a 3×3 PSD matrix related to a 3-node cycle, given that all 2×2 submatrices related to lines are PSD and rank-1, then the 3×3 matrix is also rank-1.

For convenient, we name this theorem as Theorem 1, and prove it as follows:

Consider a Hermitian and PSD matrix X related to a 3-node cycle:

$$X = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}.$$

Since the three 2×2 principal submatrices of X related to three lines are PSD and rank-1, their determinants are 0.

$$\begin{vmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{vmatrix} = \begin{vmatrix} X_{22} & X_{23} \\ X_{32} & X_{33} \end{vmatrix} = \begin{vmatrix} X_{11} & X_{13} \\ X_{31} & X_{11} \end{vmatrix} = 0, \text{ or :} \tag{3.4}$$

$$X_{11}X_{22} = |X_{12}|^2,\ X_{22}X_{33} = |X_{23}|^2,\ X_{11}X_{33} = |X_{13}|^2$$

We will examine the determinant of X.

$$\begin{aligned} |X| &= X_{11}X_{22}X_{33} + X_{12}X_{23}X_{31} + X_{13}X_{21}X_{32} \\ &\quad - |X_{13}|^2X_{22} - |X_{23}|^2X_{11} - |X_{12}|^2X_{13} \end{aligned}$$

Replacing $|X_{ij}|^2$ by $X_{ii}X_{jj}$ leads to:

$$\begin{aligned} |X| &= -2X_{11}X_{22}X_{33} + X_{12}X_{23}X_{31} + X_{13}X_{21}X_{32} \\ &= -2X_{11}X_{22}X_{33} + 2|X_{12}||X_{23}||X_{31}|\cos(\theta_{12} + \theta_{23} + \theta_{31}) \end{aligned}$$

where θ_{12}, θ_{23}, θ_{31} represent angles of X_{12}, X_{23}, and X_{31}. Note that $X_{11}X_{22}X_{33} = |X_{12}||X_{23}||X_{31}|$ according to (3.4). Thus,

$$|X| = -2X_{11}X_{22}X_{33}(1 - \cos(\theta_{12} + \theta_{23} + \theta_{31})). \tag{3.5}$$

Since $\cos(\theta_{12} + \theta_{23} + \theta_{31}) \leq 1$, $|X| \leq 0$. On the other hand, X is PSD, hence $|X| \geq 0$. Therefore, $|X| = 0$. This means that the rank of X is less than 3.

The sum of angles is found to be 0 since $|X|= 0$ enforces the following constraint.

$$\cos(\theta_{12} + \theta_{23} + \theta_{31}) = 1 \Rightarrow \quad \theta_{12} + \theta_{23} + \theta_{31} = 0$$

This means that cycle $\{(1,2),(2,3),(3,1)\}$ satisfies the SOCP exactness condition (3.3b). Therefore, based on the sparse convex relaxation formulation in (2.7), to have an exact solution, we only need to enforce 2×2 submatrices corresponding to all lines rank-1.

We will implement this requirement in convex iteration.

3.3.3 Principle of Convex Iteration

Convex iteration has been applied to SDP OPF to achieve exact solutions in [61, 62]. We will briefly review convex iteration principle in this section.

For a $n \times n$ Hermitian PSD matrix, its trace equals the sum of all its eigenvalues.

$$Tr(X) = \sum_{i=1}^{n} \lambda_i, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0 \tag{3.6}$$

where λ_i are the eigenvalues of X; $Tr(\cdot)$ is the "Trace" calculation. If X is rank-1, then all eigenvalues except λ_1 is zero. Thus:

$$Tr(X) - \lambda_1 = 0. \tag{3.7}$$

The maximum eigenvalue λ_1 can be obtained through the following equation [64]:

$$\lambda_1 = \quad Tr(X u_1 u_1^H) \tag{3.8}$$

where u_1 is the normalized eigenvector correspond to λ_1. Thus, combining (3.7) and (3.8) leads to:

$$Tr(X(I - u_1 u_1^H)) = 0 \tag{3.9}$$

Define $W \triangleq I - u_1 u_1^H$. If $Tr(XW) = 0$, then X is rank-1. Thus by adding $Tr(XW)$ as a penalty term on the objective function of the SOCP formulation, we may enforce X to be rank-1.

Note that W is also a variable. This makes the problem a bilinear problem. To solve this bilinear problem, iterative approach can be implemented. Denote the problem including rank-1 penalty term as $F(X, W)$ whose objective function includes an additional term $\omega Tr(XW)$ (ω is the penalty factor). We may fix W to solve a convex problem and find X. Then for the given X, we may find W. Below is the iterative procedure:

$$\text{problem 1:} \quad F(X, W^*) \rightleftharpoons \text{problem 2:} \quad F(X^*, W)$$

where W^* is the solution of the problem 2; X^* is the solution of the problem 1. Consider our derivation from (3.6) to (3.9), it implies that for a fixed X^*, W can be found through eigenvalue-based decomposition of X^*.

$$X^* = U \Lambda U^H \tag{3.10}$$

where Λ is a diagonal matrix with its elements eigenvalues, U is a unitary matrix, and its columns are the eigenvectors of X^*, i.e., $U = [u_1, u_2, \ldots, u_n]$. Thus, problem 2 for iteration procedure is equivalent to:

$$W = UU^H - u_1 u_1^H, \quad \Rightarrow W = U(:, 2:n)U(:, 2:n)^H \tag{3.11}$$

3.3.4 Sparse Implementation

Further, we seek sparse matrix-based implementation. The sparse convex relaxation solver does not give the full matrix X. According to Theorem 1, for the 3-node cycle-based convex relaxation, we only need to enforce all 2×2 submatrices related to lines and virtual lines rank-1 to achieve the exactness. Therefore, the rank penalty term $Tr(XW)$ can be replaced by:

$$\sum_{i \in \mathcal{AL}} Tr(\widehat{X}^{(i)} \widehat{W}^{(i)}) \tag{3.12}$$

where $\mathcal{AL}$ is the set of all lines, including original lines and virtue lines, $\widehat{X}^i$ is the i^{th} 2×2 submatrix and $\widehat{W}^i$ can be found based on $\widehat{X}^i$ using (3.11).

During the iterative procedure, some $Tr(\widehat{X}^{(i)} \widehat{W}^{(i)})$ term might not keep monotonic decreasing. This may lead to the increase of the ranks of those submatrices. To keep the submatrices rank-1

once they have reached rank-1 during iteration, we implement the following constraints to problem 1:

$$Tr(\widehat{X}^{(i)}\widehat{W}^{(i)}) \leq \epsilon \quad i \in \mathcal{DL} \tag{3.13}$$

where ϵ is tolerance, $\mathcal{DL}$ is the set for all submatrices that have achieved rank-1. The equivalent form of this constraint has been adopted in [62].

Our experiments show that this constraint is capable to decrease the iterations for large size cases. E.g. for nesta_case1354_pegase and $\omega = 1000$, with constraint (3.13), convex iteration can converge at 3 steps; without constraint (3.13), convex iteration can not converge after 10 steps. The formulation of 3-node based convex iteration is defined as problem 1 and problem 2.

We define the problem 1 as the following equations:

$$\begin{aligned} \min \quad & f(P_g) + \omega \sum_{i \in \mathcal{AL}} Tr(\widehat{X}^{(i)}\widehat{W}^{*(i)}) \\ s.t. \quad & \text{Constraints (2.6)} \quad \text{(2.7b)} \sim \text{(2.7h)} \\ & Tr(\widehat{X}^{(i)}\widehat{W}^{*(i)}) \leq \epsilon \quad i \in \mathcal{DL} \\ & \text{For all cliques } MC \\ & \widetilde{X}^{(i)} \succeq 0 \quad i \in \mathcal{S}_{MC} \end{aligned} \tag{3.14}$$

We define the problem 2 as the following equations:

$$\widehat{W}^{*(i)} = U^{(i)}(:,2)U^{(i)}(:,2)^H \quad i \in \mathcal{AL} \tag{3.15}$$

where $U^{(i)}$ is obtained through the eigenvalue decomposition of $\widehat{X}^{*(i)}$: $\widehat{X}^{*(i)} = U^{(i)}\Lambda^{(i)}U^{(i)H}$.

The initial values of the iteration can be provided by the solution of Formulation (2.7).

3.4 Nonlinear Rank-1 Formulation

3.4.1 2×2 Minor-based Rank-1 Constraints

To implement exactness constraints, we convert the constraints in (3.1) to minor constraints [22]. The reformulation is based on Proposition 3.1 in [22]: a PSD matrix X is rank 1 if and only if all

its 2×2 minors are zeros and the diagonal elements of X are non-negative. Note a $m \times m$ minor of the matrix $X \in \mathbb{C}^{n \times n}$ is defined as the submatrix of X by deleting $n-m$ rows and $n-m$ columns.

For example, suppose that for an Hermitian X, rows except i, l and columns except j, k are eliminated. The resulting minor constraint is as follows

$$\begin{vmatrix} X_{ij} & X_{ik} \\ X_{lj} & X_{lk} \end{vmatrix} = 0$$

$$\Rightarrow \quad X_{ij}X_{lk} - X_{ik}X_{lj} = 0$$

Separating the real and imaginary parts leads to:

$$\Rightarrow \quad \begin{cases} R_{ij}R_{lk} - I_{ij}I_{lk} - R_{lj}R_{ik} + I_{lj}I_{ik} = 0 \\ I_{ij}R_{lk} - R_{ij}I_{lk} - I_{lj}R_{ik} - R_{lj}I_{ik} = 0 \end{cases} \tag{3.16}$$

Thus we may find all 2×2 minors of the SDP relaxation's decision variable X. The challenge is that the number of minors is large. The total number of minor constraints in terms of X is $C_n^2 C_n^2$. Among them, C_n^2 are the number of the principal minors as shown in (3.2a). These minors can be expressed in R and I and there are total C_n^2 constraints in the real domain. The rest minors are non-principal minors and each can be separated into two constraints in the real domain. Considering the Hermitian matrix's feature ($X^T = X^*$), we will have $\frac{C_n^2 C_n^2 - C_n^2}{2}$ constraints to represent the non-principal 2×2 minors in the complex domain, or $C_n^2 C_n^2 - C_n^2$ non-principal minor constraints in the real domain. Note that the principal minor constraints include the exactness constraints for every lines. Therefore, the total number of constraints in real-domain related to all minors is $C_n^2 C_n^2$. For a 10-node system, this number is 2025.

Instead of dealing with the full matrix X, we now examine the SOCP exactness conditions. The first condition (3.2a) is related to each line. For a branch connecting Bus i and Bus j, the exactness condition is to have the principal minor related to i and j be zero. The next condition (3.2b) is the cycle constraint which should be enforced for every cycles.

In the following, we will show the cycle constraint can be replaced by non-principal minor constraints of each embedded 3-node cycle. There are three steps to lead to the conclusion. The

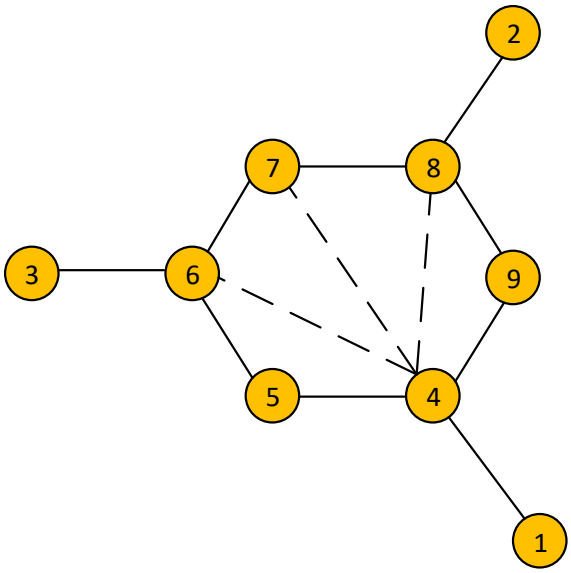


Figure 3.1: One chordless cycle become 3-node cycles with virtual lines.

first step is to show an n-node cycle can be decomposed into $(n-2)$ 3-node cycles, the second step is to show that the exactness conditions in (3.2a) and (3.2b) for a 3-node cycle indeed guarantees a PSD rank-1 matrix. Hence the conditions can be replaced by a set of 2×2 minor constraints. The third step is to show these minor constraints can be expressed as 9 quadratic equality constraints in the real domain.

3.4.1.1 The First Step: n-node Decomposition

With $(n-3)$ virtual lines, any chordless cycle of n nodes can be decomposed into $(n-2)$ 3-node cycles. The cycle constraint of the chordless cycle will be replaced by $(n-2)$ cycle constraints related to those 3-node cycles.

Fig. 3.1 shows one example power network for the decompose strategy. Three virtual lines are added in the chordless cycle $\{4 \to 9 \to 8 \to 7 \to 6 \to 5 \to 4\}$. There are now four 3-node cycles presented: $\{4,5,6\}$, $\{4,6,7\}$, $\{4,7,8\}$, and $\{4,8,9\}$. In power grids, adding virtual lines is similar to claim that any two nodes without direct line connection can be viewed as connected through a line with infinite impedance. The numbers of virtual lines added for a n-node cycle is $(n-3)$ and the number of the resulting 3-node cycles is $(n-2)$.

Obviously, if all 3-node cycles satisfy the cycle constraint (3.2b), the original cycle condition can also be satisfied.

3.4.1.2 The Second Step: Exactness Conditions Conversion

Notate a 3×3 Hermitian matrix corresponding to a 3-node cycle as X_c. We will show that the exactness conditions for SOCP relaxation (3.2) related to a 3-node cycle is indeed equivalent to the corresponding matrix X_c being PSD and rank-1.

$$X_c = \begin{bmatrix} X_{c11} & X_{c12} & X_{c13} \\ X_{c21} & X_{c22} & X_{c23} \\ X_{c31} & X_{c32} & X_{c33} \end{bmatrix}$$

Its exactness condition is shown as follows.

$$X_{cii} \geq 0,\ i = 1, 2, 3 \tag{3.17a}$$

$$|X_{cij}| = \sqrt{X_{cii} X_{cjj}}, \quad (i, j) \in \{(1, 2), (2, 3), (3, 1)\} \tag{3.17b}$$

$$\angle X_{c12} + \angle X_{c23} + \angle X_{c31} = 0 \tag{3.17c}$$

(3.17b) indicates that all 2×2 principal minors equal to 0:

$$\begin{aligned} X_{c11} X_{c22} &= |X_{c12}|^2 = |X_{c21}|^2 \\ X_{c22} X_{c33} &= |X_{c23}|^2 = |X_{c32}|^2 \\ X_{c11} X_{c33} &= |X_{c13}|^2 = |X_{c31}|^2 \end{aligned} \tag{3.18}$$

Consider a non-principal minor M in X_c such as:

$$\begin{aligned} M = \det \begin{bmatrix} X_{c21} & X_{c23} \\ X_{c31} & X_{c33} \end{bmatrix} &= X_{c21} X_{c33} - X_{c23} X_{c31} \\ &= |X_{c21}||X_{c33}| \angle X_{c21} - |X_{c23}||X_{c31}| (\angle X_{c23} + \angle X_{c23}) \end{aligned}$$

According (3.18) and (3.17c), we can derive:

$$\begin{aligned} & \begin{cases} |X_{c21}||X_{c33}| = |X_{c23}||X_{c31}| \\ \angle X_{c21} = \angle X_{c23} + \angle X_{c23} \end{cases} \\ \Rightarrow \quad & |X_{c21}||X_{c33}|\angle X_{c21} = |X_{c23}||X_{c31}|(\angle X_{c23} + \angle X_{c23}) \\ \Rightarrow \quad & M = 0 \end{aligned}$$

Through the similar processes, we can find all non-principal minors are zeros. Since all 2×2 minors are 0, the rank of X_c is less than 2. And the diagonal components of X_c are all greater than 0. Hence X_c is PSD and rank-1.

3.4.1.3 The Third Step: Constraints Reformulation

Basing on the previous step and the Proposition 3.1 [22], for each 3-node cycle, the exactness condition or X_c is PSD and rank-1 can be replaced by a set of quadratic minor constraints. The minor constraints of X_c can be expressed as constraints (3.19g).

As a summary, for a power grid that has been decomposed into lines (the set is notated as $\mathcal{L}$), virtual lines (notated as $\mathcal{L}_\mathcal{V}$), and 3-node cycles (notated as Ψ), there will be $|\mathcal{L}|+|\mathcal{L}_\mathcal{V}|$ constraints related to the principal minors, and $6 \times |\Psi|$ constraints related to non-principal minors. Take the example shown in Fig. 3.1, the system has a total 9 lines with one 6-node chordless cycle. Three virtual lines are added $4-6$, $4-7$, $4-8$ to decompose the chordless cycle into 4 3-node cycles. The system's exactness constraints consist of 12 constraints related to the 12 principal minors corresponding to 12 lines (9 lines and 3 virtual lines) and 6×4 related to the non-principal minors. Total, there are 36 quadratic equality constraints.

$$R_{ij}^2 + I_{ij}^2 = R_{ii}R_{jj}, \quad (i,j) \in \{(1,2),(2,3),(3,1)\} \tag{3.19a}$$

$$R_{12}R_{23} - I_{12}I_{23} - R_{22}R_{13} = 0, \tag{3.19b}$$

$$I_{12}R_{23} + R_{12}I_{23} - R_{22}I_{13} = 0, \tag{3.19c}$$

$$R_{23}R_{13} + I_{23}I_{13} - R_{33}R_{12} = 0, \tag{3.19d}$$

$$I_{23}R_{13} - R_{23}I_{13} + R_{33}I_{12} = 0, \tag{3.19e}$$

$$R_{13}R_{12} + I_{13}I_{12} - R_{11}R_{23} = 0, \tag{3.19f}$$

$$I_{13}R_{12} - R_{13}I_{12} - R_{11}I_{23} = 0. \tag{3.19g}$$

3.4.2 Rank-1 PSD Matrix-Based Nonlinear Programming Formulation

In Section 3.4.1, the exactness conditions have been converted to quadratic equality constraints. A nonlinear programming problem can now be formulated with those constraints. The next step is to identify cycles in a power network and decompose any chordless cycle with size greater than 3 into 3-node cycles. With all virtual lines and 3-node cycles identified, the nonlinear programming formulation can be derived.

3.4.2.1 Cycle Basis Identification

A cycle basis of a graph is the set of cycles with each cycle having only one of its edges common with the spanning tree of the graph. We use the cycle identification algorithm in [54] to identify the cycles. A MATLAB based toolbox [55] is applied to find a cycle basis. The algorithm first searches a minimal spanning tree of the network and then adds the rest of the lines back one by one. Each added line will be considered as a token to identify one cycle. For example, consider the five buses network in Fig. 3.2, the set of the lines is $\mathcal{L} = \{(1,2),(2,3),(3,4),(4,5),(5,1),(4,1)\}$. To identify its cycles, first the cycle identification algorithm will start from the minimal spanning tree in Fig. 3.3. This spanning tree contains lines $(1,2),(2,3),(1,4),(1,5)$. Next, the remaining lines are added back one by one. After adding $(3,4)$, we obtain a cycle $c_a = \{1-2-3-4-1\}$. After adding $(4,5)$, we get a cycle $c_b = \{1-4-5-1\}$.

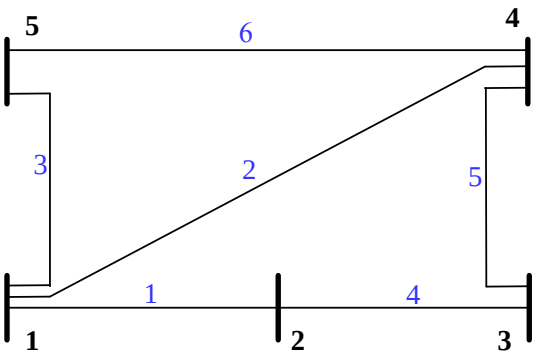


Figure 3.2: Five-bus test case with two cycles. Cycle a: nodes $\{1, 2, 3, 4\}$, lines $\{(1, 2), (2, 3), (3, 4), (4, 1)\}$; Cycle b: nodes $\{1, 4, 5\}$, lines $\{(1, 4), (4, 5), (5, 1)\}$.

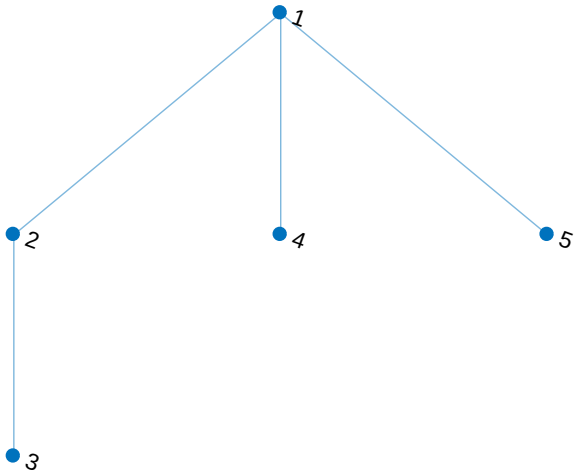


Figure 3.3: Spanning tree of the five-bus test system.

3.4.2.2 Nonlinear Programming Problem Formulation

Through the cycle identification algorithm and the graph decomposition process, we obtain a set of virtual lines and 3-node cycles. The set of all lines including virtual lines are notated as $\mathcal{L}_{ch} = \mathcal{L}_{\mathcal{V}} \cup \mathcal{L}$. The proposed rank-1 nonlinear programming formulation is shown in the problem (3.20).

This problem is a quadratically constrained quadratic program (QCQP) problem with quadratic equality constraints (3.20h) to (3.20n). The problem can be solved by nonlinear programming solvers, e.g., IPOPT.

Compared with MATPOWER, the decision variables in (3.20) are no longer voltage phasors. Instead, R and I can be initialized using the solution of the PSD matrix X from a SDP relaxation ACOPF solver developed in [65].

$$\min \quad \sum_{k\in\mathcal{G}} C_{2k}P_{gk}^2 + C_{1k}P_{gk} + C_{0k} \tag{3.20a}$$

$$P_i^g - P_i^d = \sum_{j\in Adj_i} (G_{ij}R_{ij} - B_{ij}I_{ij}), \quad i \in \mathcal{N} \tag{3.20b}$$

$$Q_i^g - Q_i^d = \sum_{j\in Adj_i} (-G_{ij}I_{ij} - B_{ij}R_{ij}), \quad i \in \mathcal{N} \tag{3.20c}$$

$$|S_{ij}(R_{ii}, R_{ij}, I_{ij})| \leq S_{ij}^{\max}, \quad (i,j) \in \mathcal{L} \tag{3.20d}$$

$$P_{gi}^{\min} \leq P_{gi} \leq P_{gi}^{\max}, \quad i \in \mathcal{G} \tag{3.20e}$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, \quad i \in \mathcal{G} \tag{3.20f}$$

$$(V_i^{\min})^2 \leq R_{ii} \leq (V_i^{\max})^2, \quad i \in \mathcal{N} \tag{3.20g}$$

$$R_{ij}^2 + I_{ij}^2 = R_{ii}R_{jj}, \quad (i,j) \in \mathcal{L}_{ch} \tag{3.20h}$$

For each 3 nodes cycle $c = \{i, j, k\} \in \Phi$

$$R_{ij}R_{jk} - I_{ij}I_{jk} - R_{jj}R_{ik} = 0, \tag{3.20i}$$

$$I_{ij}R_{jk} + R_{ij}I_{jk} - R_{jj}I_{ik} = 0, \tag{3.20j}$$

$$R_{jk}R_{ik} + I_{jk}I_{ik} - R_{kk}R_{ij} = 0, \tag{3.20k}$$

$$I_{jk}R_{ik} - R_{jk}I_{ik} + R_{kk}I_{ij} = 0, \tag{3.20l}$$

$$R_{ik}R_{ij} + I_{ik}I_{ij} - R_{ii}R_{jk} = 0, \tag{3.20m}$$

$$I_{ik}R_{ij} - R_{ik}I_{ij} - R_{ii}I_{jk} = 0. \tag{3.20n}$$

3.4.3 Voltage Recover Techniques

The solution from (3.20) is a set of R_{ii}, R_{ij}, and I_{ij}, or partial information of a PSD matrix X. Voltage vectors can be recovered from the decision variables R and I. R_{ii} will be found for

every bus. R_{ij} and I_{ij} are related to all lines and virtual lines. we implement a voltage recovery method [6] through visiting the spanning tree of the network.

First, we define the phase angle of voltage phasor at the reference node as 0°. We identify the spanning tree of the power network, and define path from the reference node to the node i as $\mathcal{P}_i$. Then the voltage phasor at any node i can be recovered through the following equations.

$$V_i = \sqrt{R_{ii}}$$
$$\theta_i = -\sum_{(j,k)\in\mathcal{P}_i} \angle(R_{jk} - jI_{jk})$$

3.5 Case Studies

In this section, we presented the case studies for the two proposed algorithms. First, the 3-node cycle-based convex iteration algorithm is tested on some cases which are selected from the Table 2.2. These cases shown relatively large gaps. The converge process of the rank errors are plotted for the "case9mod" and "nesta_case300_ieee". Second, we tested the nonlinear rank-1 formulation on two sets of cases. The first set of cases are chosen from the NICTA test archive [57]. Gaps of these cases are small or zeros. The second set of cases are special cases which are modified to have large gaps. These cases are selected from the test achieves which are provided by [17] and [5]. The related results are listed in the Table 3.2 and 3.3 respectively. Our numerical experiments are conducted on an Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30 GHZ (2 processors) computer. All solvers are implemented in MATLAB 2017a-based CVX platform [58].

3.5.1 Rank-1 Solution Through Convex Iteration

We tested 3-node cycle-based convex iteration method on several non-zero gap cases. In the experiments, we keep $\epsilon = 10^{-5}$ used in (3.13) for all cases. Commercial solver Mosek 7.1.0.12 are called in the script to complete the computation. The results are listed in Table 3.1. Column Nlines is the sum of the numbers of lines and virtual lines in a graph; Niter is the number of iterations. In this Table, we apply maximum active and reactive power mismatch to show the feasibility of the solution, where superscript before means before convex iteration, and after means after convex iteration. The mismatches are calculated through the power balancing equations using

the voltage phasor vector recovered from the solution of a sparse matrix X, where superscript sol means the solutions directly from Solver C with or without convex iteration; superscript rec means the solutions obtained through the recovered voltage phasors after the solutions from the proposed solver. V^{rec} and θ^{rec} are the magnitude and angle of the recovered voltage phasor vector. The gap is defined as: $Gap = \frac{UB-LB}{UB} \times 100\%$, where UB is the upper bound which is calculated through MATPOWER; in Table 3.1, LB is calculated by the convex iteration solver.

Table 3.1: Convex iteration results

case	UB	gap(%)	Nlines	Niter	MaxRank	$P_{\text{mis}}^{\text{before}}$	$Q_{\text{mis}}^{\text{before}}$	$P_{\text{mis}}^{\text{after}}$	$Q_{\text{mis}}^{\text{after}}$	ω
nesta_case5_pjm	17551.89	**0.00**	7	2	1	0.57	2.17	**6.27e-6**	**1.46e-5**	28000
nesta_case30_fsr_api	372.14	**0.08**	55	5	1	6.89e-2	3.71e-2	**3.81e-5**	**5.54e-5**	1210
nesta_case118_ieee_api	6383.57	**0.00**	252	6	1	9.26	4.12	**4.13e-4**	**5.25e-4**	560
nesta_case118_ieee	3689.92	**0.00**	252	4	1	5.80e-1	6.55e-1	**2.36e-4**	**1.90e-4**	22
nesta_case300_ieee	16891.28	**0.00**	602	6	1	8.12e-1	4.64e-1	**1.45e-4**	**1.87e-4**	1000
nesta_case1354_pegase	74064.11	**0.00**	2408	3	1	9.73e-1	6.61e-1	**5.27e-4**	**2.05e-3**	821.5
case9mod	4267.07	**22.89**	12	18	1	7.49e-3	2.59e-1	**2.66e-9**	**1.03e-7**	1e6
case39mod1	11221.00	**0.00**	67	19	1	9.92e-2	1.73	**1.38e-6**	**1.53e-5**	1.98e5
case300mod	378540.50	**0.00**	602	21	1	7.13	1.79	**5.95e-5**	**5.71e-4**	3.91e4

The voltage vector recovering method in [44] is adopted in this project. First, we define the phase angle of the voltage phasor at the reference node as 0°; next, we identify the spanning tree of the power network, and define the unique path from the reference node to the node i as $\mathcal{P}_i$; then the voltage phasor at any node i can be recovered through the following equations.

$$V_i^{\text{rec}} = \sqrt{c_{ii}}, \quad \theta_i^{\text{rec}} = -\sum_{(j,k)\in\mathcal{P}_i} \angle(c_{jk} - j s_{jk})$$

The results in Table 3.1 show that the proposed convex iteration is capable of decreasing maximum rank of submatrices. In Fig.3.4 we show that the rank error represented by $\sum Tr(\widehat{X}^{(i)}\widehat{W}^{(i)})$ decreasing for two instances. According to Table 3.1, all 9 instances successfully achieve rank-1 , the power mismatches are close to 0, and gaps are non-negative. It means the recovered voltage vectors from the convex iteration solutions are feasible for the original ACOPF, and the objective value is not worse than the MATPOWER. We noted in Table 3.1, the gap for case9mod case is 22.89% while MaxRank is one. The reason of this situation is that the case9mod case has multi-local optimal solutions [5]. The MATPOWER which is based on the interior point method obtained one of the

local optimal solution, while the convex iteration solver obtained another optimal solution which is closer to the global optimal.

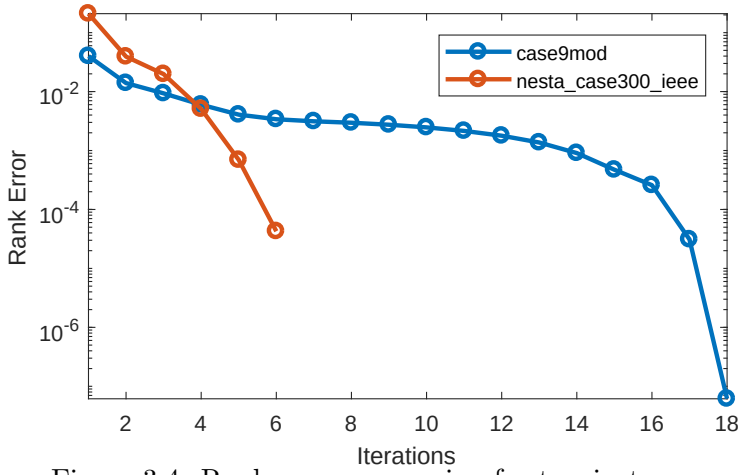


Figure 3.4: Rank error converging for two instances.

3.5.2 Nonlinear Rank-1 Formulation

In this section, we present the case study results for nonlinear rank-1 formulation. The proposed nonlinear programming ACOPF formulation was implemented in MATLAB using Yalmip and select IPOPT as the main solver. The configurations of the IPOPT are: convergence tolerance is 1×10^{-7}; maximum number of iteration is 10000; update strategy for barrier parameter is "adaptive"; Hessian information is "limit-memory"; and the other configurations are default.

To obtain the initial point for the nonlinear solver, we solve OPF problems first through a CVX [66] based sparse SOCP/SDP relaxation solver [65]. The objective value of the SOCP/SDP solver and the maximal rank of the corresponding PSD submatrices will be given for each test case. MATPOWER with IPOPT as the solver is used for comparison.

In results tables, the column "Rank" list the maximal rank of all submatrices for each case. For each submatrix, rank is counted by considering only the eigenvalues that are greater than the 0.001% of the maximal eigenvalue; N_vlines is the number of the virtual lines; N_lines is the number of the original lines; N_cycle3 is the number of the 3-node cycles based on the decomposition results.

In Table 3.2, the results show that for all tested standard cases, our formulation provides the same objective values as those from MATPOWER. In Table 3.3, we listed test results on

modified cases. From the table, we can see that for cases with large gaps, the proposed formulation is capable to obtain the same objective value as MATPOWER. These results indicate that the proposed formulation can provide similar quality results to the original ACOPF formulation that is implemented in MATPOWER. Moreover, according the number of 3-node cycles, we can see the improvement on the number of the added equality constraints. For example, in "nesta_case57_ieee", N_vlines= 55,N_lines= 78, N_cycle3= 77, which means the number of the added equality constraints are $77 \times 6 + 78 + 55 = 595$. This number is much less than $C_{57}^2 C_{57}^2 = 2547216$ which is the number of the added equality constraints without the decomposition. For the cases from "case9Tree" to "case57Tree" in table 3.3, as they are radial networks, we do not add any virtual line on them.

Table 3.2: Test case results

	SDP Relaxation			Matpower	Nonlinear Rank 1 Method				
Case	Obj	Rank	%Gap	Obj	Obj	%Diff	N_vlines	N_lines	N_cycle3
nesta_case3_lmbd	5789.914	2	0.04	5812.643	5812.643	0.00	0	3	1
nesta_case5_pjm	16635.76	2	5.22	17551.891	17551.889	0.00	1	6	3
nesta_case9_wscc	5296.685	2	0.00	5296.686	5296.686	0.00	3	9	4
nesta_case57_ieee	1143.280	2	0.00	1143.283	1143.283	0.00	55	78	77
nesta_case89_pegase	5820.217	2	0.00	5820.387	5820.387	0.00	55	206	173
nesta_case118_ieee	3689.495	3	0.01	3692.891	3692.891	0.00	73	179	135

Table 3.3: Special test case results

	SDP Relaxation			Matpower	Nonlinear Rank 1 Method				
Case	Obj	Rank	%Gap	Obj	Obj	%Diff	N_vlines	N_lines	N_cycle3
WB5	946.530	2	0.01	946.584	946.584	0.00	1	6	3
case9mod	2753.041	2	10.84	3087.842	3087.842	0.00	3	9	4
case39mod1	10804.055	2	3.72	11221.003	11221.003	0.00	21	46	29
case39mod2	940.341	2	0.15	941.738	941.738	0.00	21	46	29
case9Tree	5335.701	2	52.70	11279.476	11279.476	0.00	0	8	0
case14Tree	11861.899	2	0.59	11932.252	11932.252	0.00	0	13	0
case30Tree	4244.549	2	11.47	4794.313	4794.314	0.00	0	29	0
case39Tree	44868.452	2	0.37	45037.039	45037.042	0.00	0	38	0
case57Tree	10458.099	2	13.58	12100.849	12100.856	0.00	0	56	0

3.6 Conclusion

In this chapter, an efficient convex iteration implementation is also investigated for the proposed sparse convex solver to achieve exactness or rank-1 solutions. Our experiment results show the feasibility of the implementation. Moreover, we proposed a nonlinear programming formulation for

ACOPF. This formulation is based on decision variables that align with SOCP/SDP relaxation. The proposed formulation exploits power network sparsity feature and employs a small set of minor constraints related to all 3-node cycles as equality constraints to enforce rank-1 constraint, so the solution is exact. Case study results demonstrate the correctness of this formulation.

Chapter 4: Benders' Decomposition for MPC of a Modular Multi-level Converter

4.1 Introduction

[3]Compared with traditional two-level VSC, multilevel voltage-source converters have much lower harmonic in the output voltage, which significantly reduces the size of grid side filter [68]. Among the different multilevel converters, MMC has modular topology and the extensibility for several hundreds of output voltage levels. Therefore, MMC is ideal for high-voltage high-power applications, such as HVDC transmission [69], high-voltage motor drives [70], and electric railways [71]. Fig. 4.1 is the topology of a three phase MMC. For an $N+1$ level MMC, there are N sub-modules on each arm of the converter. Each sub-module is a half bridge dc-dc converter. Since the current flows through different sub-modules at different times, the voltages of sub-modules capacitors vary.

4.1.1 State-of-the-art MMC Switching Schemes

MMC control differs from two-level VSC control in two aspects: 1) switching sequence generation and 2) the inclusion of circulating current mitigation control. In switching sequence generation, in two-level VSCs, the output from PWM is the switching sequence directly fed to the gates. In MMCs, due to the large number of sub-modules, the output of pulse-width modulation or other types of switching schemes is the number of sub-modules to be turned on at each arm. Which sub-modules to be turned on then depends on additional sub-module voltage balance consideration. The PWM switching schemes are also very different from that of two-level VSCs. Phase-disposition (PD)-PWM and Phase shifted-PWM are often adopted [72]. In MMC's PWM, there are usually many carrier signals for the reference sinusoidal signal to be compared to; while in two-level VSC's PMW, there is usually one triangular carrier signal.

[3]This Chapter was published in 2017 North American Power Symposium (NAPS) [67]. Permission is included in the Appendix A.

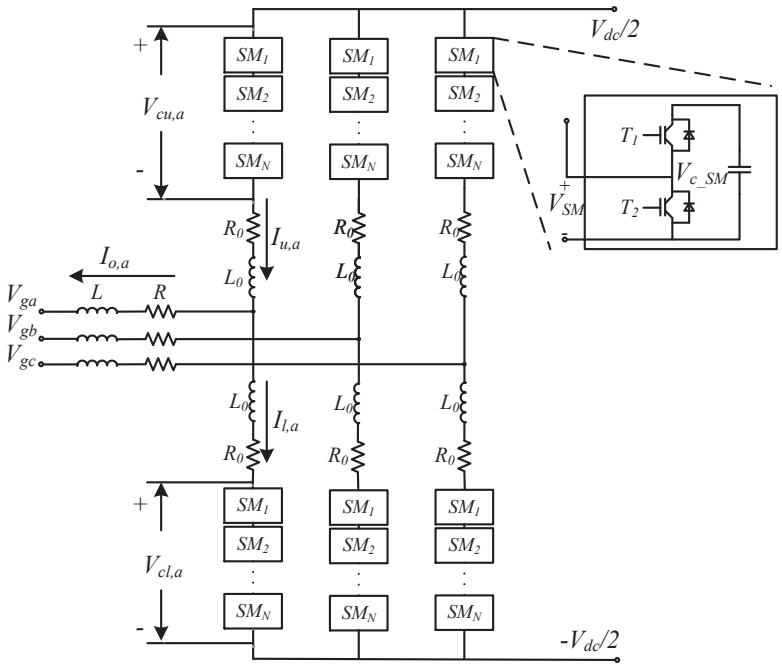


Figure 4.1: Three phase MMC topology.

MMC PWM gives only the number of modules to be switched on. To determine which modules to be switched on, another phase is required: capacitor voltage balancing. A capacitor voltage balancing block selects the proper sub-modules to be switched.

4.1.2 Our Contributions

As an advanced control method, MPC is very successful on its application for the control of power converters [25]. Its basic principle is to generate a system dynamic model based minimizing optimization problem, and provide the solutions to the controller for driving the system to reach the control target (Generally will be formulated as the objective function in the MPC optimization problem). A major challenge for implementing MPC on MMC is because the system dynamic model of MMC is nonlinear with binary terms, in other word, the MPC problem of MMC is a Mix-integer programming problem which generally is difficult to solve precisely.

In this chapter, we adopt Benders' decomposition to solve the MPC problem. Our major focus are the derivation for the MPC problem formulation and its Benders' decomposition format. In the case study section, the potential and practicability of the this application are verified, but also

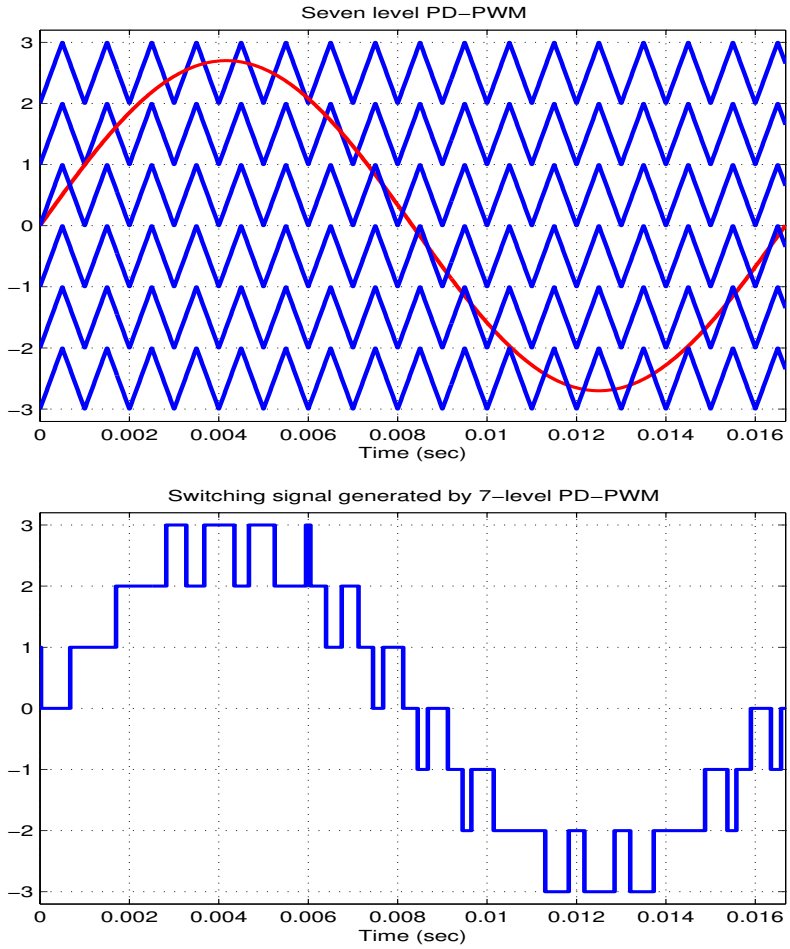


Figure 4.2: Three-level and Seven-level VSC PD-PWM scheme and switching status. (a) Three-level VSC PD-PMW scheme and switching status of a phase.(b) Seven-level VSC PD-PWM scheme and switching status of a phase.

revealed some limitations of it. The further discussion about how to overcome these limitation will be the topic of our next chapter, so it is not included in this one.

The rest of the chapter is organized as follows. Section 4.2 gives the dynamic model of MMC. Section 4.3 gives the details about the formulation of the MPC problem. And then, its Benders' algorithm format are derived in Section 4.4. Section 4.5 presents the case study and results. Finally, the chapter are concluded in Section 4.6.

4.2 Dynamic Model of MMC

Fig. 4.1 shows the overall structure of a three-phase MMC consisting of six arms. Subscripts u and l denote upper and lower arms, respectively. There are N sub-modules and one inductor L_0 on each arm. A resistor R_0 is inserted to represent the switching loss of the IGBTs on each arm. The output voltage of each sub-module has two values, U_c (when T_1 is connected) and 0 (when T_2 is connected). When the number of sub-modules or the switching frequency is high enough, the voltage across whole sub-modules in each arm can be considered as continuous. Since the dc side capacitors are usually big enough, the voltage across the arm can be considered as constant dc voltage sources. Thus, we can express a single phase-equivalent circuit of a MMC as Fig. 4.3.

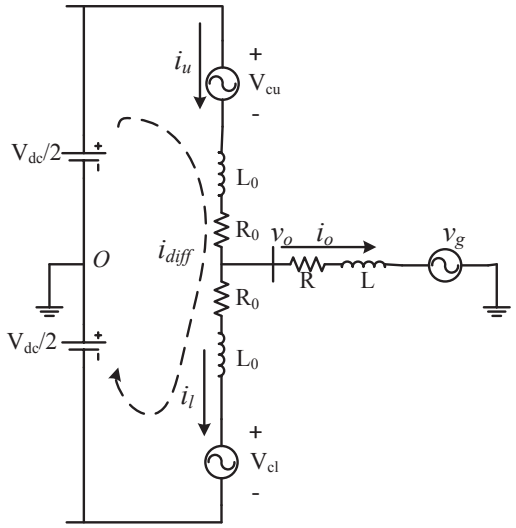


Figure 4.3: Single phase equivalent circuit of MMC.

In Fig. 4.3, i_u and i_l are the arm currents for upper and lower arms; i_o and v_o are the converter output current and voltage respectively. The circulating current flowing within the converter is denoted as i_z. Since the upper and lower arm are symmetric, ideally both lower and upper arm currents contain half of the converter output current. Therefore, with KCL, we can get following equations:

$$\begin{cases} i_u = i_z + \frac{i_o}{2} \\ i_l = i_z - \frac{i_o}{2} \end{cases} \Rightarrow \begin{cases} i_z = \frac{i_u + i_l}{2} \\ i_o = i_u - i_l. \end{cases} \tag{4.1}$$

The voltage across the arm resistance and inductance can be expressed by the arm current. Therefore, with KVL, we can have the voltage relationship as follow:

$$\begin{cases} v_u + i_u R_0 + L_0 \frac{\mathrm{d}i_u}{\mathrm{d}t} = \frac{V_{\mathrm{dc}}}{2} - v_o \\ v_l + i_l R_0 + L_0 \frac{\mathrm{d}i_l}{\mathrm{d}t} = \frac{V_{\mathrm{dc}}}{2} + v_o \end{cases} \tag{4.2}$$

Considering that the output voltage v_o can be written as $v_g + i_o R + L\frac{\mathrm{d}i_o}{\mathrm{d}t}$ and (4.1), by subtracting the two equations from (4.2) we have:

$$\frac{v_u - v_l}{2} + \left(R + \frac{1}{2}R_0\right) i_o + \left(L + \frac{1}{2}L_0\right) \frac{\mathrm{d}i_o}{\mathrm{d}t} + v_g = 0 \tag{4.3}$$

It is obvious that the term $\frac{v_u - v_l}{2}$ in (4.3) drives the output current of the converter, therefore we name this term as e, which is the inner emf of the converter. We can have an equivalent circuit of MMC as Fig. 4.4, which is the plant model of inner current control loop of an MMC.

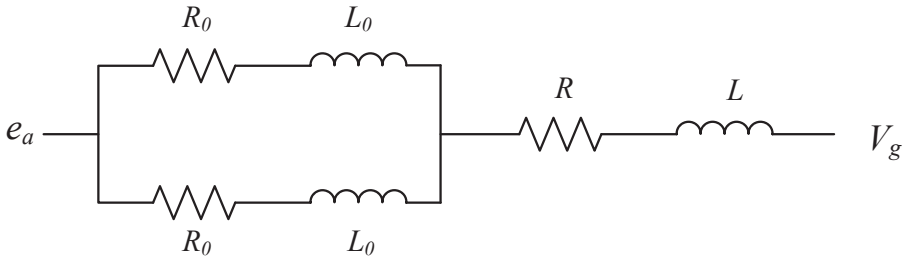


Figure 4.4: An equivalent circuit of one phase of MMC.

Adding the two equations in (4.2) leads to the plant model of the circulating current control:

$$i_z R_0 + L_0 \frac{\mathrm{d}i_z}{\mathrm{d}t} = \frac{V_{\mathrm{dc}}}{2} - \frac{v_u + v_l}{2}. \tag{4.4}$$

Considering Fig. 4.1, since v_u and v_l are the sum of all sub-module voltages on the correspond arm, we can express v_u and v_l through the following equations:

$$\begin{cases} v_u = \sum_{i=1}^{N} V_{SMi} & \text{i on the upper arm} \\ v_l = \sum_{i=N+1}^{2N} V_{SMi} & \text{i on the lower arm} \end{cases} \tag{4.5}$$

For convenient, we name the voltage of the capacitor on each sub-module as $v_c(i)$, and the state of the correspond IGBT as $u(i)$. Apparently, $V_{SMi} = v_c(i)u(i)$, combine it with the equation (4.5), then we have:

$$\begin{cases} v_u = \sum_{i=1}^{N} v_c(i) & \text{i on the upper arm} \\ v_l = \sum_{i=N+1}^{2N} v_c(i) & \text{i on the lower arm} \end{cases} \tag{4.6}$$

4.3 Optimization Problem Formulation

We propose to discretize the continuous dynamic model of MMC which described by (4.1) to (4.6). It means we consider

$$\frac{di_o}{dt} = \frac{i_o(k+1) - i_o(k)}{h} \tag{4.7}$$

$$\frac{di_z}{dt} = \frac{i_z(k+1) - i_z(k)}{h} \tag{4.8}$$

where h is step size of the discretized signal, $k \in \{1, 2, \cdots, T\}$ is the index of the time step. Combining (4.7) and (4.8) respectively with (4.3) and (4.4), then we obtain the following equations:

$$i_o(k+1) = i_o(k) + \frac{h}{L + \frac{L_0}{2}} \left[-\left(R + \frac{R_0}{2} \right) i_o(k) - v_g(k) - \frac{v_u(k) - v_l(k)}{2} \right] \tag{4.9}$$

$$i_z(k+1) = i_z(k) + \frac{h}{L} \left[-R_0 i_z(k) + \frac{V_{dc}}{2} - \frac{v_u(k) + v_l(k)}{2} \right] \tag{4.10}$$

where

$$\begin{cases} v_u(k) = \sum_{i=1}^{N} v_c(i,k)u(i,k) \\ v_l(k) = \sum_{i=N+1}^{2N} v_c(i,k)u(i,k) \end{cases} \tag{4.11}$$

In (4.11), N is the numbers of switches on one arm, $i \in \{1, 2, \cdots, 2N\}$ is the index of switches. We expect to control the MMC output current to track the current reference which is a sinusoidal waveform. This can be expressed by solving the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^{T} \left[i_o^{\text{ref}}(k) - i_o(k) \right]^2 \\ \text{subject to} \quad & (4.9), (4.10), \\ & v_u(k) = \sum_{i=1}^{N} v_c(i,k)u(i,k) \\ & v_l(k) = \sum_{i=N+1}^{2N} v_c(i,k)u(i,k) \\ & v_c(i,k+1) = \begin{cases} v_c(i,k) + \frac{h}{c} \cdot u(i,k) \left[\frac{i_o(k)}{2} + i_z(k) \right], \\ \qquad i = 1, 2, \cdots, N. \\ v_c(i,k) + \frac{h}{c} \cdot u(i,k) \left[-\frac{i_o(k)}{2} + i_z(k) \right], \\ \qquad i = N+1, \cdots, 2N \end{cases} \\ & \sum_{i=1}^{2N} u(i,k) = N \\ & u(i,k) \in \{0, 1\} \end{aligned} \tag{4.12}$$

where i_o^{ref} is our desired current, $u(i,k)$ is the state of the ith switch at the kth time step, $v_c(i,k)$ is the voltage on the ith switch capacitor at the kth time step.

If we replace the binary constraint $u(i,k) \in \{0,1\}$ by an equality constraint $u(i,k)(1-u(i,k)) = 0$ with $u(i,k)$ as continuous variable, then this problem can be solved by nonlinear programming solver *fmincon* using sequential quadratic programming (SQP) algorithm.

4.4 Benders' Decomposition Formulation

To implement Benders' decomposition for solving the problem (4.12), we separate the problem to a master problem and a sub-problem. In the sub-problem problem, the binary variable u is considered as fixed values. Therefore the sub-problem could be solved as a linear programming problem. Iteratively, the dual variables which are solved from the sub-problem will be used to generate the Benders' cuts and add them to the master problem. And then, the solution of u from the master problem will be returned to the sub-problem. This iteration process will be repeated until the stop criteria are met.

4.4.1 Subproblem

In our case, decision variables in the sub-problem are: i_o, i_z, v_u, v_l, and v_c, and define any given u as $\hat{u}$, its primal problem could be expressed as follow:

$$\min \quad \sum_{k=1}^{T} \left[i_o^{\text{ref}}(k) - i_o(k) \right]^2 \tag{4.13}$$

$$\text{subject to} \quad (4.9), (4.10)$$

$$v_u(k) = \sum_{i=1}^{N} v_c(i,k)\hat{u}(i,k)$$

$$v_l(k) = \sum_{i=N+1}^{2N} v_c(i,k)\hat{u}(i,k)$$

$$v_c(i,k+1) = \begin{cases} v_c(i,k) + \frac{h}{c} \cdot \hat{u}(i,k) \left[\frac{i_o(k)}{2} + i_z(k) \right] \\ \qquad i = 1, 2, \cdots, N. \\ v_c(i,k) + \frac{h}{c} \cdot \hat{u}(i,k) \left[-\frac{i_o(k)}{2} + i_z(k) \right] \\ \qquad i = N+1, \cdots, 2N. \end{cases}$$

To generate the Benders' cuts, we need to find dual variables which correspond with the constraints that include u. To achieve this, first we define a Lagrangian function to aggregate the

objective function and the constraints that are related to u.

$$
\begin{aligned}
L = & \sum_{k=1}^{T} \left[i_o^{\text{ref}}(k) - i_o(k) \right]^2 \\
& + \sum_i \sum_k \lambda_1(i,k) \left\{ v_c(i,k+1) - v_c(i,k) - \frac{h}{c} u(i,k) \left[\frac{i_o(k)}{2} + i_z(k) \right] \right\} \\
& + \sum_i \sum_k \lambda_2(i,k) \left\{ v_c(i,k+1) - v_c(i,k) - \frac{h}{c} u(i,k) \left[-\frac{i_o(k)}{2} + i_z(k) \right] \right\} \\
& + \sum_k \lambda_3(k) \left[v_u(k) - \sum_{i=1}^{N} v_c(i,k) u(i,k) \right] \\
& + \sum_k \lambda_4(k) \left[v_l(k) - \sum_{i=1+N}^{2N} v_c(i,k) u(i,k) \right].
\end{aligned}
$$

The partial dual of the original problem could be formulated as follows.

$$
\begin{aligned}
\max_{\lambda} \quad & \min L \\
\text{subject to} \quad & (4.9), (4.10)
\end{aligned}
$$

4.4.2 Cuts Introduced by the Subproblem

We formulate cuts in the following inequality constraint:

$$
\mu \geq v_{LB}^l \left[\hat{u}^l(i,k) \right] + \sum_k \sum_i g^l(i,k) \left[u(i,k) - \hat{u}^l(i,k) \right] \tag{4.14}
$$

where v_{LB} is the optimal solution of the subproblem while $u = \hat{u}$, l is the index of the cuts, matrix g is defined as:

$$
g = \begin{bmatrix}
\frac{\partial L(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)}{\partial u(1,1)} & \cdots & \frac{\partial L(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)}{\partial u(1,T)} \\
\vdots & \ddots & \vdots \\
\frac{\partial L(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)}{\partial u(2N,1)} & \cdots & \frac{\partial L(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)}{\partial u(2N,T)}
\end{bmatrix}
$$

Thus

$$
\begin{aligned}
g(i,k) = & -\lambda_1(i,k) \frac{h}{c} i_u(k) - \lambda_2(i,k) \frac{h}{c} i_l(k) \\
& - \lambda_3(k) v_{ci}(k) - \lambda_4(k) v_{c,i+N}(k)
\end{aligned}
$$

4.4.3 Master Problem

Associating Benders' cuts, the master problem can be written as the problem (4.15). Apparently, problem (4.15) is a mixed integer linear programming problem which can be solved by gurobi or Mosek. In the Benders' algorithm, the values of the objective function for the master problem are the upper bounds of the optimal values for the original problem. And the values of the objective function for the sub-problem are the lower bounds. Therefore, through the iteration, if the objective function's values of the master problem and sub-problem are close enough to reach a stop criteria, we can consider the optimal solutions are found. Generally, the criteria are defined as:

$$\epsilon >= |V_{mas} - V_{sub}|$$

where V_{mas} is the objective function's value of the master problem, V_{sub} is the objective function's value of the sub-problem, and ϵ is a fixed small value.

$$\begin{aligned}
\min \quad & \mu && \text{(4.15a)}\\
\text{subject to} \quad & \sum_{i=1}^{2N} u_i(k) = N && \text{(4.15b)}\\
& u_i(k) \in \{0,1\} && \text{(4.15c)}\\
& \mu \geq v_{LB}^{l} \left[\hat{u}^l(i,k)\right] \\
& \quad + \sum_k \sum_i g^l(i,k) \left[u(i,k) - \hat{u}^l(i,k)\right] && \text{(4.15d)}
\end{aligned}$$

4.5 Case Study

In this section, we select a $N = 4$ MMC as the platform for testing the effectiveness of the Benders' algorithm. The parameters of the MMC and MPC time step size are listed in Table 4.1. Matlab CVX toolbox with Mosek solver are applied to solve its master and sub-problem. Moreover, we set the stop criteria constant $\epsilon = 1 \times 10^{-5}$. For comparison, we also develop a group of MATLAB codes to solve the problem (4.12) via fmincon function based SQP method, and separately list the

Table 4.1: Parameters table

Items	Values
submodule capacitor (C_{sm})	$2500\mu F$
insert inductor (L_0)	10 mH
insert resistor (R_0)	0.1Ω
terminal inductor (L)	2 mH
terminal resistor (R)	0.03Ω
DC voltage (V_{DC})	40 kV
rated frequency(f)	60 Hz
grid voltage (v_s)	$20\sin(2\pi f \cdot t)$ kV
reference current (i_o^{ref})	$5\sin(2\pi f \cdot t)$ kA
Prediction step size (h)	25 μs

solutions based on two methods about their objective value and solving time cost for different predict horizon in Table 4.2.

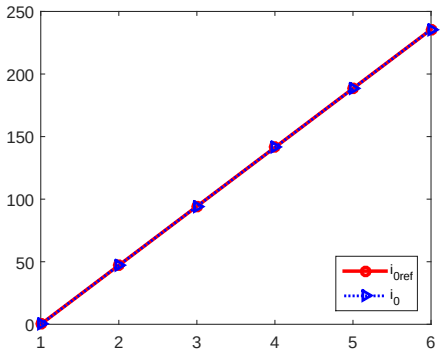


Figure 4.5: Current tracking of Benders' algorithm when $T = 5$

Fig. 4.5 and Fig. 4.6 present, when using Benders' algorithm, the current tracking performance and the convergence of the objective value for sub-problem and master problem when $T = 5$; Fig. 4.7 and Fig. 4.8 show the performance and convergence when the predict horizon is 7; Table 4.3 list the solutions of the integer variables for the correspond T. From those results, we can see the performance of the Benders' algorithm are pretty good: the current tracking are nearly prefect, the solutions of the integer values are completely feasible, and the objective value of the master problem and sub-problem are converged within a reasonable iteration steps. Moreover, according

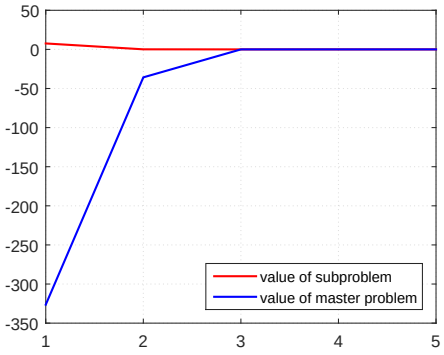


Figure 4.6: Convergence of the objective value for the master problem and sub-problem when $T = 5$.

the data in Table 4.2, which is proposed to compare the solutions of the problem (4.12) by Benders' algorithm and SQP method, we can see the Benders' algorithm has some obvious advantages than SQP method.

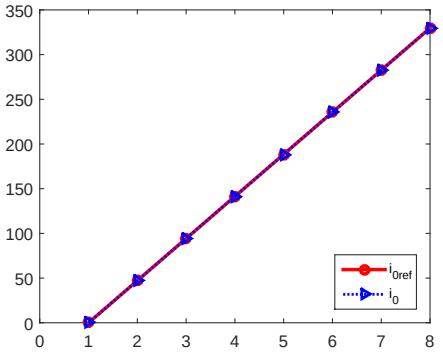


Figure 4.7: Current tracking of Benders' algorithm when $T = 7$.

Table 4.2: Results comparison of Benders' decomposition and SQP algorithm

T	Nonlinear		Benders		
	Obj	T (sec)	UB	LB	T (sec)
5	2.90	11.19	1.11e-16	1.11e-16	0.77
7	7.24	39.96	2.80e-6	2.80e-6	1.87
10	22.28	40.70	0.16	-0.72	56.61

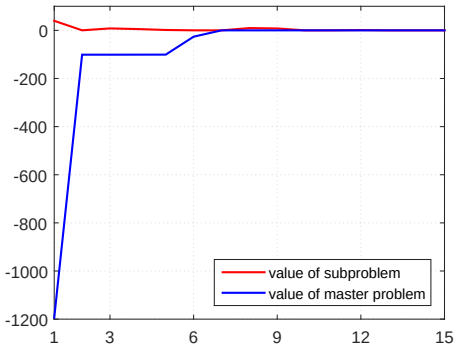


Figure 4.8: Convergence of the objective value for the master problem and sub-problem when $T = 7$.

Table 4.3: Binary solution

$T = 5$					$T = 7$						
0	1	0	0	1	0	1	1	0	0	1	1
0	0	1	1	0	0	0	0	0	1	1	0
1	0	0	1	0	1	0	0	1	1	0	1
0	1	1	0	0	1	0	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	0	1
1	0	1	0	1	1	1	0	0	0	0	0
1	1	1	1	1	0	1	1	1	0	1	0
0	1	0	0	1	0	0	0	1	0	0	0

However, from the data in Table 4.2 when $T = 10$, we can see, after more time cost than the SQP method, the difference between the master problem and sub-problem still can not reach the stop criteria. In fact, when $T = 10$, even spend more than one hour to run the Benders' algorithm to 70 iteration steps, the convergence result can not be changed significantly and not reach the stop criteria. The reasons of this situation are because the optimal value of the sub-problem can not be guaranteed to decrease during the iteration [73], and complexity of master problem is increased two much while add too much cuts on it. Fortunately, from Fig. 4.10 and Fig. 4.9, we can see the tendency of convergence is obvious for the master problem and sub-problem. Thus, we can expect, if provide longer enough time for iteration, or solve those two problems that we mentioned before, Benders' algorithm will be practicable when $T \geq 10$.

As we only implement classic Benders' algorithm to solve the problem, there is no any step to deal with the problems about the convergence of the sub-problem and the numbers of cuts. According [74] [73] and [75], some modified Benders' algorithm could improve the computational effectiveness than the classic one. However, as the time limit, we can not finish all of study about these improved Benders' algorithm. Thus, more researches achievement about this topic could be included in our further papers.

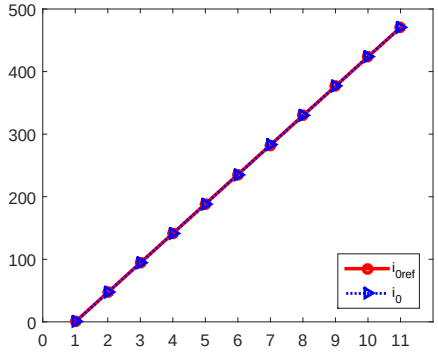


Figure 4.9: Current tracking of Benders' algorithm when $T = 10$.

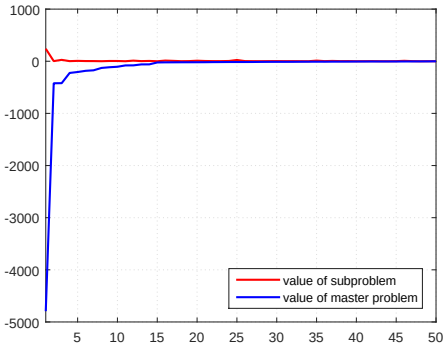


Figure 4.10: Convergence of the objective value for the master problem and sub-problem when $T = 10$.

4.6 Conclusion

In this chapter, a dynamic model of MMC are derived and then formulated to MPC problem. We separate the original MPC problem to a master problem and a sub-problem for implement Benders' decomposition to solve the problem. A $N = 4$ MMC are selected to test proposed algorithm, and

the solutions for different predict horizon are displayed and compared with the solutions by SQP. According the results, Benders' decomposition has great performance on solving the low horizon MPC problem and much better than SQP method. But for longer horizon ($T \geq 10$), even it has presented potential to solve the problem, the algorithm still need more improvement to reduce the computing cost.

Chapter 5: Security Constrained DC OPF Considering Generator Responses

5.1 Introduction

[4]Security constrained OPF(SCOPF) is an extension of OPF considering only pre-contingency or base case constraints. Its purpose is to find an operation point to optimize an objective function at base case, while the post-contingency constraints can all be satisfied. The formulation of SCOPF has two major categories: 1) preventive SCOPF (PSCOPF) [77]; 2) corrective SCOPF (CSCOPF) [29]. In PSCOPF, all control variables are considered without rescheduling except the automatic response of the system.

In conventional formulation of preventive security constrained OPF, contingencies related to generator outages are ignored [32] [33] [34]. Or, same operating conditions are assumed for generators in the pre- and post- contingency [78] [79]. On the other hand, in practical situations, generator outage is common and the system may assume a very different response compared to the related to line outages [80]. With generator outages, online generators have to re-dispatch to compensate the power loss. Their responses are governed by their automatic generator control (AGC) setting.

A generator's post contingency response is illustrated in Fig. 1.2, where $P_g^{\max}$ and $P_g^{\min}$ are the upper and lower limits of the generator active power output respectively; $P_{gi}^{(0)}$ is the generator power in pre-contingency state; $P_g^{(k)}$ is the generator power in post-contingency state; $\Delta^{(k)}$ is the active power imbalance in the system; α is the participation factor corresponding to the slop; Three feasible regions are denoted in the figure.

Modeling the non-convex generator response characteristics has been carried out in the literature. In [81] and [35], the authors assume the generator response in post-contingency always follow their predefined participation factors, i.e. only region (2) is considered. This formulation could efficiently simplify the problem, because the generator responses are formulated as a set of

[4]This Chapter was published in Electric Power Systems Research [76]. Permission is included in the Appendix A.

linear constraints. However, due to the omission of feasible regions (1) and (3), the solutions may be more costly. Alternatively, mixed integer programming (MIP) formulation has been designed in [37] and [38]. The generator response constraint is formulated as a set of MILP constraints based on big-M technique. The major disadvantage if this method is well-known disadvantage related to big-M formulation, i.e., the difficulty of finding a suitable value of M.

Another major challenge to solve SCOPF with generator response constraints is the large size of the problem. Even we only consider the "N-1" criterion, the computation cost of the SCOPF with all contingencies considered could be too high. To address this issue, decomposition techniques have been implemented to reduce the computing cost. For example, alternating direction method of multipliers (ADMM)-based decomposition is adopted in [81], [36], [31] to decompose the original SCOPF problem into one master problem and several sub-problems based on each contingency.

The ADMM methods are heuristic methods to handle non-convexity and the solution from ADMM dose not guarantee global optimum. Hence, this chapter focuses on investigating a more efficient MIP and solving strategy.

The contribution of this part is two fold. 1) A bilinear MIP formulation is designed to model generator post-contingency responses. The proposed model accurately represents the generator response characteristics. 2) An efficient Benders' decomposition is designed to decompose the bilinear MIP problem. Various benders' decomposition strategies were investigated in this research. Through preserving bilinear expressions related to the base case power generation while relaxing the rest bilinear expressions via McCormick envelop, we designed a Benders' decomposition strategy that yields to efficient solving.

5.2 DC-PSCOPF Formulation

The mathematical formulation of security constrained DCOPF is as follows:

$$\min \quad f(x^{(0)}, u^{(0)}) \tag{5.1a}$$

$$s.t. \quad g^{(k)}(x^{(k)}, u^{(k)}) = 0 \qquad k \in \mathcal{C} \tag{5.1b}$$

$$h^{(k)}(x^{(k)}, u^{(k)}) \leq 0 \qquad k \in \mathcal{C} \tag{5.1c}$$

where x refers to DC power flow state variables: bus voltage phase angles at each scenario, u refers the control variables: active power dispatch of each generator at each scenario, $\mathcal{C} = \{(0), (1), \ldots, (N_c)\}$ is the set for the index of scenarios. If $k = 0$, the correspond constraints and variables belong to the base case (pre-contingency).

The decision variables of SCOPF are the base case control actions $u^{(0)}$, state variables $x^{(0)}$ and contingency case state variables $x^{(k)}$.

The objective function is the total operation cost. Denote the cost coefficients as C_{2i}, C_{1i}, C_{0i}, the objective function could be defined as:

$$f_0 = \sum_{i \in \mathcal{G}_{on}} C_{2i}(P_{gi}^{(0)})^2 + C_{1i}P_{gi}^{(0)} + C_{0i}$$

where $P_{gi}^{(0)}$ is the i-th generator's active power dispatch in base case and $\mathcal{G}_{on}$ notates the set of online generators. The equality (g) and inequality (h) are explained as follows.

5.2.1 Equality Constraints: Power Flow Equations

The following constraints are enforced on all bus $i \in \mathcal{N}$ for base case and post-contingency scenarios:

$$P_{gi}^{(0)} - P_{di} = \sum_{(i,j) \in \mathcal{L}} B_{ij}(\theta_i^{(0)} - \theta_j^{(0)}) \qquad i \in \mathcal{N} \tag{5.2}$$

$$P_{gi}^{(k)} - P_{di} = \sum_{(i,j) \in \mathcal{L}_k} B_{ij}(\theta_i^{(k)} - \theta_j^{(k)}) \qquad i \in \mathcal{N} \tag{5.3}$$

where $\mathcal{L}$ and $\mathcal{L}_k$ are the set of transmission lines in base case and post-contingency scenarios; $P_{gi}^{(k)}$ is the generator active power dispatch in post-contingency scenarios; $P_{di}^{(k)}$ is the load at bus i; B_{ij} is the susceptance of the transmission line from bus i to bus j; θ_i is the voltage phase angle of bus i.

5.2.2 Inequality Constraints: Component Limits

Inequality constraints include transmission line limits, generator power limits, and others. Eqs. (5.4) and (5.5) notate transmission line flow limits. Eqs. (5.6) and (5.7) notate generator power

limits. Eqs. (5.8) and (5.9) enforce the reference bus angle to zero.

$$|B_{ij}(\theta_i^{(0)} - \theta_j^{(0)})| \leq F_{ij}^{\max} \qquad (i,j) \in \mathcal{L} \tag{5.4}$$

$$|B_{ij}(\theta_i^{(k)} - \theta_j^{(k)})| \leq F_{ij}^{\max} \qquad (i,j) \in \mathcal{L}_k \tag{5.5}$$

$$P_{gi}^{\min} \leq P_{gi}^{(0)} \leq P_{gi}^{\max} \qquad i \in \mathcal{G} \tag{5.6}$$

$$P_{gi}^{\min} \leq P_{gi}^{(k)} \leq P_{gi}^{\max} \qquad i \in \mathcal{G}_k \tag{5.7}$$

$$\theta_{\text{ref}}^{(0)} = 0 \tag{5.8}$$

$$\theta_{\text{ref}}^{(k)} = 0 \tag{5.9}$$

where $\mathcal{G}_{on}^{(0)}$ and $\mathcal{G}_{on}^{(k)}$ are the sets of generators in base case and post-contingency scenarios; $F_{ij}^{\max}$ is the maximum transmission line capacity; $P_{gi}^{\min}$ and $P_{gi}^{\max}$ are the lower and upper limits of generator active power output; θ_{ref} means the reference bus voltage phase angle.

5.2.3 Generator Post-contingency Response Constraints

When there is a generator outage, the rest of the generators will adjust their outputs based on their participation factors to make up the lost power. If the total load keeps constant, the load variation is zero. Thus, the definition of $\Delta^{(k)}$ has to be carefully designed to consider both load variation contingency and generator outage (while load being constant) contingency.

Hence, $\Delta^{(k)}$ is defined as total generation change for the set of online generators at kth contingency.

$$\Delta^{(k)} = \sum_{i \in \mathcal{G}_{on}} P_{gi}^{(k)} - \sum_{i \in \mathcal{G}_{on}} P_{gi}^{(0)} \tag{5.10}$$

$$P_{gi}^{(k)} = P_{gi}^{(0)} + \alpha_i^{(k)} \Delta^{(k)} \tag{5.11}$$

where $\mathcal{G}_{on} \subseteq \mathcal{G}$ is the set of all online generators at kth contingency, α is the participation factor and

$$\sum_{i \in \mathcal{G}_{on}} \alpha_i^{(k)} = 1 \tag{5.12}$$

Considering the output limit of generators, (5.11) can not always be complied by generators. If $P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)}$ is lower than the minimum limit or greater than the maximum limit, generator should dispatch at its minimum or the maximum at this contingency. The response of the generator can be illustrated in Fig. 1.2.

Moreover, according to [82], these three region can be described as a piecewise function:

$$\text{Region (1)}: \quad P_{gi}^{(k)} = P_{gi}^{\max} \tag{5.13a}$$

$$P_{gi}^{(k)} \leq P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)} \tag{5.13b}$$

$$\text{Region (2)}: \quad P_{gi}^{\min} \leq P_{gi}^{(k)} \leq P_{gi}^{\max} \tag{5.13c}$$

$$P_{gi}^{(k)} = P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)} \tag{5.13d}$$

$$\text{Region (3)}: \quad P_{gi}^{(k)} = P_{gi}^{\min} \tag{5.13e}$$

$$P_{gi}^{(k)} \geq P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)} \tag{5.13f}$$

If all generators comply with the above constraints, the problem may be infeasible. Below is an example. For a system with more than three generators, assume that the post-contingency generation of G1 is in region (1), but all the other generators are in Region (2), i.e. $P_{g1}^{(k)} = P_{g1}^{\max} < P_{g1}^{(0)} + \alpha_1^{(k)}\Delta^{(k)}$, and the other generator follow: $P_{gi}^{(k)} = P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)}$. In this situation, $P_{g1}^{\max} - P_{g1}^{(0)} + \sum_{i\in\mathcal{G}_{on}} \alpha_i^{(k)}\Delta^{(k)} < \Delta^{(k)}$. So that the power imbalance can not be fully compensated. To avoid this situation, we set a reference generator in the system. The generator dose not follow the constraint (5.13).

5.2.4 The MILP Formulation

The piecewise function (5.13) is non-convex. The set of constraints can be formulated via big-M technique into an MILP formulation is briefly introduced. Since there are three regions, two binary variables are introduced: $\omega_{1i}^{(k)}$ (1 for max limit hitting; 0 for otherwise) and $\omega_{2i}^{(k)}$ (1 for min limit hitting; 0 for otherwise).

Hence, region (1) is represented by $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) = (1, 0)$, region (2) is $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) = (0, 0)$ and region (3) is $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) = (0, 1)$.

$$P_{gi}^{\min} \leq P_{gi}^{(k)} \leq P_{gi}^{\max} \tag{5.14a}$$

$$P_{gi}^{\max} - P_{gi}^{(k)} \leq (1 - \omega_{1i}^{(k)})M \tag{5.14b}$$

$$P_{gi}^{(k)} - P_{gi}^{\min} \leq (1 - \omega_{2i}^{(k)})M \tag{5.14c}$$

$$P_{gi}^{(k)} \leq P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)} + \omega_{2i}^{(k)}M \tag{5.14d}$$

$$P_{gi}^{(k)} \geq P_{gi}^{(0)} + \alpha_i^{(k)}\Delta^{(k)} - \omega_{1i}^{(k)}M \tag{5.14e}$$

$$\omega_{1i}^{(k)} + \omega_{2i}^{(k)} \leq 1$$

$$\omega_{1i}^{(k)}, \omega_{2i}^{(k)} \in \{0, 1\}$$

- $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) : (1, 0)$, (5.14) equivalent to (5.13a) and (5.13b);
- $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) : (0, 0)$, (5.14) equivalent to (5.13c) and (5.13d);
- $(\omega_{1i}^{(k)}, \omega_{2i}^{(k)}) : (0, 1)$, (5.14) equivalent to (5.13e) and (5.13f).

M is a large enough constant number. This method is easy to implement, but there is no guaranteed way to find a appropriate value for M. If M is too small, the equivalent relation between (5.13) and (5.14) can not be held. If M is too large, it may cause numerical instability for computation.

Based on the MILP formulation, Benders' decomposition has been proposed in [38].

5.3 The Proposed the Bilinear Formulation

In this section, we will first provide the bilinear formulation which is equivalent (5.13). Benders' decomposition strategies are then examined.

5.3.1 Bilinear Formulation for Generator Response

Based on the definition of ω_{1i} and ω_{2i} in (5.14), the bilinear formulation for constraints (5.13) can be expressed as problem (5.15).

Directly solving the SCOPF problem with constraints (5.15) is difficult since there are bilinear terms consisting of binary variables. Therefore, we seek Benders' decomposition strategies.

$$P_{gi}^{\min} \leq P_{gi}^{(k)} \leq P_{gi}^{\max} \tag{5.15a}$$

$$\omega_{1i}^{(k)}(P_{gi}^{(k)} - P_{gi}^{\max}) \geq 0 \tag{5.15b}$$

$$\omega_{2i}^{(k)}(P_{gi}^{\min} - P_{gi}^{(k)}) \geq 0 \tag{5.15c}$$

$$(1 - \omega_{2i}^{(k)})(P_{gi}^{(k)} - P_{gi}^{(0)} - \alpha_i^{(k)}\Delta^{(k)}) \leq 0 \tag{5.15d}$$

$$(1 - \omega_{1i}^{(k)})(P_{gi}^{(k)} - P_{gi}^{(0)} - \alpha_i^{(k)}\Delta^{(k)}) \geq 0 \tag{5.15e}$$

$$\omega_{1i}^{(k)} + \omega_{2i}^{(k)} \leq 1 \tag{5.15f}$$

$$\omega_{1i}^{(k)}, \omega_{2i}^{(k)} \in \{0, 1\} \tag{5.15g}$$

5.3.2 Benders' Decomposition: Approach 1

The first strategy of Benders' decomposition follows the decomposition structure shown in Fig.5.1. The master problem will decide the base case and contingency scenarios. With the binary variables assigned values, the sub-problems associated with each contingency are linear programming problems with decision variables being continuous variables.

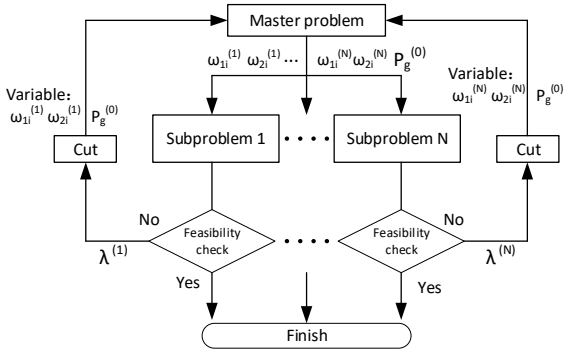


Figure 5.1: Flow chart of Benders decomposition: Approach 1.

Compared with the MILP formulation which can be directly solved by off-shelf solvers, Approach 1 shows obvious worse performance on both of the solution quality and computation efficiency. A brief comparison on the 5-bus system and 39-bus system are shown in Table 5.1.

The disadvantages of Approach 1 appear to be caused by the Benders' cuts associated with both of the binary and continuous variables. Sub-problems generate many infeasible solutions and cuts. So the problem needs many iterations. Moreover, each iteration will add at least one constraint associated to a binary variable to the master problem. The complexity of the master problem could be increased significantly along with iterations.

Table 5.1: Comparison Big-M MILP using Mosek and Approach 1.

		Big-M		Approach 1		
Case	Num_Con	Time	Obj	Time	Obj	Iterations
Case5	1	0.25	21703.48	6.09	21703.48	13
	2	0.27	21703.48	9.11	21786.09	17
Case39	1	0.16	41285.97	5.31	41493.44	10
	2	0.72	41298.97	14.95	41569.21	18

5.3.3 Benders' Decomposition: Approach 2

5.3.3.1 McCormick Envelopes of the Bilinear Formulation

To avoid the issue of Approach 1, we re-design Benders' decomposition to reduce the "coupling" between the master problem and the sub-problems.

Fig. 5.2 presents the new decomposition method: Approach 2. In this new design, the decision variables of the master problem are only associated with the base case. The base case power dispatch $P_{gi}^{(0)}$ will be passed to each sub-problem associated with each contingency. Each sub-problem will decide the power dispatch and guarantee that the post-contingency generator response constraints are satisfied.

Examining the bilinear formulation (5.15), it can be found that the sub-problem for k-th contingency has to deal with four constraints with bilinear components: (5.15b)-(5.15e). Four bilinear expressions exist: $\omega_{1i}^{(k)} P_{gi}^{(k)}, \omega_{2i}^{(k)} P_{gi}^{(k)}, \omega_{1i}^{(k)} \Delta^{(k)}$, and $\omega_{2i}^{(k)} \Delta^{(k)}$.

We implement McCormick envelopes to linearize the bilinear term in constraints so that the sub-problems are MILP problems. According to [83], the linearization based on the McCormick

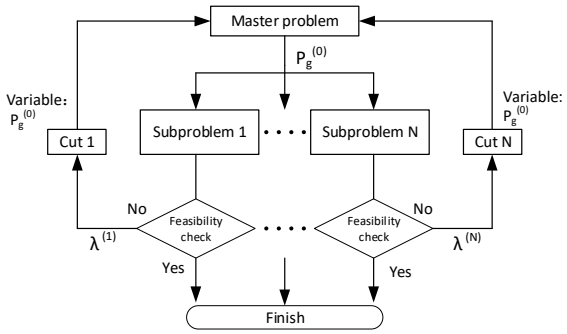


Figure 5.2: Flow chart of Benders decomposition: Approach 2.

envelopes should be tighter than the big-M based linearization. Its efficiency has been validated through implementation on unit commitment and chance constrained problems in [84] and [85].

The process for the bilinear formulation linearization based on the McCormick envelopes is presented as follow. Considering a general bilinear term $\{xy|x^{min} \leq x \leq x^{max}, y^{min} \leq y \leq y^{max}\}$, its McCormick envelops is:

$$z = xy$$

$$z \geq x^{\min}y + xy^{\min} - x^{\min}y^{\min} \tag{5.16a}$$

$$z \geq x^{\max}y + xy^{\max} - x^{\max}y^{\max} \tag{5.16b}$$

$$z \leq xy^{\max} + x^{\min}y - x^{\min}y^{\max} \tag{5.16c}$$

$$z \leq x^{\max}y + xy^{\min} - x^{\max}y^{\min} \tag{5.16d}$$

Therefore, to linearize (5.15), e first define:

$$\overline{P}_{gi}^{(k)} = \omega_{1i}^{(k)} P_{gi}^{(k)} \tag{5.17}$$

$$\underline{P}_{gi}^{(k)} = \omega_{2i}^{(k)} P_{gi}^{(k)} \tag{5.18}$$

$$\overline{\Delta}^{(k)} = \omega_{1i}^{(k)} \Delta^{(k)} \tag{5.19}$$

$$\underline{\Delta}^{(k)} = \omega_{2i}^{(k)} \Delta^{(k)} \tag{5.20}$$

The upper and lower bound of $P_{gi}^{(k)}$ are P_{gi}^{max} and P_{gi}^{min} which are given by (5.15a). $\omega_{1i}^{(k)}$ and $\omega_{2i}^{(k)}$ are binary variables which means their upper and lower limit is 1 and 0 respectively. For $\Delta^{(k)}$, we have $\Delta_{max}^{(k)} = P_{gk}^{max}$, $\Delta_{min} = P_{gk}^{min}$ for generator outage scenarios, where P_{gk}^{max} and P_{gk}^{min} are the output limits of the failure generator. Assuming $P_{gi}^{min} = 0$ in general, the McCormick envelops for constraint (5.17)-(5.20) can be expressed as follows:

$$\begin{cases} 0 \le \overline{P}_{gi}^{(k)} \le \omega_{1i}^{(k)} P_{gi}^{\max} \\ P_{gi}^{(k)} - (1-\omega_{1i}^{(k)}) P_{gi}^{\max} \le \overline{P}_{gi}^{(k)} \le P_{gi}^{(k)} \end{cases} \tag{5.21}$$

$$\begin{cases} 0 \le \underline{P}_{gi}^{(k)} \le \omega_{2i}^{(k)} P_{gi}^{\max} \\ P_{gi}^{(k)} - (1-\omega_{2i}^{(k)}) P_{gi}^{\max} \le \underline{P}_{gi}^{(k)} \le P_{gi}^{(k)} \end{cases} \tag{5.22}$$

$$\begin{cases} 0 \le \overline{\Delta}^{(k)} \le \omega_{1i}^{(k)} \Delta_{\max}^{(k)} \\ \Delta^{(k)} - (1-\omega_{1i}^{(k)}) \Delta_{\max}^{(k)} \le \overline{\Delta}^{(k)} \le \Delta^{(k)} \end{cases} \tag{5.23}$$

$$\begin{cases} 0 \le \underline{\Delta}^{(k)} \le \omega_{2i}^{(k)} \Delta_{\max}^{(k)} \\ \Delta^{(k)} - (1-\omega_{2i}^{(k)}) \Delta_{\max}^{(k)} \le \underline{\Delta}^{(k)} \le \Delta^{(k)} \end{cases} \tag{5.24}$$

Thus, equations (5.15b) to (5.15e) can be expressed as:

$$\omega_{1i}^{(k)} P_{gi}^{\max} - \overline{P}_{gi}^{(k)} \le 0 \tag{5.25}$$

$$\underline{P}_{gi}^{(k)} - \omega_{2i}^{(k)} P_{gi}^{\max} \le 0 \tag{5.26}$$

$$(P_{gi}^{(k)} - \underline{P}_{gi}^{(k)}) - (1-\omega_{2i}^{(k)}) P_{gi}^{(0)} - \alpha_i^{(k)} (\Delta^{(k)} - \underline{\Delta}^{(k)}) \le 0 \tag{5.27}$$

$$(1-\omega_{1i}^{(k)}) P_{gi}^{(0)} + \alpha_i^{(k)} (\Delta^{(k)} - \overline{\Delta}^{(k)}) - (P_{gi}^{(k)} - \overline{P}_{gi}^{(k)}) \le 0 \tag{5.28}$$

$$(5.21) \sim (5.24)$$

In the above formulation, we do not linearize terms $\omega_{1i}^{(k)} P_{gi}^{(0)}$ and $\omega_{2i}^{(k)} P_{gi}^{(0)}$. The reason is that $P_{gi}^{(0)}$ will be determined by the master problem. In sub-problems, $P_{gi}^{(0)}$ is treated as a constant.

5.3.3.2 Benders' Decomposition

Based on aforementioned definition, the sub-problem could be formulated as:

$$v^{(k)} := \min \quad ||[s_{1i}^{(k)}, s_{2i}^{(k)}, s_{3}^{(k)}]|| \tag{5.29a}$$

$s.t.$ Power flow constraints: (5.3), (5.5), (5.3), (5.9)

Generator limits: (5.15a)

McCormick envelop constraints: (5.21) $\sim$ (5.26)

$$(P_{gi}^{(k)} - \underline{P}_{gi}^{(k)}) - (1 - \omega_{2i}^{(k)})\hat{P}_{gi}^{(0)} - \alpha_i(\Delta^{(k)} - \underline{\Delta}^{(k)}) \leq s_{1i}^{(k)} \tag{5.29b}$$

$$(1 - \omega_{1i}^{(k)})\hat{P}_{gi}^{(0)} + \alpha_i(\Delta^{(k)} - \overline{\Delta}^{(k)}) - (P_{gi}^{(k)} - \overline{P}_{gi}^{(k)}) \leq s_{2i}^{(k)} \tag{5.29c}$$

$$\sum_{i \in \mathcal{G}_{on}} P_{gi}^{(k)} - \sum_{i \in \mathcal{G}_{on}} \hat{P}_{gi}^{(0)} - \Delta^{(k)} = s_{3}^{(k)} \tag{5.29d}$$

$$s_{1i}^{(k)}, s_{2i}^{(k)} \geq 0 \tag{5.29e}$$

Binary variable constraints: (5.15f), (5.15g)

where $\hat{P}_{gi}^{(0)}$ is a known value passed from the master problem, $s_{1i}^{(k)}$, $s_{2i}^{(k)}$, $s_{3}^{(k)}$ are slack variables to relax region (2) constraints defined by (5.27), (5.28) as well as the Δ^k definition constraints.

$v^{(k)}$ is the value of the objective function (5.29a). Obviously when $v^{(k)} = 0$, the solution of the master problem $\hat{P}_{gi}^{(0)}$ is feasible for the contingency cases. In the computing process, we consider $\hat{P}_{gi}^{(0)}$ is feasible for the contingency cases if $v^{(k)} \leq \epsilon$, where ϵ is a predefined small constant value.

To generator Benders' cuts, we need to know the dual variable of constraints (5.29b), (5.29c) and (5.29d). However, because there are binary variables included in (5.29b) and (5.29c) , we can not directly obtain dual variable from the sub-problem (5.29).

To Solve this issue, we implement a technique introduced in [86]. The process of this technique is described as follows.

- First, we solve the sub-problem (5.29) to obtain the value of $\omega_{1i}^{(k)}$, $\omega_{2i}^{(k)}$, denotes these value as $\hat{\omega}_{1i}^{(k)}$ and $\hat{\omega}_{2i}^{(k)}$.
- Next, replace $\omega_{1i}^{(k)}$ and $\omega_{2i}^{(k)}$ by $\hat{\omega}_{1i}^{(k)}$ and $\hat{\omega}_{2i}^{(k)}$ in (5.15), and solve the problem.

The new formulation of the sub-problem can be expressed as follows.

$$v^{(k)} := \min \quad ||[s_{1i}^{(k)}, s_{2i}^{(k)}, s_3^{(k)}]|| \tag{5.30a}$$

$$s.t. \quad \text{Power flow constraints: (5.3), (5.5), (5.3), (5.9)}$$

$$\text{Generator limits: (5.15a)} \tag{5.30b}$$

$$\text{McCormick envelop constraints: (5.21)} \sim \text{(5.26)} \tag{5.30c}$$

$$\lambda_{1i}^{(k)} : (P_{gi}^{(k)} - \underline{P}_{gi}^{(k)}) - (1 - \hat{\omega}_{2i}^{(k)})\hat{P}_{gi}^{(0)} - \alpha_i(\Delta^{(k)} - \underline{\Delta}^{(k)}) \leq s_{1i}^{(k)} \tag{5.30d}$$

$$\lambda_{2i}^{(k)} : (1 - \hat{\omega}_{1i}^{(k)})\hat{P}_{gi}^{(0)} + \alpha_i(\Delta^{(k)} - \overline{\Delta}^{(k)}) - (P_{gi}^{(k)} - \overline{P}_{gi}^{(k)}) \leq s_{2i}^{(k)} \tag{5.30e}$$

$$\lambda_3^{(k)} : \sum_{i \in \mathcal{G}_{on}} P_{gi}^{(k)} - \sum_{i \in \mathcal{G}_{on}} \hat{P}_{gi}^{(0)} - \Delta P^{(k)} = s_3^{(k)} \tag{5.30f}$$

$$s_{1i}^{(k)}, s_{2i}^{(k)} \geq 0 \tag{5.30g}$$

Based on the solution of (5.30), the feasibility cut is shown as follows:

$$\begin{aligned} 0 \geq & \hat{v}^{(k)} - \hat{\lambda}_3^{(k)} \sum_{i \in \mathcal{G}} (P_{gi}^{(0)} - \hat{P}_{gi}^{(0)}) \\ & + \sum_{i \in \mathcal{G}^-} [(1 - \hat{\omega}_{1i}^{(k)})\hat{\lambda}_{2i}^{(k)} - (1 - \hat{\omega}_{2i}^{(k)})\hat{\lambda}_{1i}](P_{gi}^{(0)} - \hat{P}_{gi}^{(0)}) \end{aligned} \tag{5.31}$$

where $\mathcal{G}^-$ is the set of online generators excluding the reference bus, $\lambda_{1i}^{(k)}$, $\lambda_{2i}^{(k)}$, $\lambda_3^{(k)}$ are the dual variables of constraints (5.30d), (5.30e), (5.30f) respectively. Symbol $(\hat{\cdot})$ means the correspond parameter is a fixed value.

In the master problem, we include all variable and constraints for the OPF problem of base case and the Benders' cuts generated by all sub-problems.

$$\min \quad C(P_{gi}^{(0)}) \tag{5.32}$$

$$s.t. \quad P_{gi}^{(0)} - P_{di}^{(0)} = \sum_{(i,j)\in\mathcal{L}} B_{ij}(\theta_i^{(0)} - \theta_j^{(0)}) \tag{5.33}$$

$$|B_{ij}(\theta_i^{(0)} - \theta_j^{(0)})| \leq F_{ij}^{\max} \tag{5.34}$$

$$P_{gi}^{\min} \leq P_{gi}^{(0)} \leq P_{gi}^{\max} \tag{5.35}$$

$$\theta_{ref}^{(0)} = 0 \tag{5.36}$$

Benders Cut:(5.31)

The solution $P_{gi}^{(0)}$ from the master problem is denoted as $\hat{P}_{gi}^{(0)}$ and sent to sub-problems. The sub-problems further create feasibility cuts for the master problem until the sub-problems no longer generators cuts.

5.4 Case Studies

In this section, we present the case study results. Our numerical experiments are conducted on an Inter(R) Core(TM) i5-8250U CPU @ 1.60 GHZ computer. All presented methods are implemented on Matlab 2019b using CVX and applying Mosek 9.0 as the main solver. To avoid too long time solving for MIP problem, we set the relative optimality tolerance of the Mosek integer optimizer as 1×10^{-3}(default:1×10^{-4}), and the maximum solving time of Mosek as 2000 seconds. All the other configurations are default. The feasibility check tolerance is $\epsilon = 10^{-3}$ for all Benders' methods. For all big-M based methods, M is fixed as 1000.

For convenience, the big-M method based on the formulation (5.14) is notated as "Method 1" in Tables and Figures; the proposed bilinear formulation and Benders' decomposition Approach 2 is notated as "Method 2".

As the proposed method is based on the parallel computation, its computation time is calculated according to the following equations: As the proposed method is based on the parallel computation,

its computation time is calculated according to the following equations:

$$T = \sum_{i \in l} \max_{j \in c}(t_i^j)$$

where, T means the total computation time, c is the set of contingency index, l is the set of iteration sequence, t_i^j means the time consuming of the j^{th} contingency at the i^{th} iteration.

For an SCOPF problem, some conflicting contingencies may exist. Those contingencies can not co-exist in the contingencies sets of SCOPF problems [87]. In this chapter, we identified he removed all conflicting contingencies in the test cases. The identified conflicting contingencies are list in Table 5.2.

Table 5.2: Conflict contingencies in all cases.

Cases	Conflict contingencies
Case3	None
Case5	G_3
Case14	None
case39	$G_{31}, G_{32}, G_{33}, G_{34}, G_{35}, G_{36}, L_{1-2}$ $L_{1-39}, L_{2-30}, L_{4-14}, L_{6-11}, L_{6-31}, L_{10-11}$ $L_{10-13}, L_{10-32}, L_{13-14}, L_{16-19}, L_{17-18}, L_{19-20}$ $L_{19-33}, L_{20-34}, L_{21-22}, L_{22-35}, L_{23-36}, L_{25-26}$ $L_{25-37}, L_{26-27}, L_{29-38}$
case118	$L_{85-86}, L_{110-111}, L_{68-116}, L_{12-117}$

5.4.1 Three-bus System

In this subsection, we show that the proposed method (Method 2) is capable of correctly reflecting the generator response based on its post-contingency response constraints (5.13). The testbed system is a three-bus three-machine system. The topology of the system is presented in Fig. 5.31. The parameters of the branches and generators are listed in Table 5.3

Table 5.3: 3-bus system parameters.

	C_2	C_1	C_0	$P_{\max}$	$P_{\min}$	α
G_1	0.11	5	150	3000	0	1/30
G_2	0.085	1.2	100	300	0	1/3
G_3	0.055	1	50	400	0	19/30
$B_{1,2,3}$	$X = 0.0504(p.u)$			$F_{\max} = 3(p.u)$		

In the first experiment, load varying is treated as contingency scenarios. Outage is not considered. Total 7 contingency scenarios are considered in the SCOPF problem. Fig.5.4 presents the

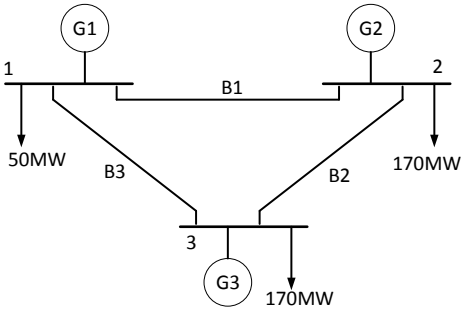


Figure 5.3: 3 bus system topology

generator post-contingency responses. The x-axis is the load change against the base case load level. The y-axis notates the variation in generator power against the base case generation. It can be seen that the response of G2 and G3 comply with the feasible region defined by participation factor, low and upper bounds. On the other hand, G1 is the reference generator. When either one of the two generators hit the limit, G1 will no longer assume the designed participation factor of 1/30.

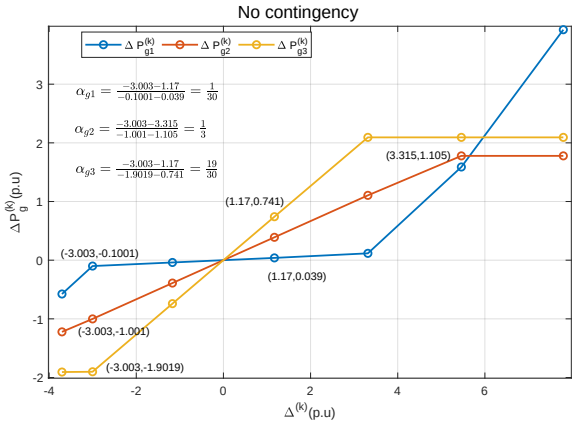


Figure 5.4: Generators response for load variation. G2 and G3 comply with the feasible regions defined by the participation factor, low and upper bounds.

In the second experiment, N-1 contingencies are considered. There are 6 contingencies (3 generator outages, and 3 branch outages). We assume that the reference generator (G1) never fails in ay scenario, so only 5 contingencies will be considered in this example.

This SCOPF is solved by the proposed method (Method 2) and the off-shelf solver (Mosek) via big-M formulation (Method 1). Comparison is made on the solution and the computing time. The

solution results are listed in Table 5.5. The results show the objective value for two methods are the same. The computation time of the big-M method is slightly less than the proposed method. This is due to the small-size of the system. The advantage of the proposed method on computation efficiency will be demonstrated for larger-scale power grids. This point will be validated using an 118-bus system.

Table 5.4: Comparison of objective values for the Big-M method and the proposed method.

	Load Variation	N-1 Contingency
Method 1 obj	4946.17	4946.17
Method 2 obj	4946.17	4946.17

Table 5.5 lists the solutions for the second SCOPF problem (the second experiment). In all branch outage scenarios, the generator power outputs in post-contingency are the same as those in the base case. In a contingency scenario with G2 outage, the set of the online generators included G1 and G3. Due to the loss of G2, the change of total is 1.2219 p.u. G1 and G3 now share the total power generation change according to their participation factors: $1/30 : 19/30 = 5\% : 95\%$. Hence G1 is expected to share 5% ($= \frac{0.8326-0.7715}{1.2219}$) of the total power change, while G3 is expected to share 95% ($= \frac{3.0647-0.7715}{1.2219}$). The solution matches the expected results.

For G3 outage, $\Delta^{(k)} = 1.9066$ p.u. G1 and G2 share the power change according to their participation factors: $1/30 : 1/3 = 1 : 10$. The solution shows that the power increase of G1 is $0.9448 - 0.7715 = 0.1733$ and the power increase of G2 is $2.9552 - 1.2219 = 1.7333$. Hence, the generators respond as expected for post contingency.

Table 5.5: Solution for the N-1 contingency SCOPF

	Base case	Gen 2 outage	Gen 3 outage	All branch outage
P_{g1}	0.7715	0.8326	0.9448	0.7715
P_{g2}	1.2219	–	2.9552	1.2219
P_{g3}	1.9066	3.0674	–	1.9066

5.4.2 IEEE 118-bus System

Data of the IEEE 118-bus system are from MATPOWER [56]. The line flow limit is set to 500 MVA. In all tests, 100% load means the total load is 42.42 pu. This system consists of 186 branches and 54 generators. Thus, based on "N-1" criterion, there could be 240 contingencies. Part of the

contingencies are selected for SCOPF. The information of the selected contingencies is listed in Table 5.6. Eight critical contingencies, identified by [79] and [88], are included in all situations. The rest are randomly selected.

Table 5.6: 118 bus contingency information.

Number of selected contingency	Number of Branch contingency	Number of Generator contingency
8	4	4
20	10	10
30	15	15
40	20	20
50	25	25
Critical contingencies		
$L_{26-30}, L_{34-37}, L_{38-37}, L_{70-71}$ $G_{65}, G_{66}, G_{80}, G_{89}$		

Fig. 5.5 presents the lower bound computed by the master problem and maximum of the solutions from sub-problems associate with contingent scenarios, along with the iteration. It can be seen that the lower bound converges when all sub-problems become feasible. The tested example is the 118 bus system with 190% load.

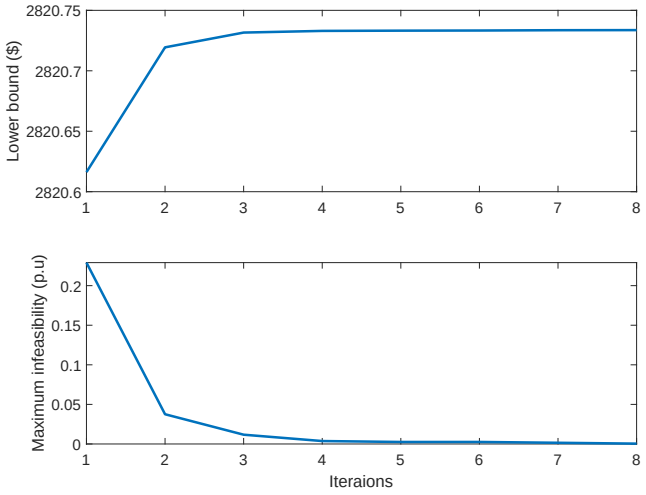


Figure 5.5: Lower bound computed from the master problem converges while an maximum of the sub-problem solutions converges to zero. The test case is the IEEE 118-bus with 190% load.

The computation time of two methods are presented in Figs. 5.6 and 5.7. Fig. 5.6 presents computing time versus loading level for three SCOPF problems which include 8, 30, 50 contingencies

respectively. It can be found that for majority of the loading levels, Method 2's computing time is two orders less than Method 1's. Fig. 5.7 presents computing versus the number of contingencies. It can be found that while for Method 1, computing time increases linearly as the number of contingency increases, computing time for Method 2 is always less than 10 seconds.

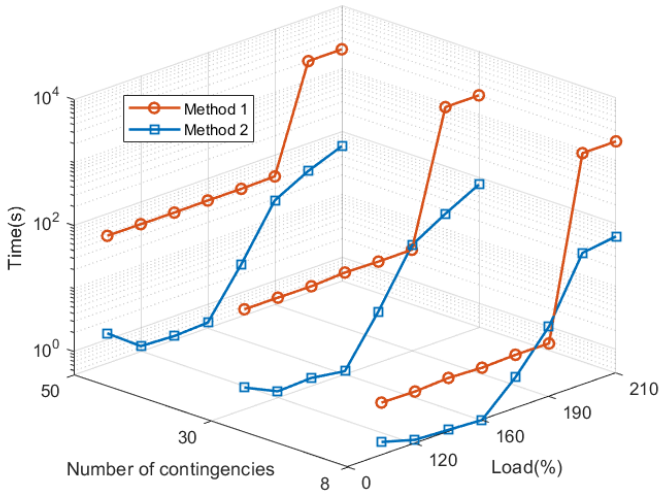


Figure 5.6: Computation time against load level.

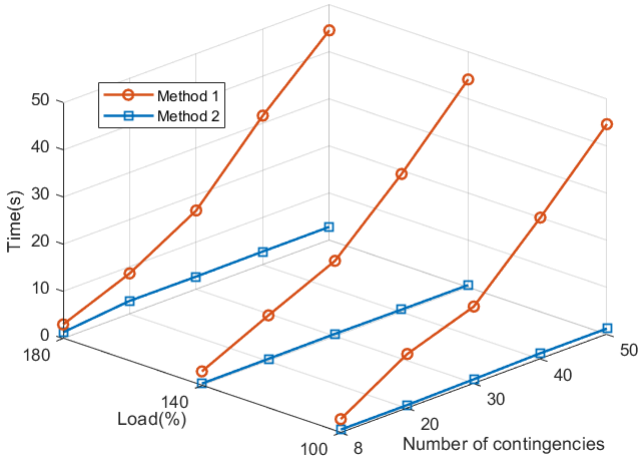


Figure 5.7: Computation time against contingency number.

5.4.3 Other Instances

As a total, five test systems have been examined. Case3 and Case118 have been discussed in the previous subsections. The rest three instances, Case 5, Case14, and Case39, are from the

MATPOWER case library. The total load of Case3 is set as 3.51 p.u. For Case14, line flow limit 100 MW is imposed. For each instance, the participation factors of generators are defined as a set of fixed positive values with their sum as 1. Table 5.7 presents the information of the three test cases solved by the two methods. In Table 5.7, the NAN result means that the correspond methods failed to find reasonable solution within time limit.

For Case3 and Case5, the two methods results the same solution. For Case14, Case39, Case118, as the problem dimension increases, Method 2 becomes more computing efficient compared to Method 1. The solution of Case39 is NAN. This means that the big-M formulation and Mosek solver can not find a reasonable solution within the time limits of 2000 seconds.

Table 5.7: Comparison in different methods

Case	# of contingencies	# of binary	Method	Obj	Time	Iteration
Case3	5	16	Method 1	4128.20	0.14	–
			Method 2	4128.20	0.55	1
Case5	9	66	Method 1	22869.60	1.59	–
			Method 2	22869.60	2.56	5
Case14	24	184	Method 1	8459.71	34.36	–
			Method 2	8459.71	1.98	3
Case39	27	480	Method 1	NAN	2000	–
			Method 2	41945.24	5.72	5
Case118	235	24804	Method 1	130508.80	1143.39	–
			Method 2	130508.70	3.89	3

5.5 Conclusion

This chapter proposes a bilinear mixed integer programming formulation and an efficient solving approach for security constrained DCOPF to consider generators' post-contingency responses. Through preserving bilinear expressions related to the base case power generation while relaxing the rest bilinear expressions via McCormick envelops, we designed a Benders' decomposition strategy that yields to efficient solving. Case study results demonstrate the efficiency of the proposed formulation compared to the state-of-the-art formulations.

Chapter 6: Conclusion and Future Plan

6.1 Conclusion

This dissertation conducted researches about the application of the power system optimization. It focuses on three major problems: ACOPF, MPC, and SCOPF. For the research about the ACOPF, we developed an convex relaxation based ACOPF solver which is capable to solve ACOPF for large size power system. To deal with the exactness issue of the convex relaxation formulation of ACOPF, two algorithm has been studied. For MPC problem, a Benders' decomposition based computational algorithm is implemented on the MMC dynamic model based MPC problem. For the SCOPF, a new method to solve SCOPF with generator response constraints is investigated. The benefits of this dissertation research are summarized as follows:

6.1.1 A Sparse Convex ACOPF Solver Based on 3-node Cycles

In the Chapter 2, we proposed a 3-node cycle decomposition based sparse convex relaxation for ACOPF. We have shown that the 3-node cycle decomposition can not guarantee that the resulting graph is a chordal graph. However, the proposed relaxation can achieve the close tightness as SDP OPF solvers. On the other hand, our method has a clearly higher computing efficiency.

6.1.2 Exactness of the Convex Relaxation

In the Chapter 3, an efficient convex iteration implementation is also investigated for the proposed sparse convex solver to achieve exactness or rank-1 solutions. Our experiment results show the feasibility of the implementation. Moreover, we proposed a nonlinear programming formulation for ACOPF. This formulation is based on decision variables that align with SOCP/SDP relaxation. The proposed formulation exploits power network sparsity feature and employs a small set of minor constrains related to all 3-node cycles as equality constraints to enforce rank-1 constraint, so the solution is exact. Case study results demonstrate the correctness of this formulation.

6.1.3 Benders' Decomposition for MPC of a Modular Multi-level Converter

In this chapter, a dynamic model of MMC are derived and then formulated to MPC problem. We separate the original MPC problem to a master problem and a sub-problem for implement Benders' decomposition to solve the problem. A $N = 4$ MMC are selected to test proposed algorithm, and the solutions for different predict horizon are displayed and compared with the solutions by SQP. According the results, Benders' decomposition has great performance on solving the low horizon MPC problem and much better than SQP method. But for longer horizon ($T \geq 10$), even it has presented potential to solve the problem, the algorithm still need more improvement to reduce the computing cost.

6.1.4 Security Constrained DC OPF Considering Generator Responses

The Chapter 4 proposes a bilinear mixed integer programming formulation and an efficient solving approach for security constrained DCOPF to consider generators' post-contingency responses. Through preserving bilinear expressions related to the base case power generation while relaxing the rest bilinear expressions via McCormick envelops, we designed a Benders' decomposition strategy that yields to efficient solving. Case study results demonstrate the efficiency of the proposed formulation compared to the state-of-the-art formulations.

6.2 Future Work

6.2.1 Security Constrained ACOPF

The new formulation to handle the generator response constraints in preventive DC SCOPF can be implemented on ACOPF. The Chapter 4 verified the efficiency of the new Benders' decomposition based bilinear formulation to solve the generator response integrated preventive DC SCOPF. Because of the formulation is only related with power, it should be easy to be implemented on the AC SCOPF. The convex relaxation can be applied to deal with the AC power flow constraints.

6.2.2 OPF in Renewable Energy Source Integrated Power System

In recent years, as the increasing of the renewable energy sources, the uncertainty of the renewable energy sources becomes an big challenge for the traditional OPF mechanism. Some new algorithm, such as the multi-stage stochastic programming, worst case optimization, and chance constrained optimization are implemented to deal with this issue. However, the complexity of the

OPF has been significantly increased, because the new algorithms need to add additional variables and constraints. Therefore, the method to improve the calculation speed still needs more investigation.

References

[1] M. E. Initiative, "The future of the electric grid," 2011. [Online]. Available: https://sites.suffolk.edu/mminobe93/2015/09/20/u-s-energy-grid/

[2] R. Bansal, "Optimization methods for electric power systems: An overview," *International Journal of Emerging Electric Power Systems*, vol. 2, no. 1, 2005.

[3] L. Li, *Convex Relaxation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 115–126. [Online]. Available: https://doi.org/10.1007/978-3-662-46356-7_6

[4] A. Castillo and R. P. O'Neill, "Survey of approaches to solving the acopf (opf paper 4)," *US Federal Energy Regulatory Commission, Tech. Rep*, 2013.

[5] W. A. Bukhsh, A. Grothey, K. I. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4780–4788, 2013.

[6] S. H. Low, "Convex relaxation of optimal power flow—part i: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014.

[7] J. BnnoBRs, "Partitioning procedures for solving mixed-variables programming problems ‘," 1962.

[8] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.

[9] A. M. Geoffrion and G. W. Graves, "Multicommodity distribution system design by benders decomposition," *Management science*, vol. 20, no. 5, pp. 822–844, 1974.

[10] J. Carpentier, "Contribution to the economic dispatch problem," *Bulletin de la Societe Francoise des Electriciens*, vol. 3, no. 8, pp. 431–447, 1962.

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

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

[13] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 6, pp. 383–392, 2008.

[14] R. A. Jabr, "Radial distribution load flow using conic programming," *IEEE transactions on power systems*, vol. 21, no. 3, pp. 1458–1459, 2006.

[15] M. B. Cain, R. P. O'neill, and A. Castillo, "History of optimal power flow and formulations," *Federal Energy Regulatory Commission*, pp. 1–36, 2012.

[16] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, 2012.

[17] B. Kocuk, S. S. Dey, and X. A. Sun, "Inexactness of sdp relaxation and valid inequalities for optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 642–651, 2016.

[18] B. C. Lesieutre, D. K. Molzahn, A. R. Borden, and C. L. DeMarco, "Examining the limits of the application of semidefinite programming to power flow problems," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE, 2011, pp. 1492–1499.

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

[20] D. K. Molzahn, C. Josz, I. A. Hiskens, and P. Panciatici, "A laplacian-based approach for finding near globally optimal solutions to opf problems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 305–315, 2017.

[21] S. You and Q. Peng, "A non-convex alternating direction method of multipliers heuristic for optimal power flow," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 788–793.

[22] B. Kocuk, S. S. Dey, and X. A. Sun, "Matrix minor reformulation and socp-based spatial branch-and-cut method for the ac optimal power flow problem," *arXiv preprint arXiv:1703.03050*, 2017.

[23] W. Wang and N. Yu, "Chordal conversion based convex iteration algorithm for three-phase optimal power flow problems," *IEEE Transactions on Power Systems*, 2017.

[24] W. Wei, J. Wang, N. Li, and S. Mei, "Optimal power flow of radial networks and its variations: A sequential convex optimization approach," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2974–2987, 2017.

[25] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4312–4324, Dec 2008.

[26] Y. Ma, L. Fan, and Z. Miao, "Realizing space vector modulation in matlab/simulink and pscad," in *2013 North American Power Symposium (NAPS)*. IEEE, 2013, pp. 1–6.

[27] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Transactions on industrial electronics*, vol. 56, no. 6, pp. 1826–1838, 2008.

[28] L. Platbrood, F. Capitanescu, C. Merckx, H. Crisciu, and L. Wehenkel, "A generic approach for solving nonlinear-discrete security-constrained optimal power flow problems in large-scale systems," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1194–1203, 2014.

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

[30] B. Stott and O. Alsaç, "Optimal power flow: Basic requirements for real-life problems and their solutions," in *SEPOPE XII Symposium, Rio de Janeiro, Brazil*, vol. 11, 2012.

[31] D. Phan and J. Kalagnanam, "Some efficient optimization methods for solving the security-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 863–872, 2013.

[32] T. Sennewald, F. Sass, and D. Westermann, "A preventive security constrained optimal power flow for mixed ac-hvdc-systems," 2017.

[33] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, "Applications of security-constrained optimal power flows," in *In Proceedings of Modern Electric Power Systems Symposium, MEPS06*, 2006.

[34] V. Hinojosa and F. Gonzalez-Longatt, "Preventive security-constrained dcopf formulation using power transmission distribution factors and line outage distribution factors," *Energies*, vol. 11, no. 6, p. 1497, 2018.

[35] K. Sundar, H. Nagarajan, L. Roald, S. Misra, R. Bent, and D. Bienstock, "Chance-constrained unit commitment with n-1 security and wind uncertainty," *IEEE Transactions on Control of Network Systems*, 2019.

[36] M. Velay, M. Vinyals, Y. Besanger, and N. Retiere, "Fully distributed security constrained optimal power flow with primary frequency control," *International Journal of Electrical Power & Energy Systems*, vol. 110, pp. 536–547, 2019.

[37] M. Carrión, Y. Dvorkin, and H. Pandžić, "Primary frequency response in capacity expansion with energy storage," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1824–1835, 2017.

[38] Y. Dvorkin, P. Henneaux, D. S. Kirschen, and H. Pandžić, "Optimizing primary response in preventive security-constrained optimal power flow," *IEEE Systems Journal*, vol. 12, no. 1, pp. 414–423, 2016.

[39] M. Ma, L. Fan, Z. Miao, B. Zeng, and H. Ghassempour, "A sparse convex ac opf solver and convex iteration implementation based on 3-node cycles," *Electric Power Systems Research*, vol. 180, p. 106169, 2020.

[40] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Transactions on power apparatus and systems*, no. 10, pp. 1866–1876, 1968.

[41] F. Milano, C. A. Cañizares, and A. J. Conejo, "Sensitivity-based security-constrained opf market clearing model," *IEEE Transactions on power systems*, vol. 20, no. 4, pp. 2051–2060, 2005.

[42] R. Zárate-Miñano, F. Milano, and A. J. Conejo, "An opf methodology to ensure small-signal stability," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1050–1061, 2011.

[43] S. H. Low, "Convex relaxation of optimal power flow—part ii: Exactness," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 177–189, 2014.

[44] ——, "Convex relaxation of optimal power flow part i: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014.

[45] R. A. Jabr, "Exploiting sparsity in SDP relaxations of the OPF problem," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1138–1139, 2012.

[46] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, "Implementation of a large-scale optimal power flow solver based on semidefinite programming," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3987–3998, 2013.

[47] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion i: General framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, 2001.

[48] R. Madani, M. Ashraphijuo, and J. Lavaei. (2014) SDP Solver of Optimal Power Flow User's Manual. [Online]. Available: http://ieor.berkeley.edu/~lavaei/Software.html

[49] B. Kocuk, S. S. Dey, and X. A. Sun, "Strong socp relaxations for the optimal power flow problem," *Operations Research*, vol. 64, no. 6, pp. 1177–1196, 2016.

[50] Z. Miao, L. Fan, H. G. Aghamolki, and B. Zeng, "Least Squares Estimation Based SDP Cuts for SOCP Relaxation of AC OPF," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 241–248, Jan 2018.

[51] C. Bingane, M. F. Anjos, and S. Le Digabel, "Tight-and-cheap conic relaxation for the ac optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7181–7188, 2018.

[52] H. Johnston, "Cliques of a graph-variations on the Bron-Kerbosch algorithm," *International Journal of Parallel Programming*, vol. 5, no. 3, pp. 209–238, 1976.

[53] J. Wildman, "Bron-Kerbosch maximal clique finding algorithm," https://www.mathworks.com/matlabcentral/fileexchange/30413-bron-kerbosch-maximal-clique-finding-algorithm, 2011.

[54] K. Mehlhorn and D. Michail, "Implementing minimum cycle basis algorithms," *Journal of Experimental Algorithmics (JEA)*, vol. 11, pp. 2–5, 2007.

[55] S. Iglin, "grTheory - Graph Theory Toolbox," https://www.mathworks.com/matlabcentral/fileexchange/4266-grtheory-graph-theory-toolbox?focused=5177310&tab=function, 2011.

[56] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2011.

[57] C. Coffrin, D. Gordon, and P. Scott. (2014) NESTA, the NICTA energy system test case archive. [Online]. Available: http://arxiv.org/abs/1411.0359

[58] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008.

[59] M. Ma and L. Fan, "Rank-1 positive semidefinite matrix-based nonlinear programming formulation for ac opf," *International Transactions on Electrical Energy Systems*, vol. 29, no. 10, p. e12095, 2019.

[60] C. Chen, A. Atamtürk, and S. S. Oren, "A spatial branch-and-cut method for nonconvex qcqp with bounded complex variables," *Mathematical Programming*, vol. 165, no. 2, pp. 549–577, 2017.

[61] W. Wang and N. Yu, "Chordal conversion based convex iteration algorithm for three-phase optimal power flow problems," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1603–1613, 2018.

[62] Y. Shi, H. Tuan, P. Apkarian, and A. Savkin, "Global optimal power flow over large-scale power transmission networks," *Systems & Control Letters*, vol. 118, pp. 16–21, 2018.

[63] S. Bose, S. H. Low, T. Teeraratkul, and B. Hassibi, "Equivalent relaxations of optimal power flow," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 729–742, 2015.

[64] J. Dattorro, *Convex optimization & Euclidean distance geometry.* Lulu. com, 2010.

[65] M. Ma, L. Fan, Z. Miao, B. Zeng, and H. Ghassempour, "Strengthened socp relaxation for ac opf and rank-1 computing based on 3 -node cycles," *under review, IEEE Transactions on Power Systems*, 2018.

[66] M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

[67] M. Ma and L. Fan, "Bender's decomposition algorithm for model predictive control of a modular multi-level converter," in *2017 North American Power Symposium (NAPS)*. IEEE, 2017, pp. 1–6.

[68] L. Franquelo, J. Rodriguez, J. Leon, S. Kouro, R. Portillo, and M. Prats, "The age of multilevel converters arrives," *Industrial Electronics Magazine, IEEE*, vol. 2, no. 2, pp. 28–39, June 2008.

[69] M. Saeedifard and R. Iravani, "Dynamic performance of a modular multilevel back-to-back hvdc system," *Power Delivery, IEEE Transactions on*, vol. 25, no. 4, pp. 2903–2912, Oct 2010.

[70] M. Hiller, D. Krug, R. Sommer, and S. Rohner, "A new highly modular medium voltage converter topology for industrial drive applications," in *Power Electronics and Applications, 2009. EPE '09. 13th European Conference on*, Sept 2009, pp. 1–10.

[71] M. Winkelnkemper, A. Korn, and P. Steimer, "A modular direct converter for transformerless rail interties," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, July 2010, pp. 562–567.

[72] B. Li, R. Yang, D. Xu, G. Wang, W. Wang, and D. Xu, "Analysis of the phase-shifted carrier modulation for modular multilevel converters," *Power Electronics, IEEE Transactions on*, vol. 30, no. 1, pp. 297–310, Jan 2015.

[73] W. Rei, J.-F. Cordeau, M. Gendreau, and P. Soriano, "Accelerating benders decomposition by local branching," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 333–345, 2009.

[74] T. L. Magnanti and R. T. Wong, "Accelerating benders decomposition: Algorithmic enhancement and model selection criteria," *Operations research*, vol. 29, no. 3, pp. 464–484, 1981.

[75] G. K. Saharidis, M. Minoux, and M. G. Ierapetritou, "Accelerating benders method using covering cut bundle generation," *International Transactions in Operational Research*, vol. 17, no. 2, pp. 221–237, 2010.

[76] M. Ma and L. Fan, "Security constrained dc opf considering generator responses," *Electric Power Systems Research*, p. 106920, 2020.

[77] O. Alsac and B. Stott, "Optimal load flow with steady-state security," *IEEE transactions on power apparatus and systems*, no. 3, pp. 745–751, 1974.

[78] Y. Yang and Y. Feng, "Large-scale preventive security constrained optimal power flow based on compensation method," in *2015 IEEE Power & Energy Society General Meeting*. IEEE, 2015, pp. 1–5.

[79] Y. Xu, Z. Y. Dong, R. Zhang, K. P. Wong, and M. Lai, "Solving preventive-corrective scopf by a hybrid computational strategy," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1345–1355, 2013.

[80] Y. Wen and C. Guo, "Discussion on "solving preventive-corrective scopf by a hybrid computational strategy"," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 3124–3124, 2014.

[81] L. Yang, C. Zhang, and J. Jian, "A parallel method for solving the dc security constrained optimal power flow with demand uncertainties," *International Journal of Electrical Power & Energy Systems*, vol. 102, pp. 171–178, 2018.

[82] G. O. Competition. (2018) SCOPF Problem Formulation: Challenge 1. [Online]. Available: https://gocompetition.energy.gov/sites/default/files/SCOPF_Formulation_GO_Comp_20181114.pdf

[83] B. Zeng, Y. An, and L. Kuznia, "Chance constrained mixed integer program: Bilinear and linear formulations, and benders decomposition," *arXiv preprint arXiv:1403.7875*, 2014.

[84] Y. Zhang, J. Wang, B. Zeng, and Z. Hu, "Chance-constrained two-stage unit commitment under uncertain load and wind power output using bilinear benders decomposition," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3637–3647, 2017.

[85] H. Haghighat and B. Zeng, "Stochastic and chance-constrained conic distribution system expansion planning using bilinear benders decomposition," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2696–2705, 2017.

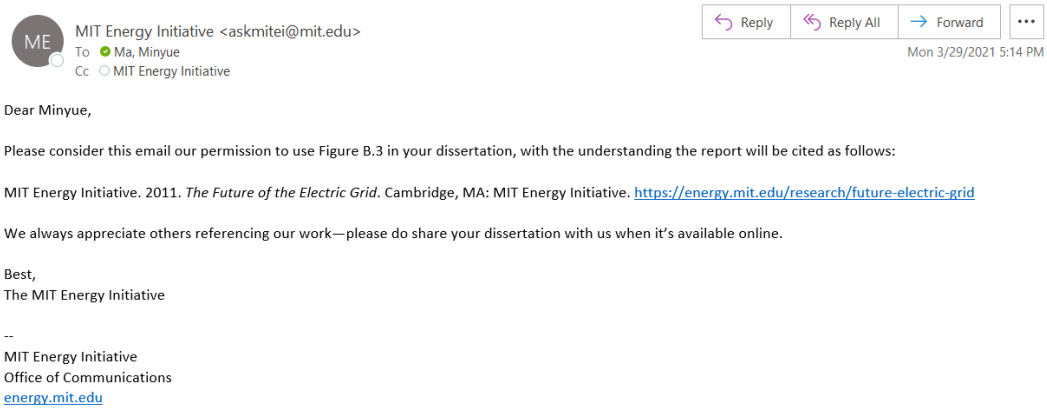
[86] Y. Wen, C. Guo, H. Pandžić, and D. S. Kirschen, "Enhanced security-constrained unit commitment with emerging utility-scale energy storage," *IEEE Transactions on power Systems*, vol. 31, no. 1, pp. 652–662, 2015.

[87] Q. Jiang and K. Xu, "A novel iterative contingency filtering approach to corrective security-constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1099–1109, 2013.

[88] F. Karbalaei, H. Shahbazi, and M. Mahdavi, "A new method for solving preventive security-constrained optimal power flow based on linear network compression," *International Journal of Electrical Power & Energy Systems*, vol. 96, pp. 23–29, 2018.

Appendix A: Copyright Permissions

The permission below is for the Fig. 1.1.



ME

MIT Energy Initiative <askmitei@mit.edu>
To Ma, Minyue
Cc MIT Energy Initiative

Reply | Reply All | Forward | ...
Mon 3/29/2021 5:14 PM

Dear Minyue,

Please consider this email our permission to use Figure B.3 in your dissertation, with the understanding the report will be cited as follows:

MIT Energy Initiative. 2011. *The Future of the Electric Grid*. Cambridge, MA: MIT Energy Initiative. https://energy.mit.edu/research/future-electric-grid

We always appreciate others referencing our work—please do share your dissertation with us when it's available online.

Best,
The MIT Energy Initiative

--
MIT Energy Initiative
Office of Communications
energy.mit.edu

The permission below is for the main contents and results of Chapter 2 and Chapter 3.

Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink®

Home | Help | Email Support | Sign in | Create Account

A sparse convex AC OPF solver and convex iteration implementation based on 3-node cycles

Author: Minyue Ma,Lingling Fan,Zhixin Miao,Bo Zeng,Hossein Ghassempour

Publication: Electric Power Systems Research

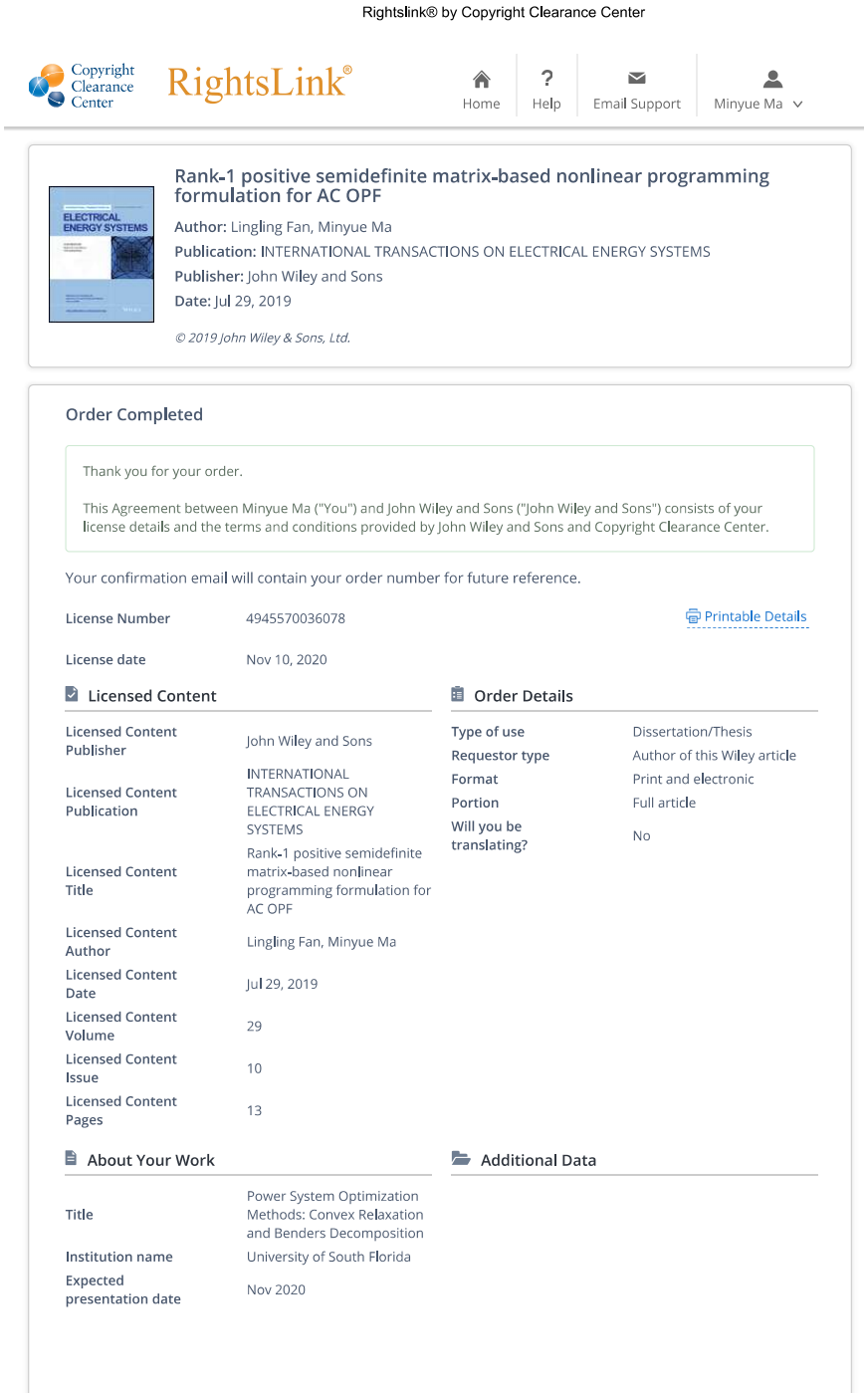
Publisher: Elsevier

Date: March 2020

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW

The permission below is for the main contents and results of Chapter 3.



Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink®

Home | Help | Email Support | Minyue Ma

Rank-1 positive semidefinite matrix-based nonlinear programming formulation for AC OPF

Author: Lingling Fan, Minyue Ma

Publication: INTERNATIONAL TRANSACTIONS ON ELECTRICAL ENERGY SYSTEMS

Publisher: John Wiley and Sons

Date: Jul 29, 2019

© 2019 John Wiley & Sons, Ltd.

Order Completed

Thank you for your order.

This Agreement between Minyue Ma ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

Your confirmation email will contain your order number for future reference.

Printable Details

License Number	4945570036078
License date	Nov 10, 2020

Licensed Content

Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	INTERNATIONAL TRANSACTIONS ON ELECTRICAL ENERGY SYSTEMS
Licensed Content Title	Rank-1 positive semidefinite matrix-based nonlinear programming formulation for AC OPF
Licensed Content Author	Lingling Fan, Minyue Ma
Licensed Content Date	Jul 29, 2019
Licensed Content Volume	29
Licensed Content Issue	10
Licensed Content Pages	13

Order Details

Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No

About Your Work

Title	Power System Optimization Methods: Convex Relaxation and Benders Decomposition
Institution name	University of South Florida
Expected presentation date	Nov 2020

Additional Data

Requestor Location

Requestor Location	Minyue Ma ENB 346, 4202 E Fowler Ave Smart Grid Lab University of South Florida TAMPA, FL 33620 United States Attn: Minyue Ma

Tax Details

Publisher Tax ID	EU826007151

Price

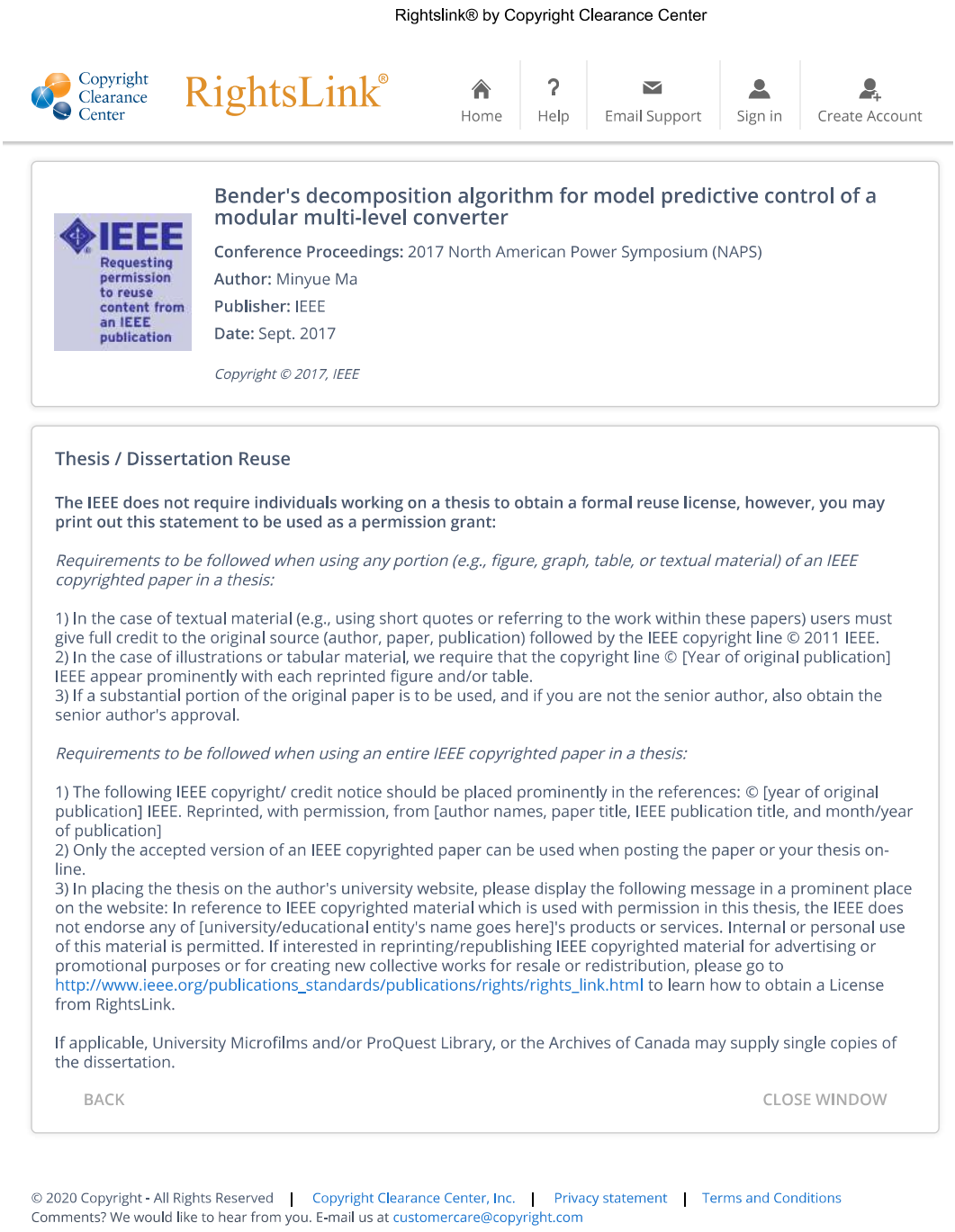
Total	0.00 USD

Would you like to purchase the full text of this article? If so, please continue on to the content ordering system located here: Purchase PDF
If you click on the buttons below or close this window, you will not be able to return to the content ordering system.

Total: 0.00 USD

CLOSE WINDOW ORDER MORE

The permission below is for the main contents and results of Chapter 4.



Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink®

Home | Help | Email Support | Sign in | Create Account

IEEE Requesting permission to reuse content from an IEEE publication

Bender's decomposition algorithm for model predictive control of a modular multi-level converter

Conference Proceedings: 2017 North American Power Symposium (NAPS)

Author: Minyue Ma

Publisher: IEEE

Date: Sept. 2017

Copyright © 2017, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

The permission below is for the main contents and results of Chapter 5.

Rightslink® by Copyright Clearance Center

Copyright Clearance Center RightsLink®

Home | Help | Email Support | Sign in | Create Account

Security constrained DC OPF considering generator responses

Author: Minyue Ma,Lingling Fan

Publication: Electric Power Systems Research

Publisher: Elsevier

Date: Available online 19 November 2020

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

BACK CLOSE WINDOW

About the Author

Minyue (Matthew) Ma, a Ph.D. majoring in Electrical Engineer at the University of South Florida. He was an intern-engineer at the National Renewable Energy Laboratory, where he assisted the NREL-MIDAS solar team to develop a renewable energy simulation software. Dr. Ma's research interests include power system analysis, modeling, optimization. He has had articles published in several peer-reviewed journals. He has also presented his works at international, national, and regional conferences. He received his Master's degree in Electrical Engineer from the University of South Florida, and his Bachelor's degree in Electrical Engineering from the Huazhong University of Science and Technology in China.