Distributed Control and Optimization for Communication and Power Systems

Thesis by

Qiuyu Peng

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2016

(Defended December 07, 2015)

© 2016
Qiuyu Peng
All Rights Reserved

This thesis is dedicated to
my girlfriend Huan,
whose love made this possible,
and my parents,
who have supported me all the way.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Steven Low, for his continuous support of my Ph.D study. Steven is a great scholar who dedicates to work on impactful and hard research problems. He introduced me to a variety of research areas in both communication and power networks. He gave me freedom and supported me to work on projects based on my own interests. It is his enthusiasm on research that motivates me to think big and work on important research problems no matter how hard they look. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor John Doyle, Professor Mani Chandy, Professor P. P. Vaidyanathan and Professor Adam Wierman. It's really my great honor to have them on my committee. They gave me their insightful comments and encouragement. I learned advanced control theory in John's class. Mani raised many interesting questions in both my candidacy exam and thesis defense that were very beneficial in the completion of my thesis. I was in P. P. V's class on signal processing and he could always explain the complex formula from different perspectives. Adam gave me great help and advice on writing good papers and making presentations. He taught me how to communicate complicated ideas through plain English that people could easily understand.

I am also thankful to my former research advisor, Professor Xinbing Wang, for his guidance during my undergraduate study at Shanghai Jiao Tong University. I was a junior when I joined Xinbing's lab. He gave me much advice on doing research and encouraged me to pursue my Ph.D. in US. It was his advice and encouragement that made it possible for me to study at Caltech.

My collaborators also gave me their great supports: Anwar Walid, Jaehyun Hwang from Bell Lab, Minghua Chen from Chinese University of Hong Kong and Seungil You, Yujie Tang from Caltech. It was a great pleasure to work with these great minds and I am grateful for all of those fruitful discussions.

I am grateful to all the colleagues in my research group RSRG. Special thanks to an incomplete list of current and past group members: Minghong Lin, Zhenhua Liu, Lingwen Gan, Desmond Cai, Changhong Zhao, Xiaoqi Ren and Niangjun Chen, etc.

The department of Electrical Engineering at California Institute of Technology provides a great

and cozy environment. It is a paradise for studying and doing research. I want to thank the great help from staff members, especially Christine Ortega, Sydney Garstang and Tanya Owen.

Finally, I would like to thank my parents for their spiritual support during the past five years. It is their supports and love that gave me the strength and stamina to finish my Ph.D. journey. I dedicate this thesis to my parents as an inadequate appreciation of everything that they have done for me.

Abstract

We are at the cusp of a historic transformation of both communication system and electricity system. This creates challenges as well as opportunities for the study of networked systems. Problems of these systems typically involve a huge number of end points that require intelligent coordination in a distributed manner. In this thesis, we develop models, theories, and scalable distributed optimization and control algorithms to overcome these challenges.

This thesis focuses on two specific areas: *multi-path TCP (Transmission Control Protocol)* and *electricity distribution system operation and control*. Multi-path TCP (MP-TCP) is a TCP extension that allows a single data stream to be split across multiple paths. MP-TCP has the potential to greatly improve reliability as well as efficiency of communication devices. We propose a fluid model for a large class of MP-TCP algorithms and identify design criteria that guarantee the existence, uniqueness, and stability of system equilibrium. We clarify how algorithm parameters impact TCP-friendliness, responsiveness, and window oscillation and demonstrate an inevitable tradeoff among these properties. We discuss the implications of these properties on the behavior of existing algorithms and motivate a new algorithm *Balia* (balanced linked adaptation) which generalizes existing algorithms and strikes a good balance among TCP-friendliness, responsiveness, and window oscillation. We have implemented *Balia* in the Linux kernel. We use our prototype to compare the new proposed algorithm *Balia* with existing MP-TCP algorithms.

Our second focus is on designing computationally efficient algorithms for electricity distribution system operation and control. First, we develop efficient algorithms for feeder reconfiguration in distribution networks. The feeder reconfiguration problem chooses the on/off status of the switches in a distribution network in order to minimize a certain cost such as power loss. It is a mixed integer nonlinear program and hence hard to solve. We propose a heuristic algorithm that is based on the recently developed convex relaxation of the optimal power flow problem. The algorithm is efficient and can successfully computes an optimal configuration on all networks that we have tested. Moreover we prove that the algorithm solves the feeder reconfiguration problem optimally under certain conditions. We also propose a more efficient algorithm and it incurs a loss in optimality of less than 3% on the test networks.

Second, we develop efficient distributed algorithms that solve the optimal power flow (OPF)

problem on distribution networks. The OPF problem determines a network operating point that minimizes a certain objective such as generation cost or power loss. Traditionally OPF is solved in a centralized manner. With increasing penetration of volatile renewable energy resources in distribution systems, we need faster and distributed solutions for real-time feedback control. This is difficult because power flow equations are nonlinear and kirchhoff's law is global. We propose solutions for both balanced and unbalanced radial distribution networks. They exploit recent results that suggest solving for a globally optimal solution of OPF over a radial network through a second-order cone program (SOCP) or semi-definite program (SDP) relaxation. Our distributed algorithms are based on the alternating direction method of multiplier (ADMM), but unlike standard ADMM-based distributed OPF algorithms that require solving optimization subproblems using iterative methods, the proposed solutions exploit the problem structure that greatly reduce the computation time. Specifically, for balanced networks, our decomposition allows us to derive closed form solutions for these subproblems and it speeds up the convergence by 1000x times in simulations. For unbalanced networks, the subproblems reduce to either closed form solutions or eigenvalue problems whose size remains constant as the network scales up and computation time is reduced by 100x compared with iterative methods.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

We are at the cusp of a historic transformation of both communication system and electricity system. This creates challenges as well as opportunities for the study of networked systems. Problems of these systems typically involve a huge number of end points that require intelligent coordinations in a distributed manner. In this thesis, we develop models, theories, and scalable distributed optimization and control algorithms that overcome these challenges.

Specifically, this thesis focuses on three topics: *multi-path TCP*, *feeder reconfiguration in distribution systems* and *distributed OPF algorithm on radial distribution networks*.

1.1 Multipath TCP

Traditional TCP uses a single path through the network even though multiple paths are usually available in today's communication infrastructure; e.g., most smart phones are enabled with both cellular and WiFi access, and servers in data centers are connected to multiple routers. Multi-path TCP (MP-TCP) has the potential to greatly improve application performance by using multiple paths transparently. It is being standardized by the MP-TCP Working Group of the Internet Engineering Task Force (IETF) [30]. We present a fluid model of MP-TCP and study how protocol parameters affect structural properties such as the existence, uniqueness and stability of equilibrium, the tradeoffs among TCP friendliness, responsiveness and window oscillation. These properties motivate a new algorithm that generalizes existing MP-TCP algorithms.

Various congestion control algorithms have been proposed as an extension of TCP NewReno for MP-TCP. A straightforward extension is to run TCP NewReno on each subpath, e.g. [40, 43]. This algorithm, however, can be highly unfriendly when it shares a path with a single-path TCP user. This motivates the Coupled algorithm, which is fair because it has the same underlying utility function as TCP NewReno, e.g. [38, 46]. It is found in [89], however, that the Coupled algorithm responds slowly in a dynamic network environment. A different algorithm is proposed in [89] (which we refer to as the Max algorithm) which is more responsive than the Coupled algorithm and still

reasonably friendly to single-path TCP users. Recently, opportunistic linked increase algorithm (OLIA) is proposed as a variant of Coupled algorithm that is as friendly as the Coupled algorithm but more responsive [50]. See [77] for more references to early work on multi-path congestion control.

Our goal is to develop structural understanding of MP-TCP algorithms so that we can systematically trade off different properties such as TCP friendliness, responsiveness, and window oscillation. Window oscillation can be detrimental to applications that require a steady throughput. For single-path TCP, one can associate a strictly concave utility function with each source so that the congestion control algorithm implicitly solves a network utility maximization problem [47, 62, 77]. The convexity of this underlying utility maximization guarantees the existence, uniqueness, and stability of most single-path TCP algorithms. For many MP-TCP proposals considered by IETF, it will be shown that the utility maximization interpretation fails to hold in general, necessitating the need for a different approach to understanding the equilibrium properties of these algorithms. Moreover the relations among different performance metrics, such as fairness, responsiveness, and window oscillation, need to be clarified.

We propose a fluid model for a large class of MP-TCP algorithms and the existence, uniqueness, and stability of system equilibrium are identified. The impact of algorithm parameters on TCP-friendliness, responsiveness, and window oscillation are clarified and an inevitable tradeoff among these properties are also derived. The implications of these properties on the behavior of existing algorithms motivate the algorithm Balia, which outperforms all the existing algorithms in our experiment.

1.2 Feeder Reconfiguration in Distribution Networks

A primary distribution system consists of buses, distribution lines, and (sectionalizing and tie) switches that can be opened or closed. There are two types of buses. *Substation buses* (or just *substations*) are connected to a transmission network from which they receive bulk power, and *load buses* that receive power from the substation buses. During normal operation the switches are configured so that

1. There is no loop in the network.

2. Each load bus is connected to a single substation.

Hence, there is a tree component rooted at each substation and we refer to each such component as a *feeder*. The optimal feeder reconfiguration (OFR) problem seeks to alter the on/off status of these switches, for the purpose of load balancing or loss minimization subject to the above two requirements, e.g., [6, 17, 19, 65]. See also a survey in [76] for many early papers and references to some recent work in [45].

The OFR problem is a combinatorial (on/off status of switches) optimization problem with nonlinear constraints (power flow equations) and can generally be NP-hard. Various algorithms have been developed to solve the OFR problems. Following the convention in [45], they roughly fall into two categories: formal methods and heuristic methods.

Formal methods solve the OFR problem using generic mixed integer optimization approaches. They usually require a significant amount of computation time, e.g. simulated annealing [17], ordinal optimization [24], bender decomposition [51], etc..

Heuristic methods exploit problem structures to solve OFR. They are usually more efficient than formal methods but lack theoretical guarantee on performance, e.g., iterative branch exchange approach [6, 19, 36] and successive branch reduction approach [35, 39, 65] etc..

We propose a new heuristic method with guaranteed performance. The effectiveness of this new approach is illustrated both through simulations on standard test systems and mathematical analysis. Specifically, the proposed algorithm only involves solving a small number of OPF problems and *no* computationally intensive mixed-integer optimization. We prove that the proposed heuristic can obtain the global optimal solution under certain assumptions. Indeed global optimal configurations can always be found on the four practical networks in our simulations.

1.3 Distributed OPF Algorithm on Radial Distribution Networks

The optimal power flow (OPF) problem seeks to optimize certain objectives such as power loss and generation cost subject to power flow physics and operational constraints. It is a fundamental problem because it underlies many power system operations and planning problems such as economic dispatch, unit commitment, state estimation, stability and reliability assessment, volt/var control, demand response, etc. The continued growth of highly volatile renewable sources on distribution systems calls for real-time feedback control. Solving the OPF problems in such an environment has at least two challenges.

First, the OPF problem is hard to solve because of its nonconvex feasible set. Recently a new approach through convex relaxation has been developed. Specifically semidefinite program (SDP) relaxation [4] and second order cone program (SOCP) relaxation [44] have been proposed in the bus injection model, and SOCP relaxation has been proposed in the branch flow model [25, 27]. See the tutorial [60, 61] for further pointers to the literature. When an optimal solution of the original OPF problem can be recovered from any optimal solution of a convex relaxation, we say the relaxation is *exact*. For radial distribution networks (whose graphs are trees), several sufficient conditions have been proved that guarantee SOCP and SDP relaxations are exact. This is important because almost all distribution systems are radial. Moreover some of these conditions have been shown to hold for

many practical networks. In those cases we can rely on off-the-shelf convex optimization solvers to obtain a globally optimal solution for the nonconvex OPF problem.

Second, most algorithms proposed in the literature are centralized and meant for applications in today's energy management systems that, e.g., centrally schedule a relatively small number of generators. In future networks that simultaneously optimize (possibly in real time) the operation of a large number of intelligent endpoints, a centralized approach will not scale because of its computation and communication overhead.

We will address the second challenge. Specifically, we develop computationally efficient distributed algorithms that can scale to large real networks for both balanced and unbalanced radial distribution networks.

1.4 Thesis Overview

The thesis is organized as follows:

1. In Chapter 2, we develop a general theoretical framework to model mutli-path TCP, which leads to a better algorithm (*Balia*) that strikes a better balance among competing performance criteria. We also implement *Balia* in the Linux kernel. This work is based on [74, 75].

2. In Chapter 3, we formulate the optimal power flow problem on both balanced and unbalanced networks and show how to solve them through second order cone program (SOCP) and semidefinite program (SDP) relaxation. They are the foundations of our works on feeder reconfiguration and distributed OPF algorithm.

3. In Chapter 4, we formulate the feeder reconfiguration problem in distribution networks, which is a mixed integer nonlinear program. We then develop heuristic algorithms and show the performance through both rigorous analysis and simulations on standard test networks. This work is based on [72, 73].

4. In Chapter 5, we first review alternating direction method of multipliers(ADMM). Based on the general ADMM, we propose efficient centralized and distributed algorithms for a broad class of graphical optimization problems.

5. In Chapter 6 and 7, we show how to develop distributed algorithms that solve the OPF problem efficiently on both balanced and unbalanced radial distribution networks through the ADMM based algorithm in Chapter 5. This work is based on [70, 71].

Chapter 2

Multipath TCP: Analysis, Design and Implementation

Multi-path TCP (MP-TCP) has the potential to greatly improve application performance by using multiple paths transparently. A fluid model is proposed for a large class of MP-TCP algorithms and the existence, uniqueness, and stability of system equilibrium are identified. The impact of algorithm parameters on TCP-friendliness, responsiveness, and window oscillation are clarified and an inevitable tradeoff among these properties are also derived. The implications of these properties on the behavior of existing algorithms are discussed and they motivate the algorithm *Balia* (balanced linked adaptation), which generalizes existing algorithms and strikes a good balance among TCP-friendliness, responsiveness, and window oscillation. *Balia* is implemented in the Linux kernel and compared with existing MP-TCP algorithms.

Literature Traditional TCP uses a single path through the network even though multiple paths are usually available in today's communication infrastructure; e.g., most smart phones are enabled with both cellular and WiFi access, and servers in data centers are connected to multiple routers. Multi-path TCP (MP-TCP) has the potential to greatly improve application performance by using multiple paths transparently. It is being standardized by the MP-TCP Working Group of the Internet Engineering Task Force (IETF) [30]. In this chapter we present a fluid model of MP-TCP and study how protocol parameters affect structural properties such as the existence, uniqueness and stability of equilibrium, the tradeoffs among TCP friendliness, responsiveness, and window oscillation. These properties motivate a new algorithm that generalizes existing MP-TCP algorithms.

Various congestion control algorithms have been proposed as an extension of TCP NewReno for MP-TCP. A straightforward extension is to run TCP NewReno on each subpath, e.g. [40, 43]. This algorithm, however, can be highly unfriendly when it shares a path with a single-path TCP user. This motivates the Coupled algorithm, which is fair because it has the same underlying utility function as TCP NewReno, e.g. [38, 46]. It is found in [89], however, that the Coupled algorithm

can respond slowly in a dynamic network environment. A different algorithm is proposed in [89] (which we refer to as the Max algorithm) which is more responsive than the Coupled algorithm and still reasonably friendly to single-path TCP users. Recently, opportunistic linked increase algorithm (OLIA) is proposed as a variant of Coupled algorithm that is as friendly as the Coupled algorithm but more responsive [50]. See [77] for more references to early work on multi-path congestion control.

Our goal is to develop structural understanding of MP-TCP algorithms so that we can systematically tradeoff different properties such as TCP friendliness, responsiveness, and window oscillation. Window oscillation can be detrimental to applications that require a steady throughput. For single-path TCP, one can associate a strictly concave utility function with each source so that the congestion control algorithm implicitly solves a network utility maximization problem [47, 62, 77]. The convexity of this underlying utility maximization guarantees the existence, uniqueness, and stability of most single-path TCP algorithms. For many MP-TCP proposals considered by IETF, it will be shown that the utility maximization interpretation fails to hold in general, necessitating the need for a different approach to understanding the equilibrium properties of these algorithms. Moreover the relations among different performance metrics, such as fairness, responsiveness, and window oscillation, need to be clarified.

Summary The main contributions of this work are three-fold. First we present a fluid model that covers a broad class of MP-TCP algorithms and identify the exact property that allows an algorithm to have an underlying utility function. This implies that some MP-TCP algorithms, e.g., the Max algorithm [89], has no associated utility function. We prove conditions on protocol parameters that guarantee the existence and uniqueness of the equilibrium, and its asymptotical stability. Indeed, algorithms that fail to satisfy these conditions, e.g. the Coupled algorithm, can be unstable and can have multiple equilibria as shown in [89]. Second, we clarify how protocol parameters impact TCP friendliness, responsiveness, and window oscillation and demonstrate the inevitable tradeoff among these properties. Finally, based on our understanding of the design space, we propose *Balia (Balanced linked adaptation)* MP-TCP algorithm that generalizes existing algorithms and strikes a good balance among these properties. This algorithm has been implemented in the Linux kernel and we evaluate its performance using our Linux prototype.

We now summarize our proposed *Balia* MP-TCP algorithm. Each source s has a set of routes r. Each route r maintains a congestion window w_r and measures its round-trip time τ_r. The window adaptation is as follows:

- For each ACK on route $r \in s$,

$$w_r \leftarrow w_r + \frac{x_r}{\tau_r \left(\sum x_k\right)^2} \left(\frac{1+\alpha_r}{2}\right) \left(\frac{4+\alpha_r}{5}\right), \tag{2.1}$$

- For each packet loss on route $r \in s$,

$$w_r \leftarrow w_r - \frac{w_r}{2} \min\{\alpha_r, 1.5\}, \tag{2.2}$$

where $x_r := w_r/\tau_r$ and $\alpha_r := \frac{\max\{x_k\}}{x_r}$.

2.1 Multipath TCP model

In this section we first propose a fluid model of MP-TCP and then use it to model MP-TCP algorithms in the literature. Unless otherwise specified, a boldface letter $\mathbf{x} \in \mathbb{R}^n$ denotes a vector with components x_i. We use $\mathbf{x}_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ to denote the $n-1$ dimensional vector without x_i and $\|\mathbf{x}\|_k := (\sum x_i^k)^{1/k}$ to denote the L_k-norm of $\mathbf{x}$. Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \geq \mathbf{y}$ means $x_i \geq y_i$ for all components i. A capital letter denotes a matrix or a set, depending on the context. A symmetric matrix P is said to be *positive (negative) semidefinite* if $\mathbf{x}^T P \mathbf{x} \geq 0 (\leq 0)$ for any $\mathbf{x}$, and *positive (negative) definite* if $\mathbf{x}^T P \mathbf{x} > 0 (< 0)$ for any $\mathbf{x} \neq \mathbf{0}$. For any matrix P, define $[P]^+ := (P + P^T)/2$ to be its symmetric part. Given two arbitrary matrices A and B (not necessarily symmetric), $A \succeq B$ means $[A - B]^+$ is positive semidefinite. For a vector $\mathbf{x}$, $diag\{\mathbf{x}\}$ is a diagonal matrix with entries given by $\mathbf{x}$.

2.1.1 Fluid model

Consider a network that consists of a set $L = \{1, \ldots, |L|\}$ of links with finite capacities c_l. The network is shared by a set $S = \{1, \ldots, |S|\}$ of sources. Available to source $s \in S$ is a fixed collection of routes (or paths) r. A route r consists of a set of links l. We abuse notation and use s both to denote a source and the set of routes r available to it, depending on the context. Likewise, r is used both to denote a route and the set of links l in the route. Let $R := \{r \mid r \in s, s \in S\}$ be the collection of all routes. Let $H \in \{0,1\}^{|L| \times |R|}$ be the routing matrix: $H_{lr} = 1$ if link l is in route r (denoted by '$l \in r$'), and 0 otherwise.

For each route $r \in R$, τ_r denotes its round trip time (RTT). For simplicity we assume τ_r are constants. Each source s maintains a congestion window $w_r(t)$ at time t for every route $r \in s$. Let $x_r(t) := w_r(t)/\tau_r$ represent the sending rate on route r. Each link l maintains a congestion price $p_l(t)$ at time t. Let $q_r(t) := \sum_{l \in L} H_{lr} p_l(t)$ be the aggregate price on route r. In this chapter $p_l(t)$ represents the packet loss probability at link l and $q_r(t)$ represents the approximate packet loss probability on route r.

We associate three state variables $(x_r(t), w_r(t), q_r(t))$ for each route $r \in s$. Let $\mathbf{x}_s(t) := (x_r(t), r \in s)$, $\mathbf{w}_s(t) := (w_r(t), r \in s)$, $\mathbf{q}_s(t) := (q_r(t), r \in s)$. Then $(\mathbf{x}_s(t), \mathbf{w}_s(t), \mathbf{q}_s(t))$ represents the corresponding state variables for each source $s \in S$. For each link l, let $y_l(t) := \sum_{r \in R} H_{lr} x_r(t)$ be its

aggregate traffic rate.

Congestion control is a distributed algorithm that adapts $\mathbf{x}(t)$ and $\mathbf{p}(t)$ in a closed loop. Motivated by the AIMD algorithm of TCP Newreno, we model MP-TCP by

$$\dot{x}_r = k_r(\mathbf{x}_s)\left(\phi_r(\mathbf{x}_s) - q_r\right)^+_{x_r} \qquad r \in s \quad s \in S \tag{2.3}$$

$$\dot{p}_l = \gamma_l\left(y_l - c_l\right)^+_{p_l} \qquad l \in L, \tag{2.4}$$

where $(a)^+_x = a$ for $x > 0$ and $\max\{0, a\}$ for $x \leq 0$. We omit the time t in the expression for simplicity. (2.3) models how sending rates are adapted in the congestion avoidance phase of TCP at each end system and (2.4) models how the congestion price is (often implicitly) updated at each link. The MP-TCP algorithm installed at source s is specified by (K_s, Φ_s), where $K_s(\mathbf{x}_s) := (k_r(\mathbf{x}_s), r \in s)$ and $\Phi_s(\mathbf{x}_s) := (\phi_r(\mathbf{x}_s), r \in s)$. Here $K_s(\mathbf{x}_s) \geq 0$ is a vector of positive gains that determines the dynamic property of the algorithm. $\Phi_s(\mathbf{x}_s)$ determines the equilibrium properties of the algorithm. The link algorithm is specified by γ_l, where $\gamma_l > 0$ is a positive gain that determines the dynamic property. This is a simplified model of the RED algorithm that assumes the loss probability is proportional to the backlog, and is used in, e.g., [47, 62].

2.1.2 Existing MP-TCP algorithms

We first show how to relate the fluid model (2.3) to the window-based MP-TCP algorithms proposed in the literature. On each route r the source increases its window at the return of each ACK. Let this increment be denoted by $I_r(\mathbf{w}_s)$, where $\mathbf{w}_s$ is the vector of window sizes on different routes of source s. The source decreases the window on route r when it sees a packet loss on route r. Let this decrement be denoted by $D_r(\mathbf{w}_s)$. Then most loss based MP-TCP algorithms take the form of the following pseudo code:

- For each ACK on route r, $w_r \leftarrow w_r + I_r(\mathbf{w}_s)$.
- For each loss on route r, $w_r \leftarrow w_r - D_r(\mathbf{w}_s)$.

We now model the above pseudo codes by the fluid model (2.3). Let δw_r be the net change to window on route r in each round trip time. Then δw_r is roughly

$$\begin{aligned} \delta w_r &= \left(I_r(\mathbf{w}_s)(1 - q_r) - D_r(\mathbf{w}_s)q_r\right)w_r \\ &\approx \left(I_r(\mathbf{w}_s) - D_r(\mathbf{w}_s)q_r\right)w_r, \end{aligned}$$

since the loss probability q_r is small. On the other hand

$$\delta w_r \approx \dot{w}_r\tau_r = \dot{x}_r\tau_r^2.$$

Therefore

$$\dot{x}_r = \frac{x_r}{\tau_r}\left(I_r(\mathbf{w}_s) - D_r(\mathbf{w}_s)q_r\right).$$

From (2.3) we have

$$\begin{cases} k_r(\mathbf{x}_s) &= \frac{x_r}{\tau_r} D_r(\mathbf{w}_s) \\ \phi_r(\mathbf{x}_s) &= \frac{I_r(\mathbf{w}_s)}{D_r(\mathbf{w}_s)} \end{cases}. \tag{2.5}$$

We now apply this to the algorithms in the literature. We first summarize these algorithms in the form of a pseudo-code and then use (2.5) to derive parameters $k_r(\mathbf{x}_s)$ and $\phi_r(\mathbf{x}_s)$ of the fluid model (2.3).

Single-path TCP (TCP-NewReno)

Single-path TCP is a special case of MP-TCP algorithm with $|s| = 1$. Hence x_s is a scalar and we identify each source with its route $r = s$. TCP-NewReno adjusts the window as follows:

- For each ACK on route r, $w_r \leftarrow w_r + 1/w_r$.
- For each loss on route r, $w_r \leftarrow w_r/2$.

From (2.5), this can be modeled by the fluid model (2.3) with

$$k_r(x_s) = \frac{1}{2}x_r^2, \quad \phi_r(x_s) = \frac{2}{\tau_r^2 x_r^2}.$$

We now summarize some existing MP-TCP algorithms, all of which degenerate to TCP NewReno if there is only one route per source.

EWTCP [40]

EWTCP algorithm applies TCP-NewReno like algorithm on each route independently of other routes. It adjusts the window on multiple routes as follows:

- For each ACK on route r, $w_r \leftarrow w_r + a/w_r$.
- For each loss on route r, $w_r \leftarrow w_r/2$.

From (2.5), this can be modeled by the fluid model (2.3) with

$$k_r(\mathbf{x}_s) = \frac{1}{2}x_r^2, \quad \phi_r(\mathbf{x}_s) = \frac{2a}{\tau_r^2 x_r^2}.$$

where $a > 0$ is a constant.

Coupled MPTCP [38, 46]

The Coupled MPTCP algorithm adjusts the window on multiple routes in a coordinated fashion as follows:

- For each ACK on route r, $w_r \leftarrow w_r + \frac{w_r/\tau_r^2}{(\sum_{k\in s} w_k/\tau_k)^2}$.
- For each loss on route r, $w_r \leftarrow w_r/2$.

From (2.5), this can be modeled by the fluid model (2.3) with

$$k_r(\mathbf{x}_s) = \frac{1}{2}x_r^2, \quad \phi_r(\mathbf{x}_s) = \frac{2}{\tau_r^2(\sum_{k\in s} x_k)^2}.$$

Semicoupled MPTCP [89]

The Semi-coupled MPTCP algorithm adjusts the window on multiple routes as follows:

- For each ACK on route r, $w_r \leftarrow w_r + \frac{1}{\tau_r(\sum_{k\in s} w_k/\tau_k)}$.
- For each loss on route r, $w_r \leftarrow w_r/2$.

From (2.5), this can be modeled by the fluid model (2.3) with

$$k_r(\mathbf{x}_s) = \frac{1}{2}x_r^2, \quad \phi_r(\mathbf{x}_s) = \frac{2}{x_r\tau_r(\sum_{k\in s} x_k)}.$$

Max MPTCP [89]

The Max MPTCP algorithm adjusts the window on multiple routes as follows:

- For each ACK on route r, $w_r \leftarrow w_r + \min\left\{\frac{\max\{w_k/\tau_k^2\}}{(\sum w_k/\tau_k)^2}, \frac{1}{w_r}\right\}$.
- For each loss on route r, $w_r \leftarrow w_r/2$.

From (2.5), this can be modeled by the fluid model (2.3) with

$$k_r(\mathbf{x}_s) = \frac{1}{2}x_r^2, \quad \phi_r(\mathbf{x}_s) = \frac{2\max\{x_k/\tau_k\}}{x_r\tau_r(\sum_{k\in s} x_k)^2},$$

where we have ignored taking the minimum with the $1/w_r$ term since the performance is mainly captured by $\frac{\max\{w_k/\tau_k^2\}}{(\sum w_k/\tau_k)^2}$.

Recently, OLIA MP-TCP algorithm [50] is shown to achieve good performance in many scenarios. OLIA uses complicated feedback congestion control signals and cannot be modeled by (2.3)-(2.4). We do, however, include OLIA in our Linux-based performance evaluation in Section 2.4.

Table 2.1: MP-TCP algorithms

	C0	C1	C2, C3	C4	C5
EWTCP	Yes	Yes	Yes	Yes	Yes
Coupled	Yes	Yes	No	Yes	Yes
Semicoupled	No	Yes	Yes	Yes	Yes
Max	No	Yes	Yes	Yes	Yes
Generalized	No	Yes	Yes	Yes	Yes
Theorem	2.1	2.2, 2.3, 2.5		2.4	2.6

2.2 Structural properties

Throughout this chapter we assume, for all $\mathbf{x}_s$, $r \in s$, $s \in S$, $k_r(\mathbf{x}_s) > 0$ and $\phi_r(\mathbf{x}_s) = 0$ only if $x_k = \infty$ for some $k \in s$. A point $(\mathbf{x}, \mathbf{p})$ is called an *equilibrium* of (2.3)–(2.4) if it satisfies, for all $r \in s, s \in S$ and $l \in L$,

$$k_r(\mathbf{x}_s)\,(\phi_r(\mathbf{x}_s) - q_r)^+_{x_r} = 0$$
$$\gamma_l\,(y_l - c_l)^+_{p_l} = 0$$

or equivalently,

$$x_r \geq 0,\ \phi_r(\mathbf{x}_s) \leq q_r \quad \text{and} \quad \phi_r(\mathbf{x}_s) = q_r \text{ if } x_r > 0 \tag{2.6}$$
$$p_l \geq 0,\ y_l \leq c_l \quad \text{and} \quad y_l = c_l \text{ if } p_l > 0. \tag{2.7}$$

We make two remarks. First an equilibrium $(\mathbf{x}, \mathbf{p})$ does not depend on K_s, but only on Φ_s. The design $(K_s, s \in S)$, however, affects dynamic properties such as stability and responsiveness as we show below. Second, since $k_r(\mathbf{x}_s) > 0$ and $\phi_r(\mathbf{x}_s) = 0$ only if $x_k = \infty$ for some $k \in s$ by assumption, any finite equilibrium $(\mathbf{x}, \mathbf{p})$ must have $q_r > 0$ for all r. In the following we always restrict ourselves to finite equilibria.

In this section we denote an MP-TCP algorithm by $(K, \Phi) := (K_s, \Phi_s, s \in S)$. We characterize MP-TCP designs (K, Φ) that guarantee the existence, uniqueness, and stability of system equilibrium. We identify design criteria that determine TCP-friendliness, responsiveness, and window oscillation and prove an inevitable tradeoff among these properties. We discuss in chapter 2.3 the implications of these structural properties on existing algorithms. All proofs are relegated to the Appendices.

2.2.1 Summary

We first present some properties of an MP-TCP algorithm (K, Φ) that we have identified. We then interpret them and summarize their implications.

C0: For each $s \in S$ and each $\mathbf{x}_s$, the Jacobians of $\Phi_s(\mathbf{x}_s)$ is continuous and symmetric, i.e.,

$$\frac{\partial \Phi_s}{\partial \mathbf{x}_s}(\mathbf{x}_s) \quad = \quad \left[\frac{\partial \Phi_s}{\partial \mathbf{x}_s}(\mathbf{x}_s)\right]^T.$$

C1: For each $s \in S$ there exists a nonnegative solution $\mathbf{x}_s := \mathbf{x}_s(\mathbf{p})$ to (2.6) for any finite $\mathbf{p} \geq 0$ such that $q_r > 0$ for all r. Moreover,

$$\frac{\partial y_l^s(\mathbf{p})}{\partial p_l} \leq 0, \qquad \lim_{p_l \to \infty} y_l^s(\mathbf{p}) = 0,$$

where $y_l^s(\mathbf{p}) := \sum_{r \in s} H_{lr} x_r(\mathbf{p})$ is the aggregate traffic at link l from source s.

C2: For each $s \in S$ and each $\mathbf{x}_s$, $\Phi_s(\mathbf{x}_s)$ is continuously differentiable; moreover, the symmetric part $[\partial \Phi_s(\mathbf{x}_s)/\partial \mathbf{x}_s]^+$ of the Jacobian is negative definite.

C3: For each $r \in R$, $\phi_r(\mathbf{x}_s) = \infty$ if and only if $x_r = 0$. The routing matrix H has full row rank.

C4: For each $r \in s$, $s \in S$, $\sum_{j \in s} [D_s]_{jr}(\mathbf{x}_s) \leq 0$, where $D_s(\mathbf{x}_s) := \left[\frac{\partial \Phi_s}{\partial \mathbf{x}_s}(\mathbf{x}_s)\right]^{-1}$.

C5: For each $r \in R$ and each $\mathbf{x}_{-r}$, $\lim_{x_r \to \infty} \phi_r(\mathbf{x}_s) = 0$.

These design criteria are intuitive and usually (but not always) satisfied; see Table 2.1.

Condition C0 guarantees the existence of utility functions $U_s(\mathbf{x}_s)$ that an equilibrium $(\mathbf{x}, \mathbf{p})$ of a multipath TCP/AQM (2.3)–(2.4) implicitly maximizes (Theorem 2.1). It is always satisfied when there is only a single path ($|s| = 1$ for all s) but not when $|s| > 1$.

Conditions C1–C3 guarantee the existence, uniqueness, and global asymptotic stability of the equilibrium $(\mathbf{x}, \mathbf{p})$ (Theorems 2.2 and 2.3). C1 says that the aggregate traffic rate through a link l from source s decreases when the congestion price p_l on that link increases, and it decreases to 0 as p_l increases without bounds. C2 implies that at steady state, if $\mathbf{x}_s, \mathbf{q}_s$ are perturbed by $\delta \mathbf{x}_s, \delta \mathbf{q}_s$, respectively, then $(\delta \mathbf{x}_s)^T \delta \mathbf{q}_s < 0$. In the case of single-path TCP ($|s| = 1$ for all s), C2 is equivalent to the curvature of the utility function $U_s(x_s)$ being negative, i.e., $U_s(x_s)$ is strictly concave. C3 means that the rate on route r is zero if and only if it sees infinite price on that route.

Condition C4 is natural and satisfied by all the algorithms considered in this chapter. It allows us to formally compare MP-TCP algorithms in terms of their TCP-friendliness (see formal definition below): under C1–C4, an MP-TCP algorithm (K, Φ) is more friendly if $\phi_r(\mathbf{x}_s)$ is smaller (Theorem 2.4). The existence of D_s in C4 is ensured by C2. To interpret C4, note that Lemma 2.10 in Appendix 2.B implies that $\Phi_s(\mathbf{x}_s^*) = \mathbf{q}_s^*$ at equilibrium. The implicit function theorem then implies

$\mathbf{1}^T \frac{\partial \mathbf{x}_s}{\partial q_r} = \sum_{j \in s} D_{jr}$ at equilibrium for all $r \in s$. Thus C4 says that the aggregate throughput $\mathbf{1}^T \mathbf{x}_s$ at equilibrium over all routes $r \in s$ of an MP-TCP flow is a nonincreasing function of the price q_r.

Condition C5 is also satisfied by all the algorithms considered in this paper. It means that the sending rate on a route r grows unbounded when the congestion price q_r is zero. Under C1–C3, an MP-TCP algorithm (K, Φ) is more responsive (see formal definition below) if the Jacobian of $\Phi_s(\mathbf{x}_s)$ is more negative definite (Theorem 2.5). C5 then implies an inevitable tradeoff: an MP-TCP algorithm that is more responsive is necessarily less TCP-friendly (Theorem 2.6).

We now elaborate on each of these properties.

2.2.2 Utility maximization

For single-path TCP (SP-TCP), one can associate a utility function $U_s(x_s) \in \mathbb{R}_+ \to \mathbb{R}$ with each flow s (x_s is a scalar and $|s| = 1$) and interpret (2.3)–(2.4) as a distributed algorithm to maximize aggregate users' utility, e.g. [47, 59, 62, 77]. Indeed, for SP-TCP, an $(\mathbf{x}, \mathbf{p})$ is an equilibrium if and only if $\mathbf{x}$ is optimal for

$$\text{maximize} \sum_{s \in S} U_s(x_s) \quad \text{s.t. } y_l \le c_l \quad l \in L \tag{2.8}$$

and $\mathbf{p}$ is optimal for the associated dual problem. Here $y_l \le c_l$ means the aggregate traffic y_l at each link does not exceed its capacity c_l. In fact this holds for a much wider class of SP-TCP algorithms than those specified by (2.3)–(2.4) [59]. Furthermore, all the main TCP algorithms proposed in the literature have strictly concave utility functions, implying a unique stable equilibrium.

The case of MP-TCP is much more delicate: whether an underlying utility function exists depends on the design choice of Φ_s and not all MP-TCP algorithms have one. Consider the multipath equivalent of (2.8):

$$\text{maximize} \sum_{s \in S} U_s(\mathbf{x}_s) \quad \text{s.t. } y_l \le c_l \quad l \in L, \tag{2.9}$$

where $\mathbf{x}_s := (x_r, r \in s)$ is the rate vector of flow s and $U_s : \mathbb{R}_+^{|s|} \to \mathbb{R}$ is a concave function.

Theorem 2.1 (utility maximization)**.** *There exists a twice continuously differentiable and concave $U_s(\mathbf{x}_s)$ such that an equilibrium $(\mathbf{x}, \mathbf{p})$ of (2.3)–(2.4) solves* (2.9) *and its dual problem if and only if condition C0 holds.*

Condition C0 is satisfied trivially by SP-TCP when $|s| = 1$. For MP-TCP ($|s| > 1$), the models derived in Section 2.1.2 show that only EWTCP and Coupled algorithms satisfy C0 and have underlying utility functions. It therefore follows from the theory for SP-TCP that EWTCP has a unique stable equilibrium while Coupled algorithm may have multiple equilibria since its corresponding utility function is not strictly concave. The other MP-TCP algorithms all have asymmetric Jacobian $\frac{\partial \Phi_s}{\partial \mathbf{x}_s}$ and do not satisfy C0.

2.2.3 Existence, uniqueness and stability of equilibrium

Even though a multipath TCP algorithm (K, Φ) may not have a utility maximization interpretation, a unique equilibrium exists if conditions C1–C3 are satisfied.

Theorem 2.2 (existence and uniqueness)**.** *1. Suppose C1 holds. Then (2.3)–(2.4) has at least one equilibrium.*

2. Suppose C2 and C3 hold. Then (2.3)–(2.4) has at most one equilibrium

Thus (2.3)–(2.4) has a unique equilibrium $(\mathbf{x}^*, \mathbf{p}^*)$ *under C1–C3.*

Conditions C1-C3 not only guarantee the existence and uniqueness of the equilibrium, they also ensure that the equilibrium is globally asymptotically stable, when the gain $k_r(\mathbf{x}_s)$ is only a function of x_r itself, i.e., $k_r(\mathbf{x}_s) \equiv k_r(x_r)$ for all $r \in R$. This is satisfied by all the existing algorithms presented in Section 2.1.2.

Theorem 2.3 (stability)**.** *Suppose C1-C3 hold and* $k_r(\mathbf{x}_s) \equiv k_r(x_r)$ *for all* $r \in R$*. Then the unique equilibrium* $(\mathbf{x}^*, \mathbf{p}^*)$ *is globally asymptotically stable. In particular, starting from any initial point* $\mathbf{x}(0) \in \mathbb{R}_+^{|R|}$ *and* $\mathbf{p}(0) \in \mathbb{R}_+^{|L|}$*, the trajectory* $(\mathbf{x}(t), \mathbf{p}(t))$ *generated by the MP-TCP algorithm (2.3)–(2.4) converges to the equilibrium* $(\mathbf{x}^*, \mathbf{p}^*)$ *as* $t \to \infty$*.*

Our proposed algorithm does not satisfy $k_r(\mathbf{x}_s) \equiv k_r(x_r)$ even though it seems to be stable in our experiments. This condition is only sufficient and needed in our Lyapunov stability proof; see Appendix 2.C. When $k_r(\mathbf{x}_s)$ depends on $\mathbf{x}_s$, one can replace $k_r(x_r)$ in the definition of the Lyapunov function V in (2.21) with $k_r(\mathbf{x}_s^*)$ evaluated at the equilibrium and the same argument there proves that $(\mathbf{x}^*, \mathbf{p}^*)$ is (locally) asymptotically stable. Also see Theorem 2.5 below for an alternative proof of local stability.

2.2.4 TCP friendliness

Informally, an MP-TCP flow is said to be 'TCP friendly' if it does not dominate the available bandwidth when it shares the same network with a SP-TCP flow [30]. To define this precisely we use the test network shared by a SP-TCP flow and a MP-TCP flow under test as shown in Fig. 2.1.

All paths traverse a single bottleneck link with capacity c, with all other links with capacities strictly higher than c. The links have fixed but possibly different delays. To compare the friendliness of two MP-TCP algorithms $\hat{M} := (\hat{K}, \hat{\Phi})$ and $\tilde{M} := (\tilde{K}, \tilde{\Phi})$, suppose that when $\hat{M}$ shares the test network with a SP-TCP it achieves a throughput of $\|\hat{\mathbf{x}}\|_1$ in equilibrium aggregated over the available paths (the SP-TCP therefore attains a throughput of $c - \|\hat{\mathbf{x}}\|_1$). Suppose $\tilde{M}$ achieves a throughput of $\|\tilde{\mathbf{x}}\|_1$ in equilibrium when it shares the test network with the same SP-TCP. Then we say that $\hat{M}$ is *friendlier (or more TCP-friendly)* than $\tilde{M}$ if $\|\hat{\mathbf{x}}\|_1 \leq \|\tilde{\mathbf{x}}\|_1$, i.e., if $\hat{M}$ receives no more bandwidth than $\tilde{M}$ does when they *separately* share the test network in Fig. 2.1 with the same SP-TCP flow.

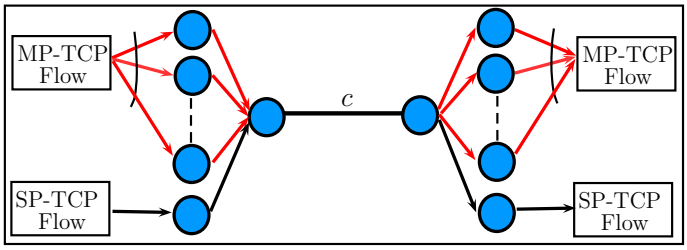


Figure 2.1: Test network for the definition of TCP friendliness. The link in the middle is the only bottleneck link with capacity c.

From the theory for single-path TCP ($|s| = 1$ for all $s \in S$), it is known that a design is more TCP-friendly if it has a smaller marginal utility $U_s'(x_s) = \Phi_s(x_s)$. The same intuition holds for MP-TCP algorithms even though the utility functions may not exist for MP-TCP algorithm.

Theorem 2.4 (friendliness)**.** *Consider two MP-TCP algorithms $\hat{M} := (\hat{K}, \hat{\Phi})$ and $\tilde{M} := (\tilde{K}, \tilde{\Phi})$. Suppose both satisfy C1–C4. Then $\hat{M}$ is friendlier than $\tilde{M}$ if $\hat{\Phi}_s(\mathbf{x}_s) \leq \tilde{\Phi}_s(\mathbf{x}_s)$ for all $s \in S$.*

2.2.5 Responsiveness around equilibrium

Suppose conditions C1–C3 hold and there is a unique equilibrium $\mathbf{z}^* := (\mathbf{x}^*, \mathbf{p}^*)$. Assume all links in L are active with $p_l^* > 0$; otherwise remove from L all links with prices $p_l^* = 0$. Let $\delta\mathbf{z}(t) := \mathbf{z}(t) - \mathbf{z}^*$. The behavior of (2.3)–(2.4) around the equilibrium is defined by the linearized system:

$$\delta\dot{\mathbf{z}} \quad = \quad J^* \; \delta\mathbf{z}(t). \tag{2.10}$$

Here J^* is the Jacobian of (2.3)–(2.4) at the equilibrium $\mathbf{z}^*$:

$$J^* \;:=\; J(\mathbf{x}^*) \quad := \quad \begin{bmatrix} \Lambda_k \frac{\partial \Phi}{\partial \mathbf{x}} & -\Lambda_k H^T \\ \Lambda_\gamma H & 0 \end{bmatrix},$$

where $\Lambda_k = diag\{k_r(\mathbf{x}_s^*), r \in R\}$, $\Lambda_\gamma = diag\{\gamma_l, l \in L\}$, and $\frac{\partial \Phi}{\partial \mathbf{x}}$ is evaluated at $\mathbf{x}^*$.

The stability and responsiveness of the linearized system (2.10) (how fast does the system converges to the equilibrium locally) is determined by the real parts of the eigenvalues of J^*. Specifically the linearized system is stable if the real parts of all eigenvalues of J^* are negative; moreover the more negative the real parts are the faster the linearized system converges to the equilibrium. We now show that the linearized system (2.10) is stable (i.e., converges exponentially fast to $\mathbf{z}^*$ locally) and characterize its responsiveness in terms of the design choices (K, Φ).

Let $Z = \{\mathbf{z} := (\mathbf{x}, \mathbf{p}) \in \mathbb{C}^{|R|+|L|} \mid \|\mathbf{z}\|_2 = 1\}$.

Theorem 2.5 (responsiveness)**.** *Suppose C1–C3 hold. Then*

1. *The linearized system* (2.10) *is stable, i.e.,* $\mathbf{Re}(\lambda) < 0$ *for any eigenvalue* λ *of* J^*. *Moreover* $\mathbf{Re}(\lambda) \le \overline{\lambda}(J^*)$, *where*

$$\overline{\lambda}(J^*) \;:=\; \max_{\mathbf{z}\in Z}\left\{\frac{\mathbf{x}^H\left[\frac{\partial\Phi}{\partial\mathbf{x}}\right]^+\mathbf{x}}{\mathbf{x}^H\Lambda_k^{-1}\mathbf{x}+\mathbf{p}^H\Lambda_\gamma^{-1}\mathbf{p}}\right\} \;\le\; 0,$$

where Λ_k *and* $\frac{\partial\Phi_s}{\partial\mathbf{x}_s}$ *are evaluated at the equilibrium point* z^*.

2. *For two MP-TCP algorithms* $(\hat{K},\hat{\Phi})$ *and* $(\tilde{K},\tilde{\Phi})$, $\overline{\lambda}(\hat{J}^*) \le \overline{\lambda}(\tilde{J}^*)$ *provided*

$$\hat{K}_s \;\ge\; \tilde{K}_s \quad \textit{and} \quad \frac{\partial\hat{\Phi}_s}{\partial\mathbf{x}_s} \preceq \frac{\partial\tilde{\Phi}_s}{\partial\mathbf{x}_s} \qquad \textit{for all } s\in S.$$

Theorem 2.5 motivates the following definition of responsiveness. Given two MP-TCP $\hat{M}$ and $\tilde{M}$, we say that $\hat{M}$ is *more responsive than* $\tilde{M}$ if $\overline{\lambda}(\hat{J}^*) \le \overline{\lambda}(\tilde{J}^*)$. Theorem 2.5(2) implies that an MP-TCP algorithm with a larger $K_s(\mathbf{x}_s^*)$ or more negative definite $\left[\frac{\partial\Phi_s}{\partial\mathbf{x}_s}(\mathbf{x}_s^*)\right]^+$ is more responsive, in the sense that the real parts of the eigenvalues of the Jacobian J^* have a smaller more negative upper bound.

Then the next result suggests an inevitable tradeoff between responsiveness and friendliness.

Theorem 2.6 (tradeoff)**.** *Consider two MP-TCP algorithms* $(K,\hat{\Phi})$ *and* $(K,\tilde{\Phi})$ *with the same gain* K. *Suppose both satisfy C1-C3 and C5. Then for all* $s\in S$

$$\frac{\partial\hat{\Phi}_s(\mathbf{x}_s)}{\partial\mathbf{x}_s} \;\preceq\; \frac{\partial\tilde{\Phi}_s(\mathbf{x}_s)}{\partial\mathbf{x}_s} \quad\Rightarrow\quad \hat{\Phi}_s(\mathbf{x}_s) \;\ge\; \tilde{\Phi}_s(\mathbf{x}_s).$$

In light of Theorems 2.4 and 2.5, Theorem 2.6 says that a more responsive MP-TCP design is inevitably less friendly if they have the same K.

The theorem is easier to understand in the case of SP-TCP, i.e., when $|s|=1$ for all $s\in S$ and $\Phi_s(x_s) = U_s'(x_s)$. Then it implies that a more concave utility function $U_s(x_s)$ has a larger marginal utility, and is hence less friendly.

2.2.6 Window oscillation

Window oscillations are inherent in loss-based additive increase multiplicative decrease (AIMD) TCP algorithms. We close this section by discussing informally why a larger design $K_s(\mathbf{x}_s)$ generally creates more severe window oscillations. This implies a tradeoff between responsiveness (which is enhanced by a large $K_s(\mathbf{x}_s)$) and oscillation (which is reduced with a small $K_s(\mathbf{x}_s)$).

The effect of $K_s(\mathbf{x}_s)$ on window fluctuations can be understood by studying how it affects the decrease $D_r(\mathbf{w}_s)$ per packet loss in the following packet level model:

- For each ACK on route r, $w_r \leftarrow w_r + I_r(\mathbf{w}_s)$.

- For each loss on route r, $w_r \leftarrow w_r - D_r(\mathbf{w}_s)$.

Let $Z_r \in \{0,1\}$ be an indicator variable of whether a packet loss is observed on route r at an arbitrary time in steady state. Then

$$D_s(\mathbf{x}_s) \;:=\; \frac{1}{\|\mathbf{x}_s\|_1}\,\mathbb{E}\left(\sum_{r\in s}\frac{D_r(\mathbf{w}_s)}{\tau_r}Z_r\,\middle|\,\sum_{k\in s}Z_k\ge 1\right).$$

represents the expected relative reduction in *aggregate* throughput $\sum_{r\in s} D_r(\mathbf{w}_s)/\tau_r$, given that there is at least one packet loss on some route $r \in s$. It is a measure of throughput fluctuation for each packet loss that an application experiences. For TCP-NewReno (for which $s = \{r\}$ and w_s is a scalar), the window size is halved on each packet loss, $D_r(w_s) = w_r/2$, and hence $D_s(x_s) = 1/2$.

To understand $D_s(\mathbf{x}_s)$ for MP-TCP algorithms, we need the following result.

Lemma 2.7. *Let $A_i := \{a_{i1}, a_{i2}, \ldots\}$ with $|A_i|$ elements. Each element a_{ij} is an independent binary random variable with $\mathbb{P}(a_{ij} = 1) = 1 - \mathbb{P}(a_{ij} = 0) = q_i$. Define $D_i(A_i) := d_i 1_{(\sum_j a_{ij}\ge 1)}$. Then*

$$\mathbb{E}\left(\sum_k D_k(A_k)\,\middle|\,\sum_{i,j}a_{ij}\ge 1\right) \;=\; \frac{\sum_k d_k q_k |A_k|}{\sum_k q_k|A_k|} + o\left(\sum_k q_k\right).$$

Suppose each route has a fixed loss probability q_r. Then within each RTT, Lemma 2.7 implies

$$D_s(\mathbf{x}_s) \;=\; \frac{1}{\|\mathbf{x}_s\|_1}\left(\frac{\sum_{r\in s} w_r q_r D_r(\mathbf{w}_s)/\tau_r}{\sum_{r\in s} q_r w_r} + o\left(\sum_{r\in s} q_r\right)\right).$$

Substituting $w_r = x_r\tau_r$ and $x_r D_r(\mathbf{w}_s) = \tau_r k_r(\mathbf{x}_s)$ from (2.5), we get, ignoring the high-order terms,

$$D_s(\mathbf{x}_s) \;=\; \frac{1}{\|\mathbf{x}_s\|_1}\left(\frac{\sum_{r\in s}\tau_r q_r k_r(\mathbf{x}_s)}{\sum_{r\in s}\tau_r q_r x_r}\right). \tag{2.11}$$

to the first order. Note that $k_r(\mathbf{x}_s)$ does not affect the *equilibrium* rates $\mathbf{x}_s$. Hence, with the assumption that τ_r are constants, $D_s(\mathbf{x}_s)$ is determined by the functions $k_r(\mathbf{x}_s)$ in steady state.

Specifically an MP-TCP algorithm with a larger $K_s(\mathbf{x}_s)$ tends to have a larger $D_s(\mathbf{x}_s)$ and hence more severe window oscillations. Theorem 2.5, however, suggests that a larger $K_s(\mathbf{x}_s)$ also leads to better responsiveness, suggesting an inevitable tradeoff between responsiveness and window oscillation.

2.3 Implications and a new algorithm

In this section we discuss the implications of these structural properties on the behavior of existing MP-TCP algorithms. They are further illustrated in the experiment results in Section 2.4. The

discussion motivates a new design that generalizes the existing MP-TCP algorithm.

2.3.1 Implications on existing algorithms

Recall Table 2.1 that summarizes the conditions satisfied by the various algorithms. Only EWTCP and Coupled algorithms satisfy C0. Their equilibrium properties can be studied in the standard utility maximization model as done for single-path TCP. Semicoupled and Max algorithms do not satisfy C0 and therefore analysis through utility maximization is not applicable. However, Theorem 2.8 below implies that both Semicoupled and Max algorithms satisfy C1–C3 provided they enable no more than 8 routes. Theorem 2.2 and 2.3 then imply that they have a unique and globally stable equilibrium. It is also easy to show that EWTCP satisfies C1-C3. The Coupled algorithm does not satisfy C2 and is found to have multiple equilibria in [46].

Next we discuss friendliness of existing MP-TCP algorithms. It can be shown that the $\phi_r(\mathbf{x}_s)$ corresponding to these algorithms satisfy:

$$\phi_r^{ewtcp}(\mathbf{x}_s) \geq \phi_r^{semicoupled}(\mathbf{x}_s) \geq \phi_r^{max}(\mathbf{x}_s) \geq \phi_r^{coupled}(\mathbf{x}_s)$$

for all $\mathbf{x}_s \geq 0$ if all routes $r \in s$ have the same round trip time. Since all of them satisfy C4, Theorem 2.4 implies that their friendliness will be in the same order, i.e., their throughputs in the test network of Fig. 2.1 are ordered as follows:

$$\text{EWTCP}(a \geq 1)^1 \geq \text{Semicoupled} \geq \text{Max} \geq \text{Coupled}.$$

This is confirmed by the Linux-based experiment.

Third we will discuss responsiveness of existing MP-TCP algorithms. These algorithms have the same gain function $k_r(\mathbf{x}_s) = 0.5x_r^2$ and

$$(\frac{\partial \Phi_s}{\partial \mathbf{x}_s})^{ewtcp} \preceq (\frac{\partial \Phi_s}{\partial \mathbf{x}_s})^{semicoupled} \preceq (\frac{\partial \Phi_s}{\partial \mathbf{x}_s})^{max} \preceq (\frac{\partial \Phi_s}{\partial \mathbf{x}_s})^{coupled}.$$

Theorem 2.5 then implies that their responsiveness should be in the same order, as confirmed by our experiments in section 2.4.

Finally we discuss window oscillation of existing MP-TCP algorithms using $D_s(\mathbf{x}_s)$ as the metric. As mentioned in Section 2.2.6, $D_s(\mathbf{x}_s) = 0.5$ for TCP NewReno, a benchmark single-path TCP algorithm. According to (2.11), if $k_r(\mathbf{x}_s) \leq 0.5x_r\|\mathbf{x}_s\|_1$, we have, to the first order

$$D_s(\mathbf{x}_s) \quad \leq \quad \frac{1}{2}\frac{\sum_{r\in s}\tau_r q_r x_r \|\mathbf{x}_s\|_1}{\|\mathbf{x}_s\|_1 \sum_{r\in s}\tau_r q_r x_r} \quad = \quad \frac{1}{2}.$$

[1] When $a < 1$, the MP-TCP source can obtain even smaller throughput than the competing single-path TCP source.

All existing MP-TCP algorithms have the same $k_r(\mathbf{x}_s) = 0.5x_r^2 \leq 0.5x_r\|\mathbf{x}_s\|_1$, with strict inequality if $|s| > 1$ and $x_r > 0$ for at least two $r \in s$. Thus enabling MP-TCP always tends to reduce window oscillation for existing algorithms compared to TCP NewReno. Moreover, the window oscillation is always reduced compared to TCP NewReno when $k_r(\mathbf{x}_s) \leq 0.5x_r\|\mathbf{x}_s\|_1$.

2.3.2 A generalized algorithm

Consider the class of algorithms parametrized by (β, n, η) as follows:

$$\begin{cases} k_r(\mathbf{x}_s) & = \frac{1}{2}x_r(x_r + \eta(\|\mathbf{x}_s\|_\infty - x_r)), \quad \eta \geq 0 \\ \phi_r(\mathbf{x}_s) & = \frac{2((1-\beta)x_r + \beta\|\mathbf{x}_s\|_n)}{\tau_r^2 x_r \|\mathbf{x}_s\|_1^2}, \quad n \in \mathbb{N}_+, \beta \geq 0 \end{cases} . \tag{2.12}$$

This class includes the Max ($\beta = 1, \eta = 0, n = \infty$), Coupled ($\beta = 0, \eta = 0$), and Semicoupled ($\beta = 1, \eta = 0, n = 1$) algorithms as special cases when all RTTs on different paths of the same source are the same, i.e., $\tau_r = \tau_s$, $r \in s$.

The next result characterizes a subclass that have a unique and locally stable equilibrium point.

Theorem 2.8. *Fix any $\eta \geq 0$ and $n \in \mathbb{N}_+$. For any $s \in S$, the $\phi_r(\mathbf{x}_s)$ in (2.12) satisfies*

1. *C1 if $\beta \geq 0$.*
2. *C2–C3 if $0 < \beta \leq 1$, $|s| \leq 8$ and τ_r are the same for all $r \in s$ (assuming H has full row rank).*

The requirement that $|s| \leq 8$ is not restrictive since in practice a device may typically enable no more than 3 paths. The requirement that τ_r are the same for all $r \in s$ is used in proving the negative definiteness of the (symmetric part of the) Jacobian of $\Phi_s(\mathbf{x}_s)$. Since a negative definite matrix remains negative definite after small enough perturbations of its entries, Theorem 2.8 holds if the RTTs of the subpaths do not differ much. This (sufficient) condition seems reasonable as two paths between the same source-destination pair often have similar RTTs if both are wireline paths. Note that our experiments in chapter 2.4 show that the algorithm also converges even if the RTTs on different paths differ dramatically, e.g. the RTT of WiFi is usually much smaller than that of 3G.

For the class of algorithms specified by (2.12), Theorem 2.8 motivates a design space defined by $\beta \in (0, 1], \eta \geq 0, n \in \mathbb{N}_+$, where β and n control the tradeoff between friendliness and responsiveness and η controls the tradeoff between responsiveness and window oscillation. In Table 2.2, we summarize how the parameters (β, η, n) affect the performance.

We now describe our design philosophy. As discussed above the design of MP-TCP algorithms involves inevitable tradeoffs among responsiveness, friendliness, and the severity of window oscillation. Specifically a design is more responsive if it has a higher gain K_s or a more negative definite

Table 2.2: How design choices affect MP-TCP performance.

Performance	**Parameter**	**Parameters in** (2.12)
TCP friendliness	$\phi_r(\mathbf{x}_s) \downarrow$	$\beta \downarrow$, $n \uparrow$
Responsiveness	$k_r(\mathbf{x}_s) \uparrow$, $-\partial\Phi_s/\partial\mathbf{x}_s \uparrow$	$\beta \uparrow$, $n \downarrow$, $\eta \uparrow$
Window oscillation	$k_r(\mathbf{x}_s) \downarrow$	$\eta \downarrow$

Jacobian $[\partial\Phi_s/\partial\mathbf{x}_s]^+$ (Theorem 2.5). However, a larger K_s usually creates a bigger window oscillation; a more negative definite $[\partial\Phi_s/\partial\mathbf{x}_s]^+$ implies a larger Φ_s, usually hurting friendliness (Theorems 2.6 and 2.4). This is summarized in Table 2.2. Since enabling multiple paths already reduces window oscillation compared to single-path TCP (section 2.3.1), MP-TCP can afford to use a relatively large gain K_s for responsiveness. This does not compromise too much on window oscillation, but allows us to use a less negative definite Jacobian $[\partial\Phi_s/\partial\mathbf{x}_s]^+$ with a smaller Φ_s to maintain sufficient TCP friendliness. Moreover, responsiveness is mainly affected by subpaths with small throughput while window oscillation is mainly affected by subpaths with large throughput. The parameter η in the generalized algorithm (2.12) scales $k_r(\mathbf{x}_s)$ in the right way: a path r that has a large x_r has $k_r(\mathbf{x}_s) \approx 0.5x_r^2$ and hence a similar degree of window oscillation as existing algorithms, while a path r with a small x_r has larger $k_r(\mathbf{x}_s)$ than that under a design with zero η and therefore is more responsive.

Our experiments show that Max algorithm ($(\beta, \eta, n) = (1, 0, \infty)$) overtakes too much of the competing single-path TCP flows. Therefore, we can only use a smaller β since n is already infinite in order to improve friendliness. To compensate for the responsiveness performance, we will use a larger η, which will sacrifice window oscillation performance. The *Balia* MP-TCP algorithm given at the beginning of this chapter corresponds to the choice $(\beta, \eta, n) = (0.2, 0.5, \infty)$. Instead of allowing the window size to drop to 1 for a packet loss, we add a cap for the decrement of window size, which improves the performance as confirmed in our experiments. Note that there is no "best" parameter setting since there are tradeoffs among all the performance metrics and we choose $(\beta, \eta, n) = (0.2, 0.5, \infty)$ based on our experiments in chapter 2.4, which show that this parameter setting strikes a good balance among responsiveness, friendliness, and window oscillation.

2.4 Experiment

In this section we summarize our experimental results that illustrate the above analysis. In addition to the MP-TCP algorithms illustrated in chapter 2.1.2, we also include the recently developed OLIA MP-TCP algorithm [50]. We evaluate the MP-TCP algorithms using a reference Linux implementation of MP-TCP, Multipath TCP v0.88 [66]. Since it currently includes only Max and OLIA algorithms, we implement EWTCP, Semicoupled, Coupled, and the proposed Balia algorithm in the reference implementation. For the Coupled and our algorithm, the minimum *ssthresh* is set to 1

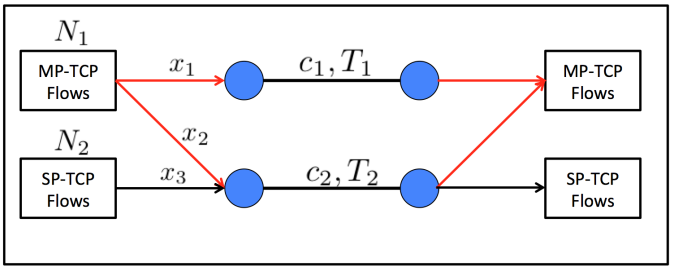


Figure 2.2: Network for our Linux-based experiments on TCP friendliness and responsiveness, with N_1 MP-TCP flows and N_2 single-path TCP flows sharing two links of capacity, c_1, c_2, and propagation delay (single trip) T_1, T_2. MP-TCP flows maintain two routes with rate x_1, x_2. Single-path TCP flows maintain one route with rate x_3.

instead of 2 when more than 1 path is available.

The network topology is shown in Fig. 2.2. In the testbed, all nodes are Linux machines with a quad-core Intel i5 3.33GHz processor, 4GB RAM, and multiple 1Gbps Ethernet interfaces, running Ubuntu 13.10 (Linux kernel 3.11.8). The network parameters such as c_1, c_2, T_1, and T_2 are controlled by Dummynet [13].

Our experiments are divided into three parts. First we compare TCP friendliness of Balia algorithm and prior algorithms. The result confirms that the Couple algorithm is the friendliest, and that the Balia algorithm is close to the Coupled algorithm and friendlier than the other algorithms. Second, we compare the responsiveness of each algorithm in a dynamic environment where flows come and go. The result shows that the Coupled and OLIA algorithms are unresponsive (illustrating the tradeoff between responsiveness and friendliness). EWTCP is the most responsive; Balia is similar in responsiveness but friendlier to single-path TCP flows. Finally we show that all MP-TCP algorithms have smaller average window oscillations than single-path TCP.

These experiments confirm our analytical results and suggest our design choice strikes a good balance among friendliness, responsiveness, and window oscillation.

2.4.1 TCP friendliness

We study TCP friendliness of each algorithm, first with paths of similar RTTs and then with paths of different RTTs, which emulates the wireless scenario. We assume all the flows are long lived and focus on the steady state throughput.

In the first set of experiments, we let $T_1 = T_2 = 5$ms, $c_1 = c_2 = 60$Mbps and $N_1 = N_2 = 30$. We repeat the experiments 20 times, and the average aggregate throughput of MP-TCP and single-path TCP users and the 95% margin of error for *confidence interval* (CI) are shown in Table 2.3. The Coupled algorithm is the friendliest and the Balia algorithm is closer to the Coupled algorithm than the others.

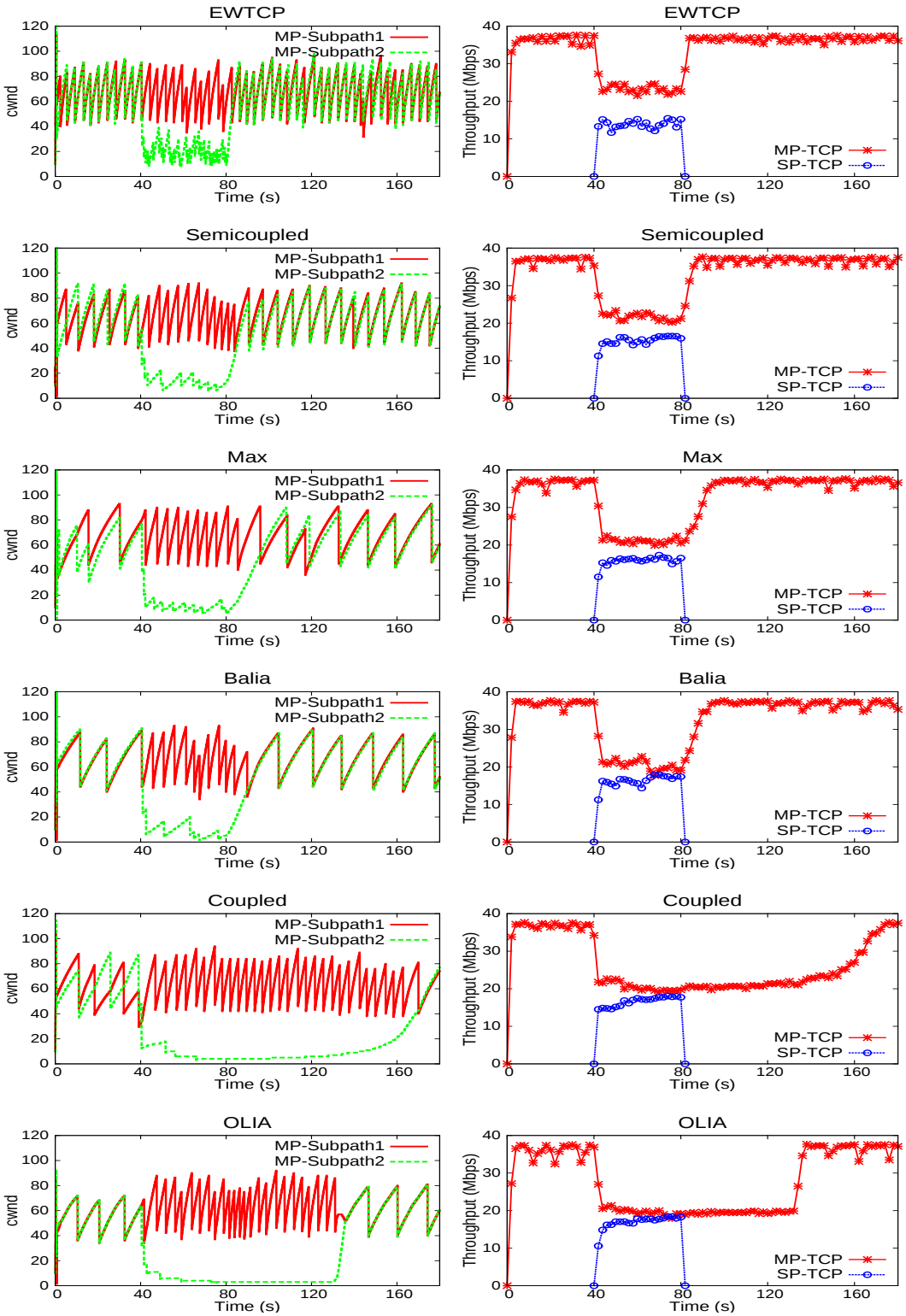


Figure 2.3: Responsiveness Performance: congestion window trajectory of MP-TCP for each path (left column). SP-TCP starts at time 40s and ends at 80s. The throughput of SP-TCP and total throughput of MP-TCP are shown in the right column. Parameters: $T_1 = T_2 = 10$ms, $c_1 = c_2 = 20$Mbps, and $N_1 = 1, N_2 = 5$.

Table 2.3: TCP friendliness (same RTTs): Average throughput (Mbps) and 95% confidence interval of MP-TCP and single-path TCP users. ($T_1 = T_2 = 5$ms, $c_1 = c_2 = 60$Mbps and $N_1 = N_2 = 30$)

	ewtcp	semi.	max	balia	coupled	olia
mp-tcp (throuput)	2.75	2.65	2.60	2.52	2.44	2.61
mp-tcp (CI)	0.005	0.004	0.005	0.006	0.005	0.004
sp-tcp (throuput)	0.951	1.07	1.13	1.22	1.29	1.12
sp-tcp (CI)	0.005	0.007	0.008	0.006	0.005	0.004

In the second set of experiments, we assume a highly heterogeneous RTTs by emulating the scenario of a mobile device with both 3G and WiFi access. WiFi access usually has higher capacity and lower delay compared to 3G. Specificially, we set $T_1 = 10$ms, $c_1 = 8$Mbps for the first link to emulate WiFi access and $T_2 = 100$ms, $c_2 = 2$Mbps for the second link to emulate 3G access. When there exists single-path TCP flows, i.e. $N_2 > 0$, the behaviors of all the algorithms are similar to the equal RTT case in the first set of simulation. The Coupled algorithm is the friendliest and the Balia algorithm is closer than other algorithms. However, when there is no single-path TCP flow, i.e. $N_1 = 1$ and $N_2 = 0$, the performance of OLIA is not stable enough to effectively take all the available capacity while the other algorithms do not have such problem. We repeat the experiments 20 times and we find that sometimes OLIA does not use the 3G access link. The average throughput of MP-TCP user and the 95% margin of error for confidence interval is shown in Table 2.4.

Table 2.4: Basic behavior (WiFi/3G): throughput (Mbps) of a MP-TCP user and 95% confidence interval. ($T_1 = 10$ms, $T_2 = 100$ms, $c_1 = 8$Mbps, $c_2 = 2$Mbps, $N_1 = 1, N_2 = 0$)

	ewtcp	semi.	max	balia	coupled	olia
throughput	9.26	9.27	9.26	9.27	9.28	9.19
confidence interval	0.008	0.006	0.006	0.01	0.01	0.09

2.4.2 Responsiveness

We use the network in Fig. 2.2 with $c_1 = c_2 = 20$Mbps, $T_1 = T_2 = 10$ms and $N_1 = 1, N_2 = 5$. To demonstrate the dynamic performance of each algorithm, we assume the MP-TCP flow is long lived while the single-path TCP flows start at 40s and end at 80s. We record the aggregate throughput of the single-path TCP flows from 40-80s, which measures the friendliness of MP-TCP. We also measure the time for the congestion window on the second path to recover[2] of MP-TCP users. It measures the responsiveness of MP-TCP. These measurements are shown in Table 2.5 and the congestion window and throughput trajectories of all algorithms are shown in Fig. 2.3. To clearly show the

[2]Defined as the first time the congestion window on the second path reaches the average congestion window (e.g., 60) after the single-path users have left.

responsiveness performance, we record the longest convergence time found in our experiment in Table 2.5 and the corresponding trajectories are shown in Fig. 2.3.

Table 2.5: Responsiveness: convergence time (s) of MP-TCP and total throughput (Mbps) of all single-path TCP users. ($T_1 = T_2 = 10$ms, $c_1 = c_2 = 20$Mbps, $N_1 = 1, N_2 = 5$)

	ewtcp	semi.	max	balia	coupled	olia
Convergence	3.25	7.46	17.75	14.73	94.36	58.5
SP-TCP	13.89	15.35	15.8	16.28	16.64	16.97

EWTCP is the most responsive among all the algorithms. Ours is as responsive as the Max algorithm, yet significantly friendlier than EWTCP. Both Coupled and OLIA algorithms take an excessively long time to recover. For Coupled algorithm, the excessively slow recovery of the congestion window on the second path (see Fig. 2.3) is due to the design that increases the window roughly by $w_r/(\sum_{k\in s} w_k)^2$ on each ACK, assuming the RTTs are similar. After the single-path TCP flow has left, w_2 is small while w_1 is large, so that $w_2/(w_1+w_2)^2$ is very small. It therefore takes a long time for w_2 to increase to its steady state value. In general, under the Coupled algorithm, a route with a large throughput can greatly suppress the throughput on another route even though the other route is underutilized. The reason of the poor responsiveness performance of OLIA can be explained using a similar argument to the Coupled algorithm since they have the same increment/decrement for each ACK/loss in this scenario.

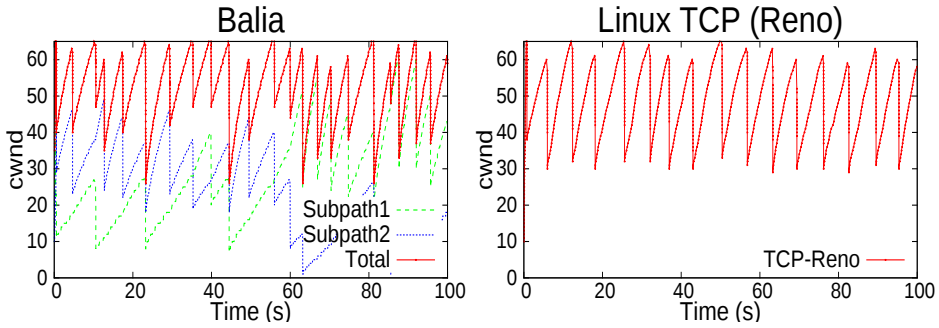


Figure 2.4: Window oscillation: the red trajectories represent throughput fluctuations experienced by the application in the case of MP-TCP and the case of single-path TCP.

2.4.3 Window oscillation

We use a single-link network model to compare window oscillation under MP-TCP and single-path TCP. First a MP-TCP flow initiates two subpaths through that link, and we measure the window size of each subpath and their aggregate window size. Then a TCP-Reno flow traverses the same link and we measure its window size. The results are shown in Fig. 2.4 for our algorithm in comparison with single-path TCP (other MP-TCP algorithms have a similar behavior). They confirm that

enabling multiple paths reduces the average window oscillation compared with only using a single path.

2.5 Conclusion

We have presented a model for MP-TCP and identified design criteria that guarantee the existence, uniqueness, and stability of the network equilibrium. We have characterized the design space and study the tradeoff among TCP friendliness, responsiveness, and window oscillation. We have proposed the Balia MP-TCP algorithm that generalizes existing algorithms and strikes a good balance among these properties. We have implemented the Balia in the Linux kernel and used it to evaluate the performance of our algorithm.

Appendix

2.A Proof of Theorem 2.1 (utility maximization)

The Lagrangian of (2.9) is:

$$\begin{aligned} L(\mathbf{x}, \mathbf{p}) &= \sum_{s \in S} U_s(\mathbf{x}_s) - \sum_{l \in L} p_l (y_l - c_l) \\ &= \sum_{s \in S} U_s(\mathbf{x}_s) - \sum_{l \in L} p_l (\sum_{r \in R} H_{lr} x_r - c_l) \\ &= \sum_{s \in S} \left(U_s(\mathbf{x}_s) - \sum_{r \in s} x_r q_r \right) + \sum_{l \in L} p_l c_l, \end{aligned}$$

where $\mathbf{p} \geq \mathbf{0}$ are the dual variables and $q_r := \sum_{r \in R} H_{lr} p_l$. Then the dual problem is

$$D(\mathbf{p}) = \sum_{s \in S} \max_{\mathbf{x}_s \geq 0} \{ B_s(\mathbf{x}_s, \mathbf{p}) \} + \sum_{l \in L} p_l c_l \quad \mathbf{p} \geq \mathbf{0},$$

where $B_s(\mathbf{x}_s, \mathbf{p}) = U_s(\mathbf{x}_s) - \sum_{r \in s} x_r q_r$. The KKT condition implies that, at optimality, we have

$$\frac{\partial U_s(\mathbf{x}_s)}{\partial x_r} < q_r \Rightarrow x_r = 0 \text{ and } x_r > 0 \Rightarrow \frac{\partial U_s(\mathbf{x}_s)}{\partial x_r} = q_r \tag{2.13}$$

$$y_l < c_l \Rightarrow p_l = 0 \text{ and } p_l > 0 \Rightarrow y_l = c_l. \tag{2.14}$$

Comparing with (2.6)–(2.7) we conclude that, if a MP-TCP algorithm defined by (2.3)–(2.4) has an underlying utility function U_s, then we must have

$$\frac{\partial U_s(\mathbf{x}_s)}{\partial x_r} = \phi_r(\mathbf{x}_s) \quad r \in s, \; x_r > 0. \tag{2.15}$$

Given $\phi_r(\mathbf{x}_s)$, (2.15) has a continuously differentiable solutions $U_s(\mathbf{x}_s)$ if and only if the Jacobian of $\Phi_s(\mathbf{x}_s)$ is symmetric, i.e., if and only if

$$\frac{\partial \Phi(\mathbf{x}_s)}{\partial \mathbf{x}_s} = \left[\frac{\partial \Phi(\mathbf{x}_s)}{\partial \mathbf{x}_s} \right]^T.$$

2.B Proof of Theorem 2.2 (existence and uniqueness)

2.B.1 Proof of part 1

For any link $l \in L$, let

$$\mathbf{p}_{-l} = \{p_1, \ldots, p_{l-1}, p_{l+1}, \ldots, p_{|L|}\},$$

whose component composes of all the elements in $\mathbf{p}$ except p_l. For $l \in L$, let

$$g_l(\mathbf{p}) := c_l - \sum_{r: l \in r} x_r = c_l - \sum_{s: r \in s, l \in r} y_l^s(p_l, \mathbf{p}_{-l})$$

and $h_l(\mathbf{p}) := -g_l^2(\mathbf{p})$. According to C1, we have the following two facts, which will be used in the proof.

- $g_l(\mathbf{p})$ is a nondecreasing function of p_l on $\mathbb{R}_+$ since $y_l^s(\mathbf{p})$ is a nonincreasing function of p_l.
- $\lim_{p_l \to \infty} g_l(p_l, \mathbf{p}_{-l}) = c_l$ since $\lim_{p_l \to \infty} y_l^s(\mathbf{p}) = 0$.

Next, we will show that $h_l(\mathbf{p})$ is a quasi-concave function of p_l. In other words, for any fixed $\mathbf{p}_{-l}$, the set $S_a := \{p_l \mid h_l(\mathbf{p}) \geq a\}$ is a convex set. If $g_l(0, \mathbf{p}_{-l}) \geq 0$, then

$$g_l(p_l, \mathbf{p}_{-l}) \geq g_l(0, \mathbf{p}_{-l}) \geq 0 \quad \forall p_l \geq 0,$$

which means $h_l(p_l, \mathbf{p}_{-l})$ is a nonincreasing function of p_l, and hence is a quasi-concave function of p_l and

$$\arg\max_{p_l} h_l(p_l, \mathbf{p}_{-l}) = 0. \tag{2.16}$$

On the other hand, if $g_l(0, \mathbf{p}_{-l}) < 0$, then there exists a $p_l^* > 0$ such that $g_l(p_l^*, \mathbf{p}_{-l}) = 0$ since $g_l(\cdot)$ is continuous and $\lim_{p_l \to \infty} g_l(p_l, \mathbf{p}_{-l}) = c_l > 0$. Note that $g_l(\mathbf{p})$ is a nondecreasing function of p_l, then $h_l(p_l, \mathbf{p}_{-l})$ is nondecreasing for $p_l \in [0, p_l^*]$ and nonincreasing for $p_l \in [p_l^*, \infty)$. Thus, $h_l(p_l, \mathbf{p}_{-l})$ is also a quasi-concave function of p_l in this case and

$$\max_{p_l} h_l(p_l, \mathbf{p}_{-l}) = 0. \tag{2.17}$$

By Nash theorem, if $h_l(p_l, \mathbf{p}_{-l})$ is a quasi-concave function of p_l for all $l \in L$ and $\mathbf{p}$ is in a bounded set, then there exists a $\mathbf{p}^\star \in \mathbb{R}_+^{|L|}$ such that

$$p_l^\star = \arg\max_{p_l \in \mathbb{R}_+} h_l(p_l, \mathbf{p}_{-l}^*).$$

According to (2.16) and (2.17), for any $l \in L$, either $p_l^* > 0$ or $g_l^*(\mathbf{p}^*) > 0$ but neither holds at any

time. Therefore $\mathbf{p}^*$ satisfies Eqn. (2.7). Since $\mathbf{q} = R^T\mathbf{p}$, there exists an $\mathbf{x}^*$ to (2.6). Thus there exists at least one solution $(\mathbf{x}, \mathbf{p})$ that satisfies (2.6) and (2.7).

2.B.2 Proof of part 2

Lemma 2.9. *Assume a function $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and $\left[\frac{\partial F}{\partial \mathbf{x}}(\mathbf{x})\right]^+$ is negative definite for all $\mathbf{x}$. Then for any $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathbb{R}^n$,*

$$(\mathbf{x}_1 - \mathbf{x}_2)^T(F(\mathbf{x}_1) - F(\mathbf{x}_2)) < 0.$$

Proof. Fix any $\mathbf{x}_1 \neq \mathbf{x}_2 \in \mathbb{R}^n$. Define $A(t) := F\left(t\mathbf{x}_1 + (1-t)\mathbf{x}_2\right)$. Since $\partial F/\partial \mathbf{x}$ is continuous, there exists a $\lambda < 0$ such that the eigenvalues of $[\partial F/\partial \mathbf{x}]^+ \leq \lambda$ over the compact set $\{t\mathbf{x}_1 + (1-t)\mathbf{x}_2 \mid 0 \leq t \leq 1\}$. Then

$$\begin{aligned}
& (\mathbf{x}_1 - \mathbf{x}_2)^T(F(\mathbf{x}_1) - F(\mathbf{x}_2)) \\
= & \int_0^1 (\mathbf{x}_1 - \mathbf{x}_2)^T \frac{dA}{dt}(\tau)\, d\tau \\
= & \int_0^1 (\mathbf{x}_1 - \mathbf{x}_2)^T \frac{\partial F}{\partial \mathbf{x}}\left(\tau\mathbf{x}_1 + (1-\tau)\mathbf{x}_2\right)(\mathbf{x}_1 - \mathbf{x}_2)\, d\tau \\
\leq & \lambda\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 < 0.
\end{aligned}$$

□

Lemma 2.10. *Suppose C3 holds. Then $x_r^* > 0$ at equilibrium for all $r \in R$.*

Proof. Suppose $x_r^* = 0$. Then $q_r^* \geq \phi_r(\mathbf{x}_r^*) = \infty$ by C3 and hence there is a link $l \in r$ with $p_l^* = \infty$. But then, for all paths $r' \ni l$, $q_{r'}^* = \infty$ and hence $x_{r'}^* = 0$ by C3. This implies $y_l^* = 0 < c_l$, and hence $p_l^* = 0$ by (2.7), contradicting $p_l^* = \infty$. □

Recall the vector notations that $\mathbf{x} := (\mathbf{x}_s, s \in S) := (x_r, r \in s, s \in S)$ and $\Phi(\mathbf{x}) := (\Phi_s(\mathbf{x}_s), s \in S) := (\Phi_r(\mathbf{x}_s), r \in s, s \in S)$. To prove uniqueness of the equilibrium, suppose for the sake of contradiction that there are two distinct equilibrium points $(\mathbf{x}, \mathbf{p})$ and $(\hat{\mathbf{x}}, \hat{\mathbf{p}})$. By Lemma 2.10 we have $\mathbf{x} > 0$ and $\hat{\mathbf{x}} > 0$. Thus (2.6) implies $\Phi(\mathbf{x}) = \mathbf{q} = H^T\mathbf{p}$ and $\Phi(\hat{\mathbf{x}}) = \hat{\mathbf{q}} = H^T\hat{\mathbf{p}}$. By Lemma 2.9 and assumption C2 we then have

$$\begin{aligned}
0 > & (\mathbf{x} - \hat{\mathbf{x}})^T(\Phi(\mathbf{x}) - \Phi(\hat{\mathbf{x}})) \\
= & (\mathbf{x} - \hat{\mathbf{x}})^T H^T(\mathbf{p} - \hat{\mathbf{p}}) \\
= & (\mathbf{p} - \hat{\mathbf{p}})^T(\mathbf{y} - \hat{\mathbf{y}}).
\end{aligned}$$

Thus

$$\mathbf{p}^T\mathbf{y} + \hat{\mathbf{p}}^T\hat{\mathbf{y}} \;<\; \mathbf{p}^T\hat{\mathbf{y}} + \hat{\mathbf{p}}^T\mathbf{y}. \tag{2.18}$$

Equilibrium condition (2.7) implies

$$\mathbf{p}^T(\mathbf{c}-\mathbf{y}) \;=\; 0 \quad \text{and} \quad \hat{\mathbf{p}}^T(\mathbf{c}-\hat{\mathbf{y}}) \;=\; 0 \tag{2.19}$$

$$\mathbf{y} \;\leq\; \mathbf{c} \quad \text{and} \quad \hat{\mathbf{y}} \;\leq\; \mathbf{c}. \tag{2.20}$$

Substituting (2.19) into (2.18) yields

$$\begin{aligned} \mathbf{p}^T\mathbf{c} + \hat{\mathbf{p}}^T\mathbf{c} &< \mathbf{p}^T\hat{\mathbf{y}} + \hat{\mathbf{p}}^T\mathbf{y} \\ \mathbf{p}^T(\mathbf{c}-\hat{\mathbf{y}}) + \hat{\mathbf{p}}^T(\mathbf{c}-\mathbf{y}) &< 0. \end{aligned}$$

But (2.20) implies that the left-hand side of the last inequality is nonnegative (since $\mathbf{p} \geq 0$, $\hat{\mathbf{p}} \geq 0$), which is a contradiction. Therefore the equilibrium is unique.

2.C Proof of Theorem 2.3 (stability)

We will construct a Lyapunov function and use LaSalle's invariance principle [49] to prove global asymptotic stability of the unique equilibrium point $(\mathbf{x}^*, \mathbf{p}^*)$. Define $\delta\mathbf{x} := \mathbf{x} - \mathbf{x}^\star$, $\delta\mathbf{p} := \mathbf{p} - \mathbf{p}^\star$. Consider the candidate Lyapunov function:

$$V(\mathbf{x},\mathbf{p}) = \sum_{r\in R}\int_{x_r^*}^{x_r} \frac{z - x_r^*}{k_r(z)}dz + \frac{1}{2}\sum_{l\in L}\frac{\delta p_l^2}{\gamma_l}. \tag{2.21}$$

By definition, $V(\mathbf{x},\mathbf{p}) > 0$ for all $(\mathbf{x},\mathbf{p}) \neq (\mathbf{x}^*,\mathbf{p}^*)$ and $V(\mathbf{x},\mathbf{p}) = 0$ if $(\mathbf{x},\mathbf{p}) = (\mathbf{x}^*,\mathbf{p}^*)$. Furthermore V is radially unbounded, i.e., $V(\mathbf{x},\mathbf{p}) \to \infty$ as $\|(\mathbf{x},\mathbf{p})\|_2 \to \infty$. Finally

$$\dot{V}(\mathbf{x},\mathbf{p}) = \sum_{r\in R}\frac{1}{k_r(x_r)}\delta x_r\dot{x}_r + \sum_{l\in L}\frac{1}{\gamma_l}\delta p_l\dot{p}_l.$$

If $\delta x_r \neq 0$ then we have (since $k_r(\mathbf{x}_s) = k_r(x_r)$)

$$\begin{aligned} \frac{1}{k_r(x_r)}\delta x_r\dot{x}_r &= \delta x_r\left(\phi_r(\mathbf{x}_s) - q_r\right)_{x_r}^+ \\ &\leq \delta x_r\left(\phi_r(\mathbf{x}_s) - q_r\right) \\ &= \delta x_r\left(\phi_r(\mathbf{x}_s) - \phi_r(\mathbf{x}_s^*) - \delta q_r\right). \end{aligned}$$

The first inequality holds since $(\phi_r(\mathbf{x}_s)-q_r)^+_{x_r} = \phi_r(\mathbf{x}_s)-q_r$ if $x_r > 0$ and $\phi_r(\mathbf{x}_s)-q_r \le 0$, $\delta x_r = -x^*_r$ if $x_r = 0$. The last equality holds since $\phi_r(\mathbf{x}^*_s) = q^*_r$ by Lemma 2.10 and (2.6). Therefore

$$\begin{aligned}\sum_{r\in R}\frac{1}{k_r(x_r)}\delta x_r\dot{x}_r &\le \delta\mathbf{x}^T(\Phi(\mathbf{x})-\Phi(\mathbf{x}^*)) - \delta\mathbf{x}^T\delta\mathbf{q}\\ &< -\delta\mathbf{x}^T H^T\delta\mathbf{p},\end{aligned}$$

where the last inequality holds since $\delta\mathbf{x}^T(\phi(\mathbf{x})-\phi(\mathbf{x}^*)) < 0$ by Lemma 2.9 and assumption C2. Similarly

$$\frac{1}{\gamma_l}\delta p_l\dot{p}_l \;=\; \delta p_l(y_l-c_l)^+_{p_l} \;\le\; \delta p_l(y_l-c_l) \;\le\; \delta p_l\delta y_l,$$

where the last inequality holds since $\delta p_l c_l \ge \delta p_l y^*_l$ by the equilibrium condition (2.7). Thus

$$\sum_{l\in L}\frac{1}{\gamma_l}\delta p_l\dot{p}_l \;\le\; \delta\mathbf{p}^T H\delta\mathbf{x}.$$

Therefore if $\delta\mathbf{x}\neq 0$ then

$$\dot{V}(\mathbf{x},\mathbf{p}) \;<\; -\delta\mathbf{x}^T H^T\delta\mathbf{p} + \delta\mathbf{p}^T H\delta\mathbf{x} \;=\; 0$$

and if $\delta\mathbf{x} = 0$ then $\dot{V}(\mathbf{x},\mathbf{p}) = 0$. This means $\dot{V}(\mathbf{x},\mathbf{p}) \le 0$ and V is indeed a Lyapunov function.

Consider the set

$$Z \;:=\; \{\,(\mathbf{x}(t),\mathbf{p}(t)) \,|\, \dot{V}(\mathbf{x}(t),\mathbf{p}(t)) = 0 \text{ for all } t\ge 0\,\}$$

of trajectories on which $\dot{V}\equiv 0$. If the only trajectory in Z is the trivial trajectory $(\mathbf{x},\mathbf{p})\equiv(\mathbf{x}^*,\mathbf{p}^*)$ then LaSalle's invariance principle implies that $(\mathbf{x}^*,\mathbf{p}^*)$ is globally asymptotically stable. We now show that this is indeed the case.

As shown above $\dot{V}\equiv 0$ implies $\delta\mathbf{x}\equiv 0$, i.e., any trajectory $(\mathbf{x}(t),\mathbf{p}(t))$ in Z must have $\mathbf{x}(t) = \mathbf{x}^*$ for all $t\ge 0$. This means $\dot{\mathbf{x}}\equiv 0$ and hence for all $t\ge 0$, $\mathbf{q}(t) = \Phi(\mathbf{x}(t))$ since $\mathbf{x}(t) = \mathbf{x}^* > 0$ by Lemma 2.10. That is, for all $t\ge 0$, $H^T\mathbf{p}(t) = \Phi(\mathbf{x}^*)$ and hence $\mathbf{p}(t) = \mathbf{p}^*$ since H has full row rank by C3. Therefore $(\mathbf{x},\mathbf{p})\equiv(\mathbf{x}^*,\mathbf{p}^*)$ is indeed the only trajectory in Z. This completes the proof of Theorem 2.3.

2.D Proof of Theorem 2.4 (friendliness)

Let the MP-TCP source be defined by

$$\phi_r(\mathbf{x}_s;\mu) = \mu\tilde{\phi}_r(\mathbf{x}_s) + (1-\mu)\hat{\phi}_r(\mathbf{x}_s), \quad \mu \in [0,1].$$

Algorithm $\hat{M}$ and $\tilde{M}$ correspond to $\mu = 0$ and $\mu = 1$, respectively. Let x_g and τ_g be the throughput and RTT of the TCP NewReno source in Fig. 2.1. The equilibrium is defined by $F(\mathbf{x},\mu) = 0$, where $\mathbf{x} := (\mathbf{x}_s, x_g)$ and F is given by

$$\begin{aligned} \Phi_s(\mathbf{x}_s;\mu) - \frac{1}{\tau_g^2 x_g^2}\mathbf{1} &= 0 \\ \mathbf{1}^T\mathbf{x}_s + x_g &= c, \end{aligned}$$

where the first equation follows from

$$p^* = \frac{1}{\tau_g^2 x_g^2} = \phi_r(\mathbf{x}_s;\mu), \quad r \in s$$

and p^* is the congestion price at the bottleneck link. Applying the implicit function theorem, we get

$$\begin{aligned} \frac{d\mathbf{x}}{d\mu} &= -\left(\frac{\partial F}{\partial \mathbf{x}}\right)^{-1}\frac{\partial F}{\partial \mu} \\ &= -\begin{bmatrix} \frac{\partial \Phi_s}{\partial \mathbf{x}_s} & \frac{2}{x_g^3}\mathbf{1} \\ \mathbf{1}^T & 1 \end{bmatrix}^{-1}\begin{bmatrix} \tilde{\Phi}_s(\mathbf{x}_s) - \hat{\Phi}_s(\mathbf{x}_s) \\ 0 \end{bmatrix}, \end{aligned}$$

where the inverse exists by condition C2. C2 also guarantees the inverse of $\frac{\partial \Phi_s}{\partial \mathbf{x}_s}(\mathbf{x}_s;\mu)$, denoted by $D(\mu)$; C4 ensures $\sum_{i\in s} D_{ij}(\mu) \le 0$. Let

$$A := \frac{\partial \Phi_s}{\partial \mathbf{x}_s} - \frac{2}{x_g^3}\mathbf{1}\mathbf{1}^T \quad \text{and} \quad d := 1 - \frac{2}{x_g^3}\sum_{i,j} D_{ij}(\mu).$$

Then

$$\begin{bmatrix} \frac{\partial \Phi_s}{\partial \mathbf{x}_s} & \frac{2}{x_p^3}\mathbf{1} \\ \mathbf{1}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -D\mathbf{1}d \\ -d\mathbf{1}^T A^{-1} & d^{-1} \end{bmatrix}.$$

Thus

$$\begin{aligned}\mathbf{1}^T\frac{\partial \mathbf{x}_s}{\partial \mu} &= -[\mathbf{1}^T 0]\left(\frac{\partial F}{\partial \mathbf{x}}\right)^{-1}\frac{\partial F}{\partial \mu} \\ &= -\mathbf{1}^T A^{-1}(\tilde{\Phi}_s(\mathbf{x}_s)-\hat{\Phi}_s(\mathbf{x}_s)). \end{aligned} \tag{2.22}$$

By matrix inverse formula,

$$\begin{aligned}A^{-1} &= \left(\frac{\partial \Phi_s}{\partial \mathbf{x}_s}-\frac{2}{x_g^3}\mathbf{1}\mathbf{1}^T\right)^{-1} \\ &= D(\mu)+\frac{1}{\frac{x_g^3}{2}-\mathbf{1}^T D(\mu)\mathbf{1}}D(\mu)\mathbf{1}\mathbf{1}^T D(\mu).\end{aligned}$$

Substitute it into (2.22), and we have

$$\begin{aligned}&\mathbf{1}^T A^{-1}(\hat{\Phi}_s(\mathbf{x}_s)-\tilde{\Phi}_s(\mathbf{x}_s)) \\ =&\left(1+\frac{\mathbf{1}^T D(\mu)\mathbf{1}}{\frac{x_g^3}{2}-\mathbf{1}^T D(\mu)\mathbf{1}}\right)\mathbf{1}^T D(\mu)(\tilde{\Phi}_s(\mathbf{x}_s)-\hat{\Phi}_s(\mathbf{x}_s)) \\ =&\frac{x_g^3}{x_g^3-2\mathbf{1}^T D(\mu)\mathbf{1}}\sum_{r\in s}\left(\sum_{i\in s}D_{ir}(\mu)\right)(\tilde{\phi}_r(\mathbf{x}_s)-\hat{\phi}_r(\mathbf{x}_s)) \\ \leq&0,\end{aligned}$$

where the inequality follows because $D(\mu)$ is negative definite, $\sum_{i\in s}D_{ir}(\mu) < 0$ and $\tilde{\phi}_r(\mathbf{x}_s) - \hat{\phi}_r(\mathbf{x}_s) \geq 0$. Thus we have $\mathbf{1}^T\frac{\partial \mathbf{x}_s}{\partial \mu} \geq 0$ for $\mu \in [0,1]$, i.e., the aggregate throughput of the MP-TCP over its available paths is increasing in μ. This means $\tilde{M}$ (corresponding to $\mu = 1$) will attain a higher throughput than $\hat{M}$ (corresponding to $\mu = 0$) when separately sharing the test network in Fig. 2.1 with the same SP-TCP.

2.E Proof of Theorem 2.5 (responsiveness)

2.E.1 Proof of part 1

Fix any eigenvalue λ of J^*. Let $\mathbf{z} := (\mathbf{x}, \mathbf{p}) \in Z$ be the corresponding eigenvector with $\|\mathbf{z}\|_2 = 1$. Then we have

$$\lambda\begin{bmatrix}\mathbf{x}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}\Lambda_k & 0\\ & \Lambda_\gamma\end{bmatrix}\begin{bmatrix}\frac{\partial \Phi}{\partial \mathbf{x}} & -H^T\\ H & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{p}\end{bmatrix}.$$

Thus

$$\lambda \begin{bmatrix} \Lambda_k^{-1} & 0 \\ & \Lambda_\gamma^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \frac{\partial \Phi}{\partial \mathbf{x}} & -H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix}.$$

Premultiplying $\mathbf{z}^H$ on both sides, we have

$$\lambda \;=\; \frac{\mathbf{x}^H \frac{\partial \Phi}{\partial \mathbf{x}} \mathbf{x} \;+\; (\mathbf{p}^H H \mathbf{x} - \mathbf{x}^H H^T \mathbf{p})}{\mathbf{x}^H \Lambda_k^{-1} \mathbf{x} + \mathbf{p}^H \Lambda_\gamma^{-1} \mathbf{p}}.$$

The denominator is real and positive, and $(\mathbf{p}^H H \mathbf{x} - \mathbf{x}^H H^T \mathbf{p})$ in the numerator is imaginary. Thus

$$\begin{aligned} \mathbf{Re}(\lambda) &= \frac{\mathbf{Re}\left(\mathbf{x}^H \frac{\partial \Phi}{\partial \mathbf{x}} \mathbf{x}\right)}{\mathbf{x}^H \Lambda_k^{-1} \mathbf{x} + \mathbf{p}^H \Lambda_\gamma^{-1} \mathbf{p}} \\ &= \frac{\mathbf{x}^H \left[\frac{\partial \Phi}{\partial \mathbf{x}}\right]^+ \mathbf{x}}{\mathbf{x}^H \Lambda_k^{-1} \mathbf{x} + \mathbf{p}^H \Lambda_\gamma^{-1} \mathbf{p}} \;<\; 0, \end{aligned}$$

where the last inequality holds because the numerator is negative by condition C2 and the denominator is positive. Since this holds for all eigenvalues λ of J^*, the linearized system (2.10) is stable. Moreover $\mathbf{Re}(\lambda) \le \overline{\lambda}(J^*) \le 0$ as desired.

2.E.2 Proof of part 2

Consider two MP-TCP algorithms $(\hat{K}, \hat{\Phi})$ and $(\tilde{K}, \tilde{\Phi})$ such that

$$\hat{K}_s \;\ge\; \tilde{K}_s \quad \text{and} \quad \frac{\partial \hat{\Phi}_s}{\partial \mathbf{x}_s} \preceq \frac{\partial \tilde{\Phi}_s}{\partial \mathbf{x}_s} \qquad \text{for all } s \in S.$$

For any (nonzero) $\mathbf{z} = (\mathbf{x}, \mathbf{p}) \in Z$ we have

$$0 \;\le\; \mathbf{x}^H \hat{\Lambda}_k^{-1} \mathbf{x} \;\le\; \mathbf{x}^H \tilde{\Lambda}_k^{-1} \mathbf{x} \tag{2.23}$$

$$\mathbf{x}^H \left[\frac{\partial \hat{\Phi}}{\partial \mathbf{x}}\right]^+ \mathbf{x} \;\le\; \mathbf{x}^H \left[\frac{\partial \tilde{\Phi}}{\partial \mathbf{x}}\right]^+ \mathbf{x} \;<\; 0. \tag{2.24}$$

Thus $\overline{\lambda}(\hat{J}^*) \le \overline{\lambda}(\tilde{J}^*)$.

2.F Proof of Theorem 2.6 (tradeoff)

Fix an s. Let $f_r(\mathbf{x}_s) := \hat{\phi}_r(\mathbf{x}_s) - \tilde{\phi}_r(\mathbf{x}_s)$ and $F(\mathbf{x}_s) := (f_r(\mathbf{x}_s), r \in s) = \hat{\Phi}_s(\mathbf{x}_s) - \tilde{\Phi}_s(\mathbf{x}_s)$. Suppose for the sake of contradiction that $\partial \hat{\Phi}_s(\mathbf{x}_s)/\partial \mathbf{x}_s \preceq \partial \tilde{\Phi}_s(\mathbf{x}_s)/\partial \mathbf{x}_s$ but $\hat{\Phi}_s(\mathbf{x}_s) \ge \tilde{\Phi}_s(\mathbf{x}_s)$ does not hold,

i.e., there exists a finite $\mathbf{x}_s^0$ and a $r \in s$ such that

$$f_r(\mathbf{x}_s^0) \;=\; \hat{\phi}_r(\mathbf{x}_s^0) - \tilde{\phi}_r(\mathbf{x}_s^0) \quad < \quad 0. \tag{2.25}$$

Since $[\partial F/\partial \mathbf{x}_s]^+ \preceq 0$ by assumption, a trivial modification of Lemma 2.9 shows that for all $\mathbf{x}_s \neq \mathbf{x}_s^0$, $(\mathbf{x}_s - \mathbf{x}_s^0)^T(F(\mathbf{x}_s) - F(\mathbf{x}_s^0)) \leq 0$, i.e.,

$$0 \quad \geq \quad \sum_{r' \in s} (x_{r'} - x_{r'}^0)\,(f_{r'}(\mathbf{x}_s) - f_{r'}(\mathbf{x}_s^0)). \tag{2.26}$$

Choose an $\mathbf{x}_s$ as follows: for all $r' \neq r$, choose $x_{r'} = x_{r'}^0$ and then use condition C5 to choose an $x_r < \infty$ large enough so that $x_r > x_r^0$ and $f_r(\mathbf{x}_s) > f_r(\mathbf{x}_s^0)/2$. With this $\mathbf{x}_s$, (2.26) becomes

$$\begin{aligned} 0 \quad &\geq \quad (x_r - x_r^0)\,(f_r(\mathbf{x}_s) - f_r(\mathbf{x}_s^0)) \\ &> \quad (x_r - x_r^0)\left(-\frac{f_r(\mathbf{x}_s^0)}{2}\right) \;>\; 0, \end{aligned}$$

where the last inequality follows from (2.25). This is a contradiction and hence $\hat{\Phi}_s(\mathbf{x}_s) \geq \tilde{\Phi}_s(\mathbf{x}_s)$.

2.G Proof of Theorem 2.8

We will show the results hold for any $n \in \mathbb{N}_+$. Since $\lim_{n\to\infty} \|\mathbf{x}_s\|_n = \|\mathbf{x}_s\|_\infty$, the results also hold for $n = \infty$. When $\beta = 0$, it is easy to show that ϕ_r satisfies C1 and $\left[\frac{\partial \Phi_s}{\partial \mathbf{x}_s}\right]^+$ is negative semidefinite under the conditions of the theorem. We hence prove the theorem for $\beta > 0$.

2.G.1 Proof of part 1

Fix any $n \in \mathbb{N}_+$ and $\beta > 0$. Fix any finite $\mathbf{p} \geq 0$ such that $q_r > 0$ for all r. Fix any $s \in S$. We now show that there exists an $\mathbf{x}_s > 0$ that satisfies (2.6), in particular $\phi_r(\mathbf{x}_s) = q_r$, in two steps.

First, there exists an $\mathbf{x}_s$ that satisfies $\phi_r(\mathbf{x}_s) = q_r$ if and only if

$$\phi_r(\mathbf{x}_s) = \frac{2}{\tau_r^2\,\|\mathbf{x}_s\|_1^2}\left(1 \;+\; \beta\left(\frac{\|\mathbf{x}_s\|_n}{x_r} - 1\right)\right) = q_r, \tag{2.27}$$

which is equivalent to

$$\frac{x_r}{\|\mathbf{x}_s\|_n} \quad = \quad \frac{2\beta}{2\beta + q_r\tau_r^2\|\mathbf{x}_s\|_1^2 - 2}. \tag{2.28}$$

Since this holds for all $r \in s$, we have

$$\begin{aligned} 1 &= \sum_{r\in s}\left(\frac{x_r}{\|\mathbf{x}_s\|_n}\right)^n \\ &= \sum_{r\in s}\left(\frac{2\beta}{2\beta + q_r\tau_r^2\|\mathbf{x}_s\|_1^2 - 2}\right)^n =: \psi\left(\|\mathbf{x}_s\|_1^2\right). \end{aligned} \tag{2.29}$$

Clearly $\psi(C) \to 0$ as $C \to \infty$. Let

$$\underline{C} := \frac{2}{\min_{r\in s}\, q_r\tau_r^2}. \tag{2.30}$$

Then $\underline{C} < \infty$ since $q_r > 0$ for all r by assumption. Moreover $q_r\tau_r^2\underline{C} \geq 2$ for all $r \in s$ and hence

$$\psi(\underline{C}) = 1 + \sum_{r\neq \underline{r}}\left(\frac{2\beta}{2\beta + q_r\tau_r^2\underline{C} - 2}\right)^n > 1,$$

where $\underline{r}$ is a minimizing $r \in s$ in (2.30). Since $\psi(C)$ is continuous, there exists an $\tilde{C} \in [\underline{C}, \infty)$ with $\psi(\tilde{C}) = 1$. Moreover such a $\tilde{C}$ is unique since $\psi(C)$ is strictly decreasing.

Finally consider the set of $\mathbf{x}_s$ with $\|\mathbf{x}_s\|_1^2 = \tilde{C}$. All such $\mathbf{x}_s$ satisfy (2.28) with

$$x_r = \frac{2\beta}{2\beta + q_r\tau_r^2\tilde{C} - 2}\|\mathbf{x}_s\|_n =: a_r\|\mathbf{x}_s\|_n. \tag{2.31}$$

But $\tilde{C} = \|\mathbf{x}_s\|_1^2 = \left(\sum_{r\in s} a_r\|\mathbf{x}_s\|_n\right)^2$, implying

$$\|\mathbf{x}_s\|_n = \frac{\sqrt{\tilde{C}}}{\sum_{r\in s} a_r}.$$

In summary, given any finite $\mathbf{p} \geq 0$ such that $q_r > 0$ for all r, a solution $\mathbf{x}_s > 0$ to (2.28) is *uniquely* given by

$$x_r = \frac{a_r}{\sum_{k\in s} a_k}\sqrt{\tilde{C}}, \qquad r \in s, \tag{2.32}$$

where

$$a_r := \frac{2\beta}{2\beta + q_r\tau_r^2\tilde{C} - 2}$$

and $\tilde{C} = \|\mathbf{x}_s\|_1^2$ is the unique value at which $\psi(\tilde{C}) = 1$.

We now prove the other conditions in C1:

$$\frac{\partial y_l^s(\mathbf{p})}{\partial p_l} \leq 0, \qquad \lim_{p_l\to\infty} y_l^s(\mathbf{p}) = 0.$$

According to (2.29), we can show that $\tilde{C}$ is a decreasing function of q_r and $q_r\tau_r^2\tilde{C}$ is an increasing function of q_r for $r \in s$. Thus, $\tilde{C}$ is a decreasing function of p_l and $q_r\tau_r^2\tilde{C}$ is an increasing of p_l if $l \in r$ because $q_r = \sum_{l\in L} H_{lr}p_l$. For each $l \in L$, let $s_l := \{r \mid l \in r, r \in s\}$, then by definition and (2.32), we have

$$y_l^s(\mathbf{p}) = \frac{\sum_{r\in s_l} a_r}{\sum_{r\in s} a_r}\sqrt{\tilde{C}} = \frac{\sum_{r\in s_l} a_r}{\sum_{r\in s_l} a_r + \sum_{r\notin s_l} a_r}\sqrt{\tilde{C}}.$$

Since a_r is a decreasing function of $q_r\tau_r^2\tilde{C}$, it is also a decreasing function of p_l if $l \in r$. Recall that $\sqrt{\tilde{C}}$ is also a decreasing function of p_l, $y_l^s(\mathbf{p})$ is thus a decreasing function of p_l, in other words, $\frac{\partial y_l^s(\mathbf{p})}{\partial p_l} \le 0$.

On the other hand, as $p_l \to \infty$, $q_r \to \infty$ for all paths r traversing l. Then $x_r \to 0$ by (2.27) for $l \in r$, which shows $\lim_{p_l\to\infty} y_l^s(\mathbf{p}) = 0$.

2.G.2 Proof of part 2

To prove $\phi_r(\mathbf{x}_s)$ satisfies C2 and C3 for $\beta > 0$, we will show that the Jacobian $\partial\Phi_s(\mathbf{x}_s)/\partial\mathbf{x}_s$ is negative definite if $0 < \beta \le 1$, $|s| \le 8$ and τ_r are the same for $r \in s$. Other properties of C2 and C3 are easy to prove and we omit the proof. Fix an s and let $\tau_r = \tau$, the common round-trip time for all $r \in s$.

Let $\Lambda_s := \mathrm{diag}\{\mathbf{x}_s\}$ and

$$\mathbf{a}_s \;:=\; \left(\frac{2x_r}{\|\mathbf{x}_s\|_1} - \frac{x_r^n}{\|\mathbf{x}_s\|_n^n},\ r \in s\right).$$

Then the Jacobian of Φ_s at $\mathbf{x}_s$ is

$$\frac{\partial\Phi_s}{\partial\mathbf{x}_s} = -\frac{4(1-\beta)}{\tau^2\|\mathbf{x}_s\|_1^3}\mathbf{1}\mathbf{1}^T - 2\beta\frac{\|\mathbf{x}_s\|_n}{\tau^2\|\mathbf{x}_s\|_1^2}\Lambda_s^{-1}\left(I_{|s|} + \mathbf{1}\mathbf{a}_s^T\right)\Lambda_s^{-1}$$

and it is negative definite for $\beta > 0$ if $\left[I_{|s|} + \mathbf{1}\mathbf{a}_s^T\right]^+$ is positive definite. We now show that this is indeed the case when $|s| \le 8$, i.e., for any $\mathbf{z}_s \in \mathbb{R}^{|s|}$,

$$\mathbf{z}_s^T(I_{|s|} + \mathbf{1}\mathbf{a}_s^T)\mathbf{z}_s = \|\mathbf{z}_s\|_2^2 + \sum_{r\in s} z_r \sum_{r\in s} a_r z_r \;>\; 0. \tag{2.33}$$

By Lemma 2.11 below, $\mathbf{1}^T\mathbf{a}_s = 1$ and $\|\mathbf{a}_s\|_2^2 \le 1$. Then (2.33) follows from Lemma 2.12 below provided $|s| \le 8$. Thus the Jacobian is negative definite.[3] The proof of Theorem 2.8 is complete after Lemmas 2.11 and 2.12 are proved.

[3] If $\beta = 0$ the Jacobian degenerates to

$$\frac{\partial\Phi_s}{\partial\mathbf{x}_s} = -\frac{4}{\tau^2\|\mathbf{x}_s\|_1^3}\mathbf{1}\mathbf{1}^T, \tag{2.34}$$

which is merely negative semidefinite.

To show that it satisfies C3, it follows directly from (2.27) that if $x_r = 0$ then $\phi_r(\mathbf{x}_s) = \infty$. It is also clear from (2.27) that the converse holds. This proves C3.

Lemma 2.11. *Fix any integer $p \geq 1$. Given any $\mathbf{x} \in \mathbb{R}^m_+$, define a vector $\mathbf{a}$ in $\mathbb{R}^m$ as follows:*

$$a_i \quad = \quad \frac{2x_i}{\sum_{j=1}^m x_j} - \frac{x_i^p}{\sum_{j=1}^m x_j^p}, \quad 1 \leq i \leq m.$$

Then $\sum_{i=1}^m a_i = 1$ and $\sum_{i=1}^m a_i^2 \leq 1$.

Proof. It is obvious that $\sum_{i=1}^m a_i = 1$. To show $\sum_{i=1}^m a_i^2 \leq 1$, we have

$$\begin{aligned}
\sum_{i=1}^m a_i^2 &= \frac{\sum_i x_i^{2p}}{\left(\sum_j x_j^p\right)^2} + \frac{4\sum_i x_i^2}{\left(\sum_j x_j\right)^2} - \frac{4\sum_i x_i^{p+1}}{\left(\sum_j x_j^p\right)\left(\sum_j x_j\right)} \\
&\leq 1 + \frac{4\sum_i x_i^2}{\left(\sum_j x_j\right)^2} - \frac{4\sum_i x_i^{p+1}}{\left(\sum_j x_j^p\right)\left(\sum_j x_j\right)} \\
&= 1 - 4\,\frac{\sum_{1\leq i<j\leq m} x_i x_j (x_i - x_j)\left(x_i^{p-1} - x_j^{p-1}\right)}{\left(\sum_j x_j\right)^2\left(\sum_j x_j^p\right)} \\
&\leq 1.
\end{aligned}$$

□

Lemma 2.12. *Let $\mathbf{a} \in \mathbb{R}^m$ that satisfies $\sum_{i=1}^m a_i = 1$ and $\sum_{i=1}^m a_i^2 \leq 1$. Then for any nonzero $\mathbf{z} \in \mathbb{R}^m$ we have*

$$f(\mathbf{z}) := \sum_{i=1}^m z_i^2 + \sum_{i=1}^m z_i \sum_{i=1}^m a_i z_i \quad > \quad 0$$

provided $m \leq 8$.

Proof. Given any M let $Z_M := \{z \mid \sum_{i=1}^m z_i = M\}$. It then suffices to show that, for every $M \in \mathbb{R}$, $f(z) > 0$ for $z \in Z_M$. Given any M, consider

$$\min_{\mathbf{z}\in Z_M} f(\mathbf{z}) \quad = \quad \min_{\mathbf{z}\in Z_M} \sum_{i=1}^m z_i^2 + M\sum_{i=1}^m a_i z_i. \tag{2.35}$$

Its Lagrangian is

$$L(\mathbf{z}, \mu) = \sum_{i=1}^m z_i^2 + M\sum_{i=1}^m a_i z_i + \mu\left(\sum_{i=1}^m z_i - M\right),$$

where μ is the Lagrange multiplier. Setting $\partial L/\partial z_i = 0$ for all $1 \leq i \leq m$ and substituting it into

$\sum_{i=1}^m z_i = M$, we obtain the unique minimizer given by $\mu = -3M/m$ and $z_i = \frac{M}{2}(\frac{3}{m} - a_i)$. Then

$$\min_{\mathbf{z}\in Z_M} f(\mathbf{z}) \;=\; \frac{M^2}{4}\left(\frac{9}{m} - \sum_{i=1}^m a_i^2\right) \;\geq\; \frac{M^2}{4}\left(\frac{9}{m} - 1\right).$$

Thus, when $M \neq 0$, $\min_{\mathbf{z}\in Z_M} f(\mathbf{z}) > 0$ if $n < 9$. When $\mathbf{z}$ is nonzero but $M = 0$, then $f(\mathbf{z}) > 0$ from (2.35). □

2.H Proof of Lemma 2.7

By the definition of $D_k(A_k)$, we have

$$\begin{aligned}\mathbb{E}\left[D_k(A_k) \mid \sum_{i,j} a_{ij} \geq 1\right] =& d_k \mathbb{P}\left(\sum_j a_{kj} \geq 1 \mid \sum_{i,j} a_{ij} \geq 1\right)\\ =& d_k \frac{\mathbb{P}(\sum_j a_{kj} \geq 1)}{\mathbb{P}(\sum_{i,j} a_{ij} \geq 1))}\\ =& d_k \frac{q_k|A_k|}{\sum_i q_i|A_i|} + o\left(\sum_i q_i\right),\end{aligned}$$

where the last equality follows from the independence of a_{ij} and $\mathbb{P}(\sum_j a_{kj} \geq 1) = 1 - (1-q_k)^{|A_k|} = |A_k|q_k + o(q_k)$, $\mathbb{P}(\sum_{ij} a_{ij} \geq 1) = 1 - \prod_i (1-q_i)^{|A_i|} = \sum_i |A_i|q_i + \sum_i o(q_i)$. Thus,

$$\mathbb{E}\left[\sum_i D_k(A_k) \mid \sum_{i,j} a_{ij} \geq 1\right] = \frac{\sum_k d_k q_k |A_k|}{\sum_k q_k |A_k|} + o\left(\sum_k q_k\right).$$

Chapter 3

Optimal Power Flow and Convex Relaxation

The optimal power flow (OPF) problem is a mathematical optimization program that seeks to minimize a certain objective such as generation cost or power loss, subject to the Kirchhoff's laws and operational constraints. It is fundamental in power system as it underlies many applications such as economic dispatch, unit commitment, state estimation, stability and reliability assessment, volt/var control, demand response, etc. It was first formulated in the seminal work [14] in 1962.

Literature The power flow equations are quadratic and hence the OPF problem can be formulated as a quadratic constrained quadratic program (QCQP). It is generally non-convex and hence hard to solve. DC-OPF, which is a linear approximation of the non-convex OPF problem, is obtained through linearization of the power flow equations [3, 78, 79]. The accuracy of DC-OPF hinges on the fact that 1) line resistances are small, 2) voltages magnitudes are approximately constants, and 3) voltage angle differences between adjacent buses are small. DC-OPF approximation is widely used for transmission networks, but does not apply to distribution networks where line resistances are not small and voltage magnitudes may significantly deviate from their nominal value. Furthermore, DC-OPF is not applicable for many applications where reactive power needs to be actively controlled, e.g. volt/var control.

Recently a new approach through convex relaxation has been developed. There are two types convex relaxations of the OPF problem: semidefinite programming (SDP) relaxation and second order cone programming (SOCP) relaxation. For SDP relaxation, the nonlinear power flow equations are transformed into linear constraints on a rank one positive semidefinite matrix, and the rank one constraint is removed to obtain an SDP relaxation. It is first proposed in [4] for the *bus injection model* and shown to be exact for all the IEEE test networks [56]. SDP is computationally prohibitive for large networks because the original n dimensional space is lifted to n^2 dimensional space. By assuming *balanced* power flow, SOCP relaxation has been proposed in both the bus

injection model [44] and the branch flow model [27]. Compared with the semidefinite program, the second order cone program can be solved more efficiently [2]. However, the assumption of balanced power flow only holds for transmission networks. In distribution networks, which are the focus of this thesis, power flow is not balanced. Recently, a computationally efficient SDP relaxation, which leverages the radial structure of distribution networks, has been proposed in [33] for networks with unbalanced power flow. See the tutorial [60, 61] for further pointers to the literature.

When an optimal solution of the original OPF problem can be recovered from any optimal solution of a convex relaxation, we say the relaxation is *exact*. For radial distribution networks (whose graphs are trees), it is shown that both SOCP and SDP relaxation are exact for standard IEEE test networks and many practical networks, both in the balanced case (*e.g.* [27, 56]) and unbalanced case (*e.g.* [22, 33]). This is important because almost all distribution systems are radial. Then we can rely on off-the-shelf convex optimization solvers to obtain a globally optimal solution for the nonconvex OPF problem.

Summary In this chapter, we will review the mathematical formulation of the OPF problem on both *balanced* and *unbalanced* networks. We will also review the recent development of SDP/SOCP relaxation on solving the OPF problems. These results will be used extensively in this thesis and are foundations for solving feeder reconfiguration problem (Chapter 4) and developing distributed OPF algorithms (Chapter 6 and 7).

3.0.1 Notations

The distribution network consists of substation and load buses, and distribution lines that connect these buses. Substation buses receive bulk power from the transmission networks and deliver power to load buses through the distribution lines. We model a distribution network by a *directed* graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ represents the set of buses and $\mathcal{E}$ represents the set of distribution lines connecting the buses in $\mathcal{N}$. Let $\mathcal{N}_s \subset \mathcal{N}$ denote the set of substation buses. Index one of the substation bus by 0, defined as the root of the graph. Each directed line $(i, j) \in \mathcal{E}$ connects bus i and j, where bus j lies between bus 0 and bus i. Then we call j as i's ancestor and i as j's child. For each bus $i \in \mathcal{N}$, let $A_i := \{j \mid (i, j) \in \mathcal{E}\}$ denote the set of i's ancestors, $C_i := \{k \mid (k, i) \in \mathcal{E}\}$ denote the set of i's children, and $N_i := \{i\} \cup A_i \cup C_i$ denote the set of i's neighbor including itself. The notations are illustrated in Figure 3.1.

We denote the set of complex numbers with $\mathbb{C}$, the set of n-dimensional complex numbers with $\mathbb{C}^n$, and the set of $m \times n$ complex matrix with $\mathbb{C}^{m\times n}$. The set of hermitian (positive semidefinite) matrix is denoted by $\mathbb{S}$ ($\mathbb{S}_+$). The hermitian transpose of a vector (matrix) x is denoted by x^H.

The trace of a square matrix $x \in \mathbb{C}^{n\times n}$ is denoted by $tr(x) := \sum_{i=1}^n x_{ii}$. The inner product of two matrices (vectors) $x, y \in \mathbb{C}^{m\times n}$ is denoted by $\langle x, y\rangle := \mathbf{Re}(tr(x^H y))$. The Frobenius (Euclidean)

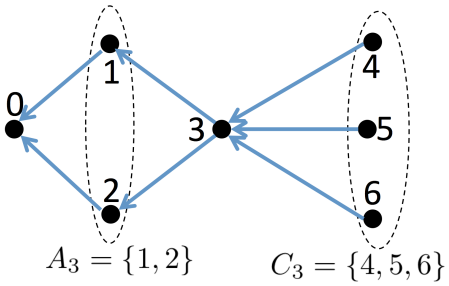


Figure 3.1: Notations of graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$, where the ancestor and children set of node 3 are also labeled explictly.

norm of a matrix (vector) $x \in \mathbb{C}^{m\times n}$ is defined as $\|x\|_2 := \sqrt{\langle x, x\rangle}$. Given $x \in \mathbb{C}^{n\times n}$, let $\mathrm{diag}(x) \in \mathbb{C}^{n\times 1}$ denote the vector composed of x's diagonal elements. A variable without a subscript denotes a column vector with appropriate components.

3.1 OPF and its SOCP Relaxation on Balanced Networks

In this section, we formulate the OPF problem on balanced networks and show how to solve it through SOCP relaxation. We employ the *branch flow model* (BFM) to model the power flow equations, which are first proposed in [7, 8].

3.1.1 Branch flow model

Under the assumption that the network has balanced three phases, we can simply analyze each phase independently. For each bus $i \in \mathcal{N}$, let $V_i = |V_i|e^{\mathbf{i}\theta_i}$ be its complex voltage. Let $s_i := p_i + \mathbf{i}q_i$ be its net complex power injection, which is generation minus load. For each line $(i, j) \in \mathcal{E}$, let $z_{ij} = r_{ij} + \mathbf{i}x_{ij}$ be its complex impedance and $y_{ij} = z_{ij}^{-1}$ be its complex admittance. Let I_{ij} be the complex branch current from bus i to j. Then bus injection model (BIM) can be described as

$$V_i - V_j - z_{ij}I_{ij} = 0 \qquad (i, j) \in \mathcal{E} \tag{3.1a}$$

$$s_i + \sum_{j\in C_i} V_i I_{ji} - \sum_{k\in A_i} V_i I_{ik} = 0 \qquad i \in \mathcal{N}, \tag{3.1b}$$

where (3.1a) describes the Kirchhoff's law and (3.1b) describes power balance at bus i. BIM does not directly model branch variables, which are crucial for some applications. Branch flow model (BFM), on the other hand, models both nodal and branch variables. It is first proposed in [7, 8] and is more numerically stable than BIM. It has been advocated for the design and operation for radial distribution networks; see [16, 27, 32, 57, 84, 86] for recent papers. We employ BFM in this thesis and its formulation is given below.

Some new notations are needed to describe BFM. For each bus $i \in \mathcal{N}$, denote by $v_i := |V_i|^2$ the

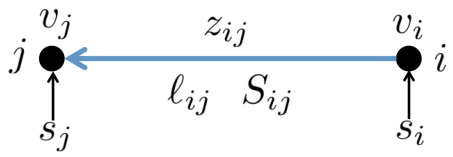


Figure 3.1: Notations for Balanced Network.

voltage magnitude squared. For each line $(i,j) \in \mathcal{E}$, denote by $S_{ij} := P_{ij} + \mathbf{i}Q_{ij}$ the branch power flow from bus i to j. Then BFM ignores the phase angles of voltages and currents and uses only the set of variables (v, s, ℓ, S), where

$$\begin{aligned} v &:= (v_i, i \in \mathcal{N}) \\ s &:= (s_i, i \in \mathcal{N}) \\ \ell &:= (\ell_{ij}, (i,j) \in \mathcal{E}) \\ S &:= (S_{ij}, (i,j) \in \mathcal{E}). \end{aligned}$$

Given a network $\mathcal{G}(\mathcal{N}, \mathcal{E})$, the branch flow model is defined by:

$$v_j - v_i + (z_{ij} S_{ij}^* + S_{ij} z_{ij}^*) - \ell_{ij} |z_{ij}|^2 = 0 \qquad (i,j) \in \mathcal{E} \tag{3.2a}$$

$$\sum_{k \in C_i} (S_{ki} - \ell_{ki} z_{ki}) + s_i - \sum_{j \in A_i} S_{ij} = 0 \qquad i \in \mathcal{N} \tag{3.2b}$$

$$|S_{ij}|^2 = v_i \ell_{ij}. \qquad (i,j) \in \mathcal{E} \tag{3.2c}$$

Given a vector (v, s, ℓ, S) that satisfies (3.2), the phase angles of the voltages and currents can be uniquely determined if the network is a tree, i.e. $|A_i| = 1$ for $i \in \mathcal{N} \setminus \{0\}$ and $|A_0| = 0$. Hence the branch flow model (3.2) is equivalent to the BIM for radial network.

3.1.2 OPF and SOCP Relaxation

The OPF problem seeks to optimize certain objective, e.g. total line loss or total generation cost, subject to power flow equations (3.2) and various operational constraints. We consider an objective function of the following form:

$$F(s) = \sum_{i \in \mathcal{N}} f_i(s_i). \tag{3.3}$$

For instance,

- to minimize total line loss, we can set for each $i \in \mathcal{N}$,

$$f_i(s_i) = p_i.$$

- to minimize generation cost, we can set for each $i \in \mathcal{N}$,

$$f_i(s_i) = \frac{\alpha_i}{2} p_i^2 + \beta_i p_i,$$

  where $\alpha_i, \beta_i \geq 0$ depends on the type of bus i, e.g. $\alpha_i = 0$ and $\beta_i = 0$ for bus i where there is no generator and for generator bus i, the corresponding α_i, β_i depends on the characteristic of the generator.

We consider two operational constraints. First, the power injection s_i at each bus i is constrained to be in a region $\mathcal{I}_i$, i.e.

$$s_i \in \mathcal{I}_i \quad \text{for } i \in \mathcal{N}. \tag{3.4}$$

The feasible power injection region $\mathcal{I}_i$ is determined by the types of device attached to bus i. Some common devices are given below.

- For controllable load, whose real power can vary within $[\underline{p}_i, \overline{p}_i]$ and reactive power can vary within $[\underline{q}_i, \overline{q}_i]$, the injection region $\mathcal{I}_i$ is

$$\mathcal{I}_i := \{p + \mathbf{i}q \mid p \in [\underline{p}_i, \overline{p}_i], q \in [\underline{q}_i, \overline{q}_i]\} \subseteq \mathbb{C}. \tag{3.5a}$$

- For solar panel connecting the grid through an inverter with nameplate $\overline{s}_i$, the injection region $\mathcal{I}_i$ is

$$\mathcal{I}_i := \{p + \mathbf{i}q \mid p \geq 0, p^2 + q^2 \leq \overline{s}_i^2\} \subseteq \mathbb{C}. \tag{3.5b}$$

Second, the voltage magnitude at each bus $i \in \mathcal{N}$ needs to be maintained within a prescribed region, i.e.

$$\underline{v}_i \leq v_i \leq \overline{v}_i \quad \text{for } i \in \mathcal{N}. \tag{3.6}$$

Typically the voltage magnitude at substation buses is assumed to be fixed at some prescribed value, i.e. $\underline{v}_i = \overline{v}_i$ for $i \in \mathcal{N}$. At other load buses $i \in \mathcal{N} \setminus \mathcal{N}_s$, the voltage magnitude is typically allowed to deviate by 5% from its nominal value 1, i.e. $\underline{v}_i = 0.95^2$ and $\overline{v}_i = 1.05^2$.

To summarize, the OPF problem for balanced networks is

$$\begin{aligned} \textbf{OPF:}\ \min \quad & \sum_{i\in\mathcal{N}} f_i(s_i) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (3.2), (3.4) \text{ and } (3.6). \end{aligned} \tag{3.7}$$

The OPF problem (3.7) is nonconvex due to the quadratic equality constraint (3.2c). In [25,27], (3.2c) is relaxed to a second order cone constraint:

$$|S_{ij}|^2 \le v_i \ell_{ij} \quad \text{for} \quad (i,j) \in \mathcal{E}, \tag{3.8}$$

resulting in a second-order cone program (SOCP) relaxation of (3.7)

$$\begin{aligned} \textbf{ROPF:}\ \min \quad & \sum_{i\in\mathcal{N}} f_i(s_i) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (3.2a),\ (3.2b),\ (3.8) \text{ and } (3.4), (3.6). \end{aligned} \tag{3.9}$$

Clearly the relaxation ROPF (3.9) provides a lower bound for the original OPF problem (3.7) since the original feasible set is enlarged. The relaxation is called *exact* if every optimal solution of ROPF attains equality in (3.2c) and hence is also optimal for the original OPF. For networks with tree topology, SOCP relaxation is exact under some mild conditions [27, 32]. There are also other sufficient conditions for the exactness of SOCP relaxation. Even though they are proved under BIM, the of BFM and BIM implies that these conditions also guarantee that ROPF (3.9) is exact. See [60, 61] for extensive references on these conditions.

3.2 OPF and its SDP relaxation on Unbalanced Networks

In this section, we will formulate the OPF problem on *unbalanced* networks and show how to solve it through SDP relaxation. We employ the branch flow model, which is first generalized to the unbalanced network in [33] that inherents the numerical stability of BFM for balanced networks.

3.2.1 Branch flow model

Let a, b, c denote the three phases of the network. For each bus $i \in \mathcal{N}$, let $\Phi_i \subseteq \{a, b, c\}$ denote the set of phases. In typical networks, the set of phases for bus i is a subset of the phases of its parents and superset of the phases of its children, i.e. $\Phi_i \subseteq \Phi_k$ for $k \in A_i$ and $\Phi_j \subseteq \Phi_i$ for $j \in C_i$. On each phase $\phi \in \Phi_i$, let $V_i^\phi \in \mathbb{C}$ denote the complex voltage and $s_i^\phi := p_i^\phi + jq_i^\phi$ denote the

complex power injection. Denote $V_i := (V_i^\phi, \phi \in \Phi_i) \in \mathbb{C}^{|\Phi_i|}$, $s_i := (s_i^\phi, \phi \in \Phi_i) \in \mathbb{C}^{|\Phi_i|}$. For each line $(i,j) \in \mathcal{E}$ connecting bus i and j, the set of phases is $\Phi_i \cap \Phi_j = \Phi_i$ since $\Phi_i \subseteq \Phi_j$ by our definition of graph orientation. On each phase $\phi \in \Phi_i$, let $z_{ij} \in \mathbb{C}^{|\Phi_i| \times |\Phi_i|}$ denote the impedance and $y_{ij} := z_{ij}^{-1}$ denote the admittance. Let $I_{ij}^\phi \in \mathbb{C}$ denote the complex branch current and denote $I_{ij} := (I_{ij}^\phi, \phi \in \Phi_i) \in \mathbb{C}^{|\Phi_i|}$. Then BIM for unbalanced networks can be written as

$$V_i - \mathcal{P}_i(V_j) - z_{ij} I_{ij} = 0 \qquad (i,j) \in \mathcal{E} \tag{3.10a}$$

$$s_i + \sum_{j \in C_i} V_i \mathcal{P}_i(I_{ji}) - \sum_{k \in A_i} V_i I_{ik} = 0 \qquad i \in \mathcal{N}, \tag{3.10b}$$

where $\mathcal{P}_i(x)$ denotes projecting x onto the set of phases on bus i if x has more phases than Φ_i and lifting x onto the phases of bus i with missing phase filled with 0 if x has less phases than Φ_i, e.g. if $(j,i),(i,k) \in \mathcal{E}$ and $\Phi_k = \{a,b,c\}$, $\Phi_i = \{a,b\}$ and $\Phi_j = \{a\}$, then

$$\begin{aligned} \mathcal{P}_i(V_k) &:= \left(V_k^a, V_k^b\right) \\ \mathcal{P}_i(I_{ji}) &:= \left(I_{ji}^a, 0\right). \end{aligned}$$

Similar to BIM on balanced networks in section 3.1, BIM on unbalanced networks also does not model branch variables directly. In [33], BFM is first generalized to the case of unbalanced networks that both models branch variables and has better numerical stability than BIM.

For each bus $i \in \mathcal{N}$, let $v_i := V_i^H V_i \in \mathbb{C}^{|\Phi_i| \times |\Phi_i|}$. For each line $(i,j) \in \mathcal{E}$, $\ell_{ij} := I_{ij} I_{ij}^H \in \mathbb{C}^{|\Phi_{ij}| \times |\Phi_{ij}|}$ and $S_{ij} := V_i I_{ij}^H \in \mathbb{C}^{|\Phi_i| \times |\Phi_i|}$. The notations are illustrated in Figure 3.1. Then BFM for unbalanced networks uses the set of matrix variables (v, s, ℓ, S), where

$$\begin{aligned} v &:= (v_i, i \in \mathcal{N}) \\ s &:= (s_i, i \in \mathcal{N}) \\ \ell &:= (\ell_{ij}, (i,j) \in \mathcal{E}) \\ S &:= (S_{ij}, (i,j) \in \mathcal{E}). \end{aligned}$$

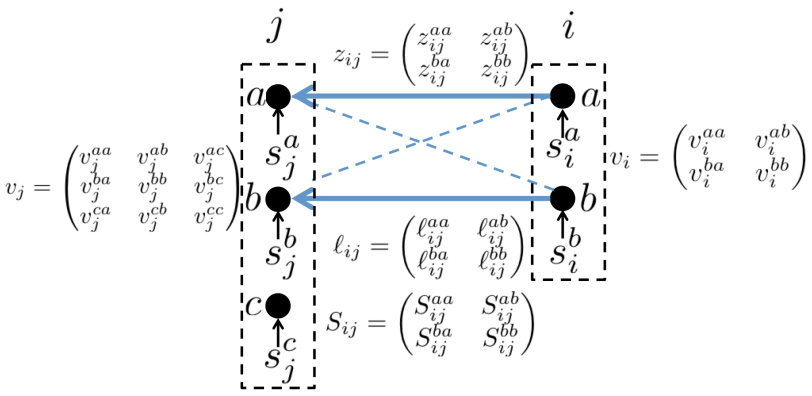


Figure 3.1: Notations for Unbalanced Networks.

Given a network $\mathcal{G}(\mathcal{N}, \mathcal{E})$, the branch flow model for unbalanced networks is defined by:

$$\mathcal{P}_i(v_j) - v_i + z_{ij} S_{ij}^H + S_{ij} z_{ij}^H - z_{ij} \ell_{ij} z_{ij}^H = 0 \qquad (i,j) \in \mathcal{E} \tag{3.11a}$$

$$\operatorname{diag}\left(\sum_{j \in C_i} \mathcal{P}_i(S_{ji} - z_{ji}\ell_{ji}) - \sum_{k \in A_i} S_{ik} \right) - s_i = 0 \qquad i \in \mathcal{N} \tag{3.11b}$$

$$\begin{pmatrix} v_i & S_{ij} \\ S_{ij}^H & \ell_{ij} \end{pmatrix} \in \mathbb{S}_+ \qquad (i,j) \in \mathcal{E} \tag{3.11c}$$

$$\operatorname{rank} \begin{pmatrix} v_i & S_{ij} \\ S_{ij}^H & \ell_{ij} \end{pmatrix} = 1 \qquad i \in \mathcal{E}, \tag{3.11d}$$

where $\mathcal{P}_i(v_j)$ denotes projecting v_j to the set of phases on bus i and $\mathcal{P}_i(S_{ji} - z_{ji}\ell_{ji})$ denotes lifting the result of $S_{ji} - z_{ji}\ell_{ji}$ to the set of phases Φ_i and filling the missing phase with 0.

Given a vector (v, s, ℓ, S) that satisfies (3.11), it is proved in [33] that the bus voltages V_i and branch currents I_{ij} can be uniquely determined if the network is a tree. Hence this model (3.11) is equivalent to a full unbalanced AC power flow model, i.e. BIM in (3.10). See [33, Section IV] for details.

3.2.2 OPF and SDP relaxation

Similar to the OPF problem on balanced networks, the OPF problem on unbalanced networks seeks to optimize certain objective, e.g. total power loss or generation cost, subject to unbalanced power flow equations (3.11) and various operational constraints. We consider an objective function of the following form:

$$F(s) := \sum_{i \in \mathcal{N}} f_i(s_i) := \sum_{i \in \mathcal{N}} \sum_{\phi \in \Phi_i} f_i^\phi(s_i^\phi). \tag{3.12}$$

For instance,

- to minimize total line loss, we can set for each $\phi \in \Phi_i$, $i \in \mathcal{N}$,

$$f_i^\phi(s_i^\phi) = p_i^\phi.$$

- to minimize generation cost, we can set for each $i \in \mathcal{N}$,

$$f_i^\phi(s_i^\phi) = (\frac{\alpha_i^\phi}{2}(p_i^\phi)^2 + \beta_i^\phi p_i^\phi),$$

  where $\alpha_i^\phi, \beta_i^\phi > 0$ depend on the load type on bus i, e.g. $\alpha_i^\phi = 0$ and $\beta_i^\phi = 0$ for bus i where there is no generator and for generator bus i, the corresponding $\alpha_i^\phi, \beta_i^\phi$ depends on the characteristic of the generator.

For each bus $i \in \mathcal{N}$, there are two operational constraints on each phase $\phi \in \Phi_i$. First, the power injection s_i^ϕ is constrained to be in a injection region $\mathcal{I}_i^\phi$, i.e.

$$s_i^\phi \in \mathcal{I}_i^\phi \quad \text{for } \phi \in \Phi_i \text{ and } i \in \mathcal{N}. \tag{3.13}$$

The feasible power injection region $\mathcal{I}_i^\phi$ is determined by the types of load attached to phase ϕ on bus i. Some common controllable loads are:

- For controllable load, whose real power varies within $[\underline{p}_i^\phi, \overline{p}_i^\phi]$ and reactive power varies within $[\underline{q}_i^\phi, \overline{q}_i^\phi]$, the injection region $\mathcal{I}_i^\phi$ is

$$\mathcal{I}_i^\phi = \{p + \mathbf{i}q \mid p \in [\underline{p}_i^\phi, \overline{p}_i^\phi], q \in [\underline{q}_i^\phi, \overline{q}_i^\phi]\} \subseteq \mathbb{C}. \tag{3.14a}$$

  For instance, the power injection of each phase ϕ on substation bus 0 is unconstrained, and thus $\underline{p}_i^\phi, \underline{q}_i^\phi = -\infty$ and $\overline{p}_i^\phi, \overline{q}_i^\phi = \infty$.

- For solar panel connecting the grid through a inverter with nameplate $\overline{s}_i^\phi$, the injection region $\mathcal{I}_i^\phi$ is

$$\mathcal{I}_i^\phi = \{p + \mathbf{i}q \mid p \geq 0, p^2 + q^2 \leq (\overline{s}_i^\phi)^2\} \subseteq \mathbb{C}. \tag{3.14b}$$

Second, the voltage magnitude needs to be maintained within a prescribed region. Note that the diagonal element of v_i describes the voltage magnitude square on each phase $\phi \in \Phi_i$. Thus the constraints can be written as

$$\underline{v}_i^\phi \leq v_i^{\phi\phi} \leq \overline{v}_i^\phi \quad i \in \mathcal{N}, \tag{3.15}$$

where $v_i^{\phi\phi}$ denotes the ϕ_{th} diagonal element of v_i. Typically the voltage magnitude at substation buses is assumed to be fixed at a prescribed value, i.e. $\underline{v}_i^{\phi} = \overline{v}_i^{\phi} = 1$ for $\phi \in \Phi_i$, $i \in \mathcal{N}_s$. At other load buses $i \in \mathcal{N} \setminus \mathcal{N}_s$, the voltage magnitude is typically allowed to deviate by 5% from its nominal value, i.e. $\underline{v}_i^{\phi} = 0.95^2$ and $\overline{v}_i^{\phi} = 1.05^2$ for $\phi \in \Phi_i$.

To summarize, the OPF problem for unbalanced multi-phase radial distribution networks is:

$$\begin{aligned} \textbf{OPF:} \ \min \quad & \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} f_i^{\phi}(s_i^{\phi}) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (3.11) \text{ and } (3.13) - (3.15). \end{aligned} \tag{3.16}$$

The OPF problem (3.16) is nonconvex due to the rank constraint (3.11d). In [33], an SDP relaxation for (3.16) is obtained by removing the rank constraint (3.11d), resulting in a semidefinite program (SDP):

$$\begin{aligned} \textbf{ROPF:} \ \min_{x} \quad & \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} f_i^{\phi}(s_i^{\phi}) \\ \text{over} \quad & v, s, S, \ell \\ \text{s.t.} \quad & (3.11a) - (3.11c) \text{ and } (3.13) - (3.15). \end{aligned} \tag{3.17}$$

Clearly the relaxation ROPF (3.17) provides a lower bound for the original OPF problem (3.16) since the original feasible set is enlarged. The relaxation is called *exact* if every optimal solution of ROPF satisfies the rank constraint (3.11d) and hence is also optimal for the original OPF problem. It is shown empirically in [33] that the relaxation is exact for all the tested distribution networks, including IEEE test networks [48] and some real distribution networks from Southern California Edison.

3.3 Conclusion

In this chapter, we first formulate the optimal power flow problem on balanced networks and review how to solve it through the second order cone relaxation. Then, we formulate the optimal power flow problem on unbalanced networks and show how to solve it through the semidefinite relaxation. The results in this chapter underlie our work on feeder reconfiguration and distributed algorithm for optimal power flow problems.

Chapter 4

Feeder Reconfiguration in Distribution Networks Based on Convex Relaxation of OPF

The feeder reconfiguration problem chooses the on/off status of the switches in a distribution network in order to minimize a certain cost such as power loss. It is a mixed integer nonlinear program and hence hard to solve. In this chapter we propose a heuristic algorithm that is based on the recently developed convex relaxation of the AC optimal power flow problem. The algorithm is computationally efficient and scales linearly with the number of redundant lines. It requires neither parameter tuning nor initialization for different networks. It successfully computes an optimal configuration on all four networks we have tested. Moreover we have derived a sub-optimality bound for the proposed algorithm under certain conditions for the case where only a single redundant line needs to be opened. We also propose a more computationally efficient algorithm and show that it incurs a loss in optimality of less than 3% on the four test networks.

Literature A primary distribution system consists of buses, distribution lines, and (sectionalizing and tie) switches that can be opened or closed. There are two types of buses: *Substation buses* (or just *substations*) that are connected to a transmission network from which they receive bulk power, and *load buses*[1] that receive power from the substation buses. During normal operation the switches are configured so that

1. There is no loop in the network.

2. Each load bus is connected to a single substation.

Therefore, there is a tree component rooted at each substation and we refer to each such component as a *feeder*. The optimal feeder reconfiguration (OFR) problem seeks to alter the on/off status of these switches, for the purpose of load balancing or loss minimization subject to the above two

[1]Distributed generations are viewed as loads with negative real power injections in this chapter.

requirements, e.g., [6, 17, 19, 65]. See also a survey in [76] for many early papers and references to some recent work in [45].

The OFR problem is a combinatorial (on/off status of switches) optimization problem with nonlinear constraints (power flow equations) and can generally be NP-hard. Various algorithms have been developed to solve the OFR problems. Following the convention in [45], they roughly fall into two categories: formal methods and heuristic methods.

Formal Methods solve the OFR problem using existing optimization approach. They usually require a significant amount of computation time. In [17], the problem is solved using a simulated annealing technique where the problem is formulated as a multi-objective mixed integer constrained optimization. In [24], ordinal optimization is proposed to reduce the computational burden through order comparison and goal softening. In [51], the problem is solved using generalized Benders decompositions. In [10], a mixed integer linear programming solver is applied to solve the problem after linearization of the power flow equations. In [39, 45], the problem is formulated as a mixed integer nonlinear program which is then solved as a mixed integer convex program through the second-order cone program (SOCP) relaxation.

Heuristic Methods exploit structural properties to solve the OFR problem. They are usually more computationally efficient than formal methods. In [19], an "iterative branch exchange approach" is applied to OFR. The network is initialized with a feasible topology. At each iteration, an opened switch is closed and a closed switch is opened to reduce the cost and maintain the radial structure. The algorithm stops once a local minimum is reached, i.e. for each currently opened switch, closing it and opening another switch will not further decrease the cost. See [6, 36] for further developments on this approach. This approach has the advantage that the intermediate configuration is always feasible, and hence we can terminate the algorithm at any iteration to obtain a feasible solution. However, the performance is sensitive to the initial configuration and sometimes it takes too many iterations for the algorithm to terminate. A different heuristic approach, first proposed in [65] and termed "successive branch reduction approach" in this chapter, assumes that all the switches are initially closed and they are sequentially opened based on a given criteria until a radial configuration is reached. This approach has two major advantages: 1) unlike the "iterative branch exchange approach", no initialization is required; and 2) the number of iterations is bounded by the number of redundant lines, which is usually small in practice. Some developments on this approach include relaxing the binary variable representing the status on the switch [35] and generalization to unbalanced network based on a constant current model [20].

Summary Optimal feeder reconfiguration is a mixed integer nonlinear optimization problem and therefore NP-hard in general. To overcome the first difficulty (mixed integer optimization), we propose a heuristic approach that only involves solving a small number of AC optimal power flow

(OPF) problems and *no* mixed-integer optimization. We theoretically show that the proposed heuristic can obtain the global optimal solution under certain assumptions. Indeed global optimal configurations can always be found on the four practical networks in our simulations. To overcome the second difficulty (nonconvexity of AC OPF), we build on the recent development of SOCP relaxation of AC OPF. The effectiveness of this new approach is illustrated both through simulations of standard test systems and mathematical analysis under certain assumptions. Specifically the main contributions of the chapter are twofold.

First, we propose an algorithm to optimize the "successive branch reduction approach". The algorithm uses a branch flow model introduced in [6, 7] for radial systems and exploits the recent development on solving the optimal power flow problem through convex relaxation [26, 27, 31]; see the tutorials in [60, 61] for more details. The algorithm has three major advantages:

1. Efficient: the complexity is linear in the number redundant lines that need to be opened.

2. Accurate: The algorithm is proved to solve OFR optimally under certain assumptions in the case where there is a single line that needs to be opened. Simulations on four practical networks show that it can find a globally optimal solution in the general case as well.

3. Hassle free: There are no parameters and initialization that need to be tuned for different networks.

Second, we simplify the above algorithm into one that has a constant complexity, i.e. the time complexity is independent of the number of redundant lines. Simulations on the same four practical networks show that the loss in optimality is less than 3%.

4.1 Problem Formulation

In this section, we define the optimal feeder reconfiguration (OFR) problem on a *balanced* distribution network.

4.1.1 Notations

In addition to the notations defined in Chapter 3, we introduce the following new notations, which will only be used in this chapter.

Given two real vectors $x, y \in \mathbb{R}^n$, $x \leq y$ means $x_i \leq y_i$ for $1 \leq i \leq n$ and $x < y$ means $x_i < y_i$ for at least one component. The Pareto front (See [12] for more properties) of a compact set $A \subseteq \mathbb{R}^n$ is defined as

$$\mathcal{O}(A) := \{x \in A \mid \nexists \tilde{x} \in A \setminus \{x\} \text{ such that } \tilde{x} \leq x\}. \tag{4.1}$$

Given a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$. Let $\mathcal{N}_s$ denote the set of substation buses, $\mathcal{N}_l$ denote the set of load buses and $\mathcal{N}_s \cup \mathcal{N}_l = \mathcal{N}$. For each line $(i, j) \in \mathcal{E}$, define S_{ji} in terms of S_{ij} and ℓ_{ij} by $S_{ji} := -S_{ij} + \ell_{ij} z_{ij}$. Hence $-S_{ji}$ represents the power received by bus j from bus i.

For each bus $i \in \mathcal{N}$, denote $E(i)$ the set of lines in $\mathcal{E}$ that has one end at i, i.e.

$$E(i) := \{(i, k) \mid i \in A_i, (i, k) \in \mathcal{E}\} \cup \{(j, i) \mid j \in C_i, (j, i) \in \mathcal{E}\}.$$

For any $\mathcal{E}' \subseteq \mathcal{E}$, a path exists between two nodes $i, j \in \mathcal{N}$ in graph $\mathcal{G}(\mathcal{N}, \mathcal{E}')$ if and only if there is a collection of edges in $\mathcal{E}'$ that connect node i and j. Denote

$$D^1_{\mathcal{E}'} :=\# \text{ of paths in } \mathcal{G}(\mathcal{N}, \mathcal{E}') \text{ among buses in } \mathcal{N}_s \tag{4.2}$$

$$D^2_{\mathcal{E}'} :=\# \text{ of loops in } \mathcal{G}(\mathcal{N}, \mathcal{E}') \tag{4.3}$$

$$D_{\mathcal{E}'} :=D^1_{\mathcal{E}'} + D^2_{\mathcal{E}'}. \tag{4.4}$$

4.1.2 Model and Problem formulation

There are sectionalizing or tie switches on the lines that can be opened or closed. Optimal feeder reconfiguration (OFR) is the problem of reconfiguring the switches to optimize certain objective subject to the topological constraints, power flow equations, and operational constraints on voltage magnitudes and power injections.

The objective function, power flow equations, and operational constraints for the OFR problem are the same as the OPF problem on *balanced* networks in section 3.1. There are two topological constraints on configuring the switches during normal operations:

1. Each load bus is connected to a single substation.

2. There is no loop in the network.

Any subset of lines $\mathcal{E}' \subseteq \mathcal{E}$ whose switches can be closed concurrently to satisfy both 1) and 2) is defined as a feasible configuration. Let

$$\mathcal{S}_T := \{\mathcal{E}_T \mid \mathcal{G}(\mathcal{N}, \mathcal{E}_T) \text{ satisfies 1) and 2)}\},$$

which represents the set of all feasible configurations.

When $|\mathcal{N}_s| = 1$, i.e. there is only one substation, $\mathcal{S}_T$ consists of the set of $\mathcal{E}_T$ such that $\mathcal{G}(\mathcal{N}, \mathcal{E}_T)$ is a spanning tree of $\mathcal{G}(\mathcal{N}, \mathcal{E})$.

Given a configuration line subset $\mathcal{E}' \subseteq \mathcal{E}$, define

- $\ell(\mathcal{E}')$ and $S(\mathcal{E}')$

$$\begin{aligned} \ell(\mathcal{E}') &:= \{\ell_{ij}, (i,j) \subseteq \mathcal{E}'\} \\ S(\mathcal{E}') &:= \{S_{ij}, (i,j) \subseteq \mathcal{E}'\}, \end{aligned}$$

  which represent the branch current and power flow on the distribution lines that are actively used in configuration $\mathcal{E}'$, i.e. $\ell(\mathcal{E}')$ ($S(\mathcal{E}')$) collects all the variables in ℓ (S) except the branch current ℓ_{ij} (branch power S_{ij}) for $(i,j) \in \mathcal{E} \setminus \mathcal{E}'$. Then denote

$$x(\mathcal{E}') := (v, s, \ell(\mathcal{E}'), S(\mathcal{E}')),$$

  which represents all the physical variables in configuration $\mathcal{E}'$ (The nodal variables v and s will always be used regardless of the configuration.).

- $\mathbb{X}(\mathcal{E}')$ and $\mathbb{X}_c(\mathcal{E}')$

$$\begin{aligned} \mathbb{X}(\mathcal{E}') &:= \{x(\mathcal{E}') \mid x(\mathcal{E}') \text{ satisfies (3.2), (3.4) and (3.6)}\} \\ \mathbb{X}_r(\mathcal{E}') &:= \{x(\mathcal{E}') \mid x(\mathcal{E}') \text{ satisfies (3.2a), (3.2b), (3.8) and (3.4), (3.6)}\}, \end{aligned}$$

  where $\mathbb{X}(\mathcal{E}')$ represents the feasible physical variables that satisfy the branch flow model (3.2), power injection constraints (3.4), and voltage constraints (3.6). $\mathbb{X}_r(\mathcal{E}')$ represents the feasible sets after SOCP relaxation, i.e. the feasible set of the ROPF problem (3.9).

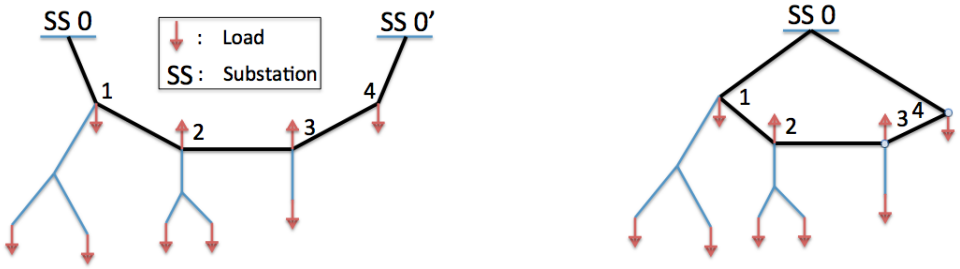
- OPF-$\mathcal{E}'$ and ROPF-$\mathcal{E}'$

$$\begin{aligned} \text{OPF}-\mathcal{E}': &\quad \min F(p) \ \text{ over } x(\mathcal{E}') \in \mathbb{X}(\mathcal{E}') \\ \text{ROPF}-\mathcal{E}': &\quad \min F(p) \ \text{ over } x(\mathcal{E}') \in \mathbb{X}_r(\mathcal{E}'), \end{aligned}$$

  which represent the OPF problem defined on configuration $\mathcal{E}'$ and the relaxed OPF problem, respectively.

Using the above notations, the OFR problem can be written as

$$\begin{aligned} \textbf{OFR}: \ \min &\quad F(p^*(\mathcal{E}_T)) \\ \text{over} &\quad \mathcal{E}_T \in \mathcal{S}_T, \end{aligned} \tag{4.5}$$



(a) $|\mathcal{N}_s| = 2$ and $|\mathcal{E}| = |\mathcal{N}| - 1$.

(b) $|\mathcal{N}_s| = 1$ and $|\mathcal{E}| = |\mathcal{N}|$.

Figure 4.1: Possible network topology with one redundant line.

where

$$x^*(\mathcal{E}_T) := \arg\min_x \{F(p) \ \text{ s.t. } x(\mathcal{E}_T) \in \mathbb{X}(\mathcal{E}_T)\}. \tag{4.6}$$

Different configurations $\mathcal{E}_T$ are implemented by different switch settings. OFR is difficult to solve due to the nonlinear feasible set $\mathbb{X}(\mathcal{E}_T)$ for a given configuration $\mathcal{E}_T$ and the discrete nature of $\mathcal{E}_T$. In section 3.1.2, SOCP relaxation is shown to be exact for the OPF problem under mild conditions if the underlying undirected graph is acyclic, which holds for all the feasible configuration $\mathcal{E}_T$. The proposed algorithm leverages the SOCP relaxation to deal with the nonlinearity. Throughout this chapter, we assume the SOCP relaxation is always exact. Then we have the following result of [31, Theorem 3], which will be useful for us.

Theorem 4.1. *Suppose the ROPF problem* (3.9) *is exact and the feasible set is nonempty. Then there exists a unique solution* $(v^*, \ell^*, \ell^*, S^*)$ *provided the objective function* $F(s)$ *is convex and non-decreasing in* $\mathbf{Re}(s)$ *and* $\mathbf{Im}(s)$.

4.2 Network Configuration with Single Redundant Line

In this section we consider the special case where there is only one redundant line that needs to be opened, i.e. $D_{\mathcal{E}} = 1$. We develop an algorithm to solve the OFR problem in this case and prove that the algorithm solves OFR optimally under certain assumptions. In addition, we simplify the above algorithm to reduce its computation complexity and incur negligible loss in optimality. We extend both algorithms to general networks in the next section.

4.2.1 Algorithms

When there is only one redundant line that needs to be opened, there are two possible cases as illustrated in Fig 4.1.

1. $|\mathcal{N}_s| = 2$ and $|\mathcal{E}| = |\mathcal{N}| - 1$, i.e. there are two substations and $|\mathcal{N}| - 1$ lines as shown in Figure 4.1a. Then each load bus is connected to two substations and we need to open one line from the path between the two substations.

2. $|\mathcal{N}_s| = 1$ and $|\mathcal{E}| = |\mathcal{N}|$, i.e. there is one substation and $|\mathcal{E}| = |\mathcal{N}|$ lines as in figure 4.1b. Then there exists a loop and we need to open one line to break the loop.

Algorithm 4.1 Network with one redundant line

$\mathcal{E}_T^* \leftarrow \mathcal{E}$
Solve OPF-$\mathcal{E}$ with an optima x^*
Calculate $\hat{e} \in \arg\min_e \{|P_e^*(\mathcal{E}_T^*)| \mid D_{\mathcal{E}_T^* \setminus e} = 0\}$
Denote $\hat{e} := (n_1, n_2)$
if $P_{\hat{e}} > 0$ **then**
    $e^* \leftarrow \arg\min_e \{F(p^*(\mathcal{E}_T^* \setminus e)) \mid e \in N(n_2)\}$
else
    $e^* \leftarrow \arg\min_e \{F(p^*(\mathcal{E}_T^* \setminus e)) \mid e \in N(n_1)\}$
end if
$\mathcal{E}_T^* \leftarrow \mathcal{E}_T^* \setminus e^*$
return $\mathcal{E}_T^*$

The algorithm to solve both cases in Fig. 4.1 is stated in Algorithm 4.1. The basic idea of Algorithm 4.1 is simple and we illustrate it using the line network in Fig. 4.2. For the line network in Fig. 4.2, let the buses at the two ends be substation buses and buses in between be load buses. Then $\mathcal{N}_s := \{0, 0'\}$, $\mathcal{N}_l := \{1, \ldots, n\}$ and $\mathcal{N} := \{0, 1, \ldots, n, 0'\}$. We use $n+1$ and $0'$ interchangeably for notational convenience.

For the line network shown in Figure 4.2, each load bus is connected to both substation 0 and $0'$, and thus the set of feasible configuration is given as

$$\mathcal{S}_T := \{\mathcal{E} \setminus (k, k+1) \mid 1 \le k \le n\},$$

i.e. each line in $\mathcal{E}$ can be opened to create a feasible configuration. For each bus k, the set of lines with one end at it is given as

$$E(k) = \begin{cases} \{(k, k+1), (k-1, k)\} & k \neq 0, n+1 \\ \{(0, 1)\} & k = 0 \\ \{(n, n+1)\} & k = n+1 \end{cases}.$$

In Algorithm 4.1, we first solve OPF-$\mathcal{E}$, which provides an optimal solution x^* assuming all the lines are closed. Then we search for a branch $\hat{e}$, whose branch power flow is minimum in $\mathcal{E}$. Denote $\hat{e} = (k, k+1)$ and the line we will open is based on the following criteria:

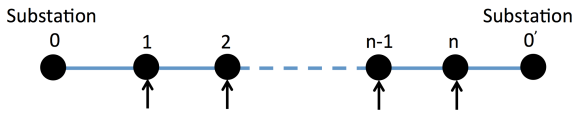


Figure 4.2: A line Network

1. $P_{\hat{e}} > 0$ and $k = n + 1$: There is only one candidate, i.e. $E(n + 1) = \{(n, n + 1)\}$ and line $(n, n + 1)$ is opened. It means substation $0'$ absorbs real power.

2. $P_{\hat{e}} > 0$ and $k < n+1$: There are two candidates, i.e. $E(k) = \{(k, k+1), (k+1, k+2)\}$. Either line $(k, k + 1)$ or $(k + 1, k + 2)$ is opened, depending on which gives a smaller objective value.

3. $P_{\hat{e}} \leq 0$ and $k = 0$: There is only one candidate, i.e. $E(0) = \{(0, 1)\}$ and line $(0, 1)$ is opened. It means substation 0 absorbs real power.

4. $P_{\hat{e}} \leq 0$ and $k > 0$: There are two candidates, i.e. $E(k + 1) = \{(k, k + 1), (k - 1, k)\}$. Either line $(k, k + 1)$ or $(k - 1, k)$ is opened, depending on which gives a smaller objective value.

The intuition behind Algorithm 4.1 is that the line which will be opened is close to the line where there is minimum branch flow power if we solve the problem assuming all the lines are closed (OPF-$\mathcal{E}$). Thus, we need to solve two other OPF problems for comparing the objective of the two candidates in addition to OPF-$\mathcal{E}$. Indeed, we can directly open the line with minimum branch power flow to simplify the algorithm after OPF-$\mathcal{E}$ is solved. By doing this, we sacrifice accuracy, but simulation results show that the solution of the corresponding algorithm incurs a similar cost as that of Algorithm 4.1. The simplified algorithm is stated in Algorithm 4.2.

Algorithm 4.2 Network with one redundant line (simplified)

1: Solve OPF-$\mathcal{E}$ with an optima x^*.
2: Calculate $\hat{e} \in \arg\min_e \{|P_e^*| \mid D_{\mathcal{E}\setminus e} = 0\}$
3: $\mathcal{E}_T^* \leftarrow \mathcal{E} \setminus \hat{e}$
4: **return** $\mathcal{E}_T^*$

4.2.2 Performance analysis

We analyze the performance of Algorithm 4.1, i.e. whether the configuration $\mathcal{E}_T^*$ returned by Algorithm 4.1 is optimal for OFR. There are two possible cases as illustrated in Fig. 4.1. Case (b) can be reduced to case (a) by replacing the substation 0 by two virtual substations 0 and $0'$ as shown in Fig. 4.1a, where $\mathcal{N}_s := \{0, 0'\}$, $\mathcal{N}_l := \{1, \ldots, n\}$. Thus, we only need to focus on case (a). For ease of presentation we only prove the results for a line network as shown in Fig. 4.2. They generalize in a straightforward manner to radial networks as shown in Fig. 4.1a. We make several assumptions below for our analysis:

A1 : $\overline{p}_i < 0$ for $i \in \mathcal{N}_l$ and $\overline{p}_i > 0$ for $i \in \mathcal{N}_s$.

A2 : $\overline{v}_i = \underline{v}_i = 1$ for $i \in \mathcal{N}$.

A3 : $|\theta_i - \theta_j| < \arctan(x_{ij}/r_{ij})$ for $(i,j) \in \mathcal{E}$.

A4 : The objective function $F(s) = F(p_0, p_{0'})$ is convex and increasing of $p_0, p_{0'}$.

A5 : The feasible set $\mathbb{X}(\mathcal{E})$ is compact.

A6 : The injection region $\mathcal{I}_i$ takes the box constraint defined in (3.5a).

A1 says that only substation buses 0 and $0'$ inject real power while load buses $1, \ldots, n$ absorb real power. A2 says that the voltage magnitude at each bus is fixed at their nominal value. A3 bounds the angle difference between adjacent buses.[2] A4 says that the objective function is merely a function of the power injections at two substations. A5 are technical assumptions that guarantees that our optimization problems are feasible. A6 says the control on real and reactive power injections can be decoupled.

The assumptions A1-A6 may not hold in practice, e.g. A1 is violated when there are distributed generators at some load buses, and A2 is violated when buses have limited reactive power injection capability. However, we only need A1-A6 to make precise statements about the performance of Algorithm 4.1. We will first explain the intuition before formally stating the result in Theorem 4.3.

We now rewrite the OFR problem (4.5) for the line network in Fig. 4.2. Some new notations will be defined which will only be used in this section. For any $(k, k+1) \in \mathcal{E}$, let $\mathcal{G}_0^k$ and $\mathcal{G}_{0'}^{k+1}$ represent the two subtrees rooted at 0 and $0'$, respectively, if line $(k, k+1)$ is opened. Denote

$$(p_0^k, p_{0'}^{k+1}) := \left(p_0^*(\mathcal{E}_T^{k,k+1}), p_{0'}^*(\mathcal{E}_T^{k,k+1})\right), \tag{4.7}$$

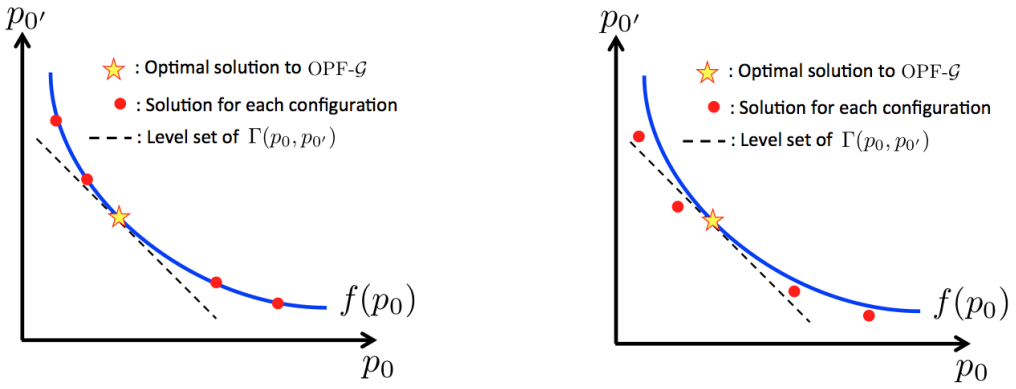
where $\mathcal{E}_T^{k,k+1} := \mathcal{E} \setminus (k, k+1)$ and $x^*(\mathcal{E}_T^{k,k+1})$ is the optimal solution to a given configuration $\mathcal{E}_T^{k,k+1}$ and defined in (4.6). $(p_0^k, p_{0'}^{k+1})$ represents the minimum power injection at the substations for the two subtrees $\mathcal{G}_0^k$ and $\mathcal{G}_{0'}^{k+1}$ after line $(k, k+1)$ is opened. Then the OFR problem (4.5) for the line network (Fig. 4.2) can be written equivalently as

$$\min_{0 \le k \le n} \quad F(p_0^k, p_{0'}^{k+1}). \tag{4.8}$$

Define an OPF problem:

$$\text{OPF-}\mathcal{E}\text{s:} \qquad f(p_0) := \min_{x \in \mathbb{X}(\mathcal{E})} \quad p_{0'} \qquad \text{s.t.} \quad p_0 \text{ is a given constant} \tag{4.9}$$

[2] Although voltage phase angles θ_i are relaxed in the relaxed branch flow model (3.2), they are uniquely determined by $\theta_i - \theta_j = \angle(v_i - z_{ij}^* S_{ij})$ in a radial network [27].



(a) Intuition under A1-A5.

(b) Intuition without A1-A5.

Figure 4.3: Intuitions of Algoirthm 4.1.

Recall that $\mathbb{X}(\mathcal{E})$ is the feasible set of physical variables given a configuration $\mathcal{E}$ and $\mathbb{X}_r(\mathcal{E})$ is the convexified $\mathbb{X}(\mathcal{E})$. Let $\mathbb{P} := \{(p_0, p_{0'}) \mid \exists x \in \mathbb{X}(\mathcal{E})\}$ represent the projection of $\mathbb{X}(\mathcal{E})$ on $\mathbb{R}^2$ and $\mathbb{P}_r := \{(p_0, p_{0'}) \mid \exists x \in \mathbb{X}_r(\mathcal{E})\}$ be the projection of $\mathbb{X}_c(\mathcal{E})$ on $\mathbb{R}^2$. By definition of the Pareto front in (4.1), the exactness of SOCP relaxation implies that $\mathcal{O}(\mathbb{P}) = \mathcal{O}(\mathbb{P}_c)$.

Lemma 4.2. *Suppose A4-A5 hold and the SOCP relaxation is exact. Then*

1. $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P}_r)$.
2. $f(p_0)$ *is a strictly convex decreasing function of* p_0.

By Lemma 4.2-1), $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P}_c)$, hence OPF-$\mathcal{E}$ can be written equivalently as

$$\min \quad F(p_0, f(p_0)). \tag{4.10}$$

In other words, solving OPF-$\mathcal{E}$ is equivalent to finding a point when the level set of $F(p_0, p_{0'})$ first hits the curve $(p_0, f(p_0))$ on a two-dimensional plane, where the x-axis and y-axis are the real power injections from substation 0 and $0'$, as shown in Fig. 4.3. On the other hand, the OFR problem can be written as (4.8) and solving OFR is equivalent to find a point when the level set of $F(p_0, p_{0'})$ first hits one point in $\{(p_0^k, p_{0'}^{k+1}) \mid 0 \le k \le n\}$ on the two-dimensional plane. Suppose A1-A5 hold, all the feasible points $(p_0^k, p_{0'}^{k+1})$ locate exactly on the curve $(p_0, f(p_0))$ as shown in Fig. 4.3a. Thus, we can obtain exactly the optimal solution to OFR by checking the points $(p_0^k, p_{0'}^{k+1})$ adjacent to the optimal solution to OPF-$\mathcal{E}$, which is performed in Algorithm 4.1. The result is formally stated in Theorem 4.3.

Theorem 4.3. *Suppose A1–A6 hold. Then the configuration* $\mathcal{E}_T^*$ *returned by Algorithm 4.1 is optimal for OFR* (4.5).

Remark: Theorem 4.3 shows that Algorithm 4.1 computes an optimal solution of OFR under assumptions A1-A6, which may not hold in practice. Without assuming A1-A6, $(p_0^k, p_{0'}^{k+1})$ does not locate exactly on the curve $(p_0, f(p_0))$ as shown in Fig. 4.3b. Thus, the points $(p_0^k, p_{0'}^{k+1})$ adjacent to the optimal solution to OPF-$\mathcal{E}$ may not be optimal for OFR. Indeed, we can create artificial examples to show that Algorithm 4.1 fails to find a global optimal configuration. However, the sub-optimality gap is usually small since the points $(p_0^k, p_{0'}^{k+1})$ are close to the the curve $(p_0, f(p_0))$, and global optimal configuration can always be found in our simulations on four practical networks.

In the following, we will derive the sub-optimality gap of Algorithm 4.1 by relaxing assumption A2 by A2+, which is still not realistic, but the result gives further intuition of the performance of the proposed algorithm.

A2+ : $\overline{v}_i = \underline{v}_i$, $\overline{q}_i = -\underline{q}_i = \infty$ for $i \in \mathcal{N}$

When the voltage magnitudes are fixed but different at different buses, Algorithm 1 is not guaranteed to find a global optimum of OFR. However, it still gives an excellent suboptimal solution to OFR. By nearly optimal, it means the suboptimality gap of Algorithm 4.1 is negligible.

Let $I_{p_0} := \{p_0 \mid \exists x \in \mathbb{X}(\mathcal{E})\}$ represent the projection of $\mathbb{X}(\mathcal{E})$ on real line. I_{p_0} is compact since $\mathbb{X}(\mathcal{E})$ is compact by A5. $f(p_0)$ is strictly convex and monotone decreasing by Corollary 4.2, it is right differentiable and denote its right derivative by $f'_+(p_0)$, which is monotone increasing and right differentiable and denote its right derivative by $f''_{++}(p_0)$. Let

$$\kappa_f := \inf_{p_0 \in I_{p_0}} f''_{++}(p_0) \geq 0. \tag{4.11}$$

κ_f represents the minimal value of the curvature on a compact interval if $f(p_0)$ is twice differentiable.

Define L_k for each line $(k, k+1) \in \mathcal{E}(0, 0')$ as sequel.

$$L_k := \frac{\delta v_k^2 \, r_{k,k+1}/|z_{k,k+1}|^2}{(v_k + v_{k+1}) + \sqrt{(v_k + v_{k+1})^2 - \delta v_k^2 \left(\frac{r_{k,k+1}^2}{x_{k,k+1}^2} + 1\right)}},$$

where $\delta v_k := v_k - v_{k+1}$. L_k represents the thermal loss of line $(k, k+1)$ when either $P_{k,k+1}$ or $P_{k+1,k}$ is 0. Conceptually it means all the real power sending from bus on one end of the line is converted to thermal loss and the other bus receives 0 real power, namely either $P_{k,k+1} = \ell_{k,k+1} r_{k,k+1}$ or $P_{k+1,k} = \ell_{k,k+1} r_{k,k+1}$. Then the expression of $L_k = \ell_{k,k+1} r_{k,k+1}$ can be obtained by substituting either $P_{k,k+1} = \ell_{k,k+1} r_{k,k+1}$ or $P_{k+1,k} = \ell_{k,k+1} r_{k,k+1}$ into (3.2a) and (3.2c). L_k is negligible compared to the power consumption of a load in a distribution system. Therefore the ratio of these two quantities, defined as $R_k := -\overline{p}_{k+1}/L_k$, is usually quite large.

Let $R := \min R_k$ and κ_f as defined in (4.11), which is a constant depending on the network. Let F^* be the optimal objective value of OFR and F_A be the objective value if we open the line e^* given

by Algorithm 4.1.

Theorem 4.4. *Suppose A1,A2+,A3-A6 hold and for all $i \in \mathcal{N}$. Then*

$$F^* \leq F_A \leq F^* + \max\left\{\frac{c_0^2}{c_{0'}}, \frac{c_{0'}^2}{c_0}\right\}\frac{2}{R^2\kappa_f},$$

if $F(p_0, p_{0'}) := c_0 p_0 + c_{0'} p_{0'}$ for some positive $c_0, c_{0'}$.

Remark: R is large, usually on the order of 10^3, in a distribution system when there is no renewable generation. Although it is difficult to estimate the value of κ_f in theory, our simulation shows that κ_f is typically around $0.025MW^{-1}$ for a feeder with loop size of 10, and thus the bound is approximately $80W$ if $c_0 = c_{0'} = 1$, which is quite small. Moreover, simulations of two SCE distribution circuits show that Algorithm 4.1 always finds the global optima of OFR problem; see section 4.4. Therefore the bound in the theorem, already negligible, is not always tight.

4.3 General network configuration

In section 4.2, we propose two algorithms to solve the OFR problem assuming there is only one redundant line that needs to be open. In this section, we will extend both Algorithms 4.1 and 4.2 to general networks where there may be more than one redundant line that need to be opened. As before, one of the algorithms has a higher accuracy but requires more computation (Algorithm 4.3) and the other lower accuracy but less computation (Algorithm 4.4).

Loosely speaking, Algorithm 4.1 consists of the following procedure:

1. Solve OPF problem assuming all the lines are closed.
2. Find the line $\hat{e}$ with minimum branch power flow.
3. Check line $\hat{e}$ against the lines adjacent to line $\hat{e}$ and the minimum of those lines as a solution.

For a general network, there are multiple lines that need to be simultaneously open. Then we generalize Algorithm 4.1 in the following manner: we iterate the procedure in Algorithm 4.1 and remove one line from $\mathcal{E}$ at the end of each iteration, resulting in a different OPF problem to solve for the next iteration. There are $|\mathcal{E}| - |\mathcal{N}_l|$ redundant lines and hence $|\mathcal{E}| - |\mathcal{N}_l|$ iterations. The algorithm is formally stated in Algorithm 4.3.

Similarly, we can mimic Algorithm 4.2 and have an efficient algorithm which merely solves one OPF problem. Algorithm 4.2 consists of the following procedure:

1. Solve OPF problem assuming all the lines are closed.
2. Open the line $\hat{e}$ with minimum branch power flow.

Algorithm 4.3 General Network Reconfiguration

$\mathcal{E}_T^* \leftarrow \mathcal{E}$
while $D_{\mathcal{E}_T^*} > 0$ **do**
  Solve OPF-$\mathcal{E}_T^*$ with optima $x^*(\mathcal{E}_T^*)$
  Calculate $\hat{e} \in \arg\min_e\{|P_e^*(\mathcal{E}_T^*)| \mid D_{\mathcal{E}_T^* \setminus e} < D_{\mathcal{E}_T^*}\}$
  Denote $\hat{e} := (n_1, n_2)$
  if $P_{\hat{e}} > 0$ **then**
    $e^* \leftarrow \arg\min_e\{F(p^*(\mathcal{E}_T^* \setminus e)) \mid e \in C(n_2) \cap \mathcal{E}_T^*\}$
  else
    $e^* \leftarrow \arg\min_e\{F(p^*(\mathcal{E}_T^* \setminus e)) \mid e \in C(n_1) \cap \mathcal{E}_T^*\}$
  end if
  $\mathcal{E}_T^* \leftarrow \mathcal{E}_T^* \setminus e^*$
end while
return $\mathcal{E}_T^*$

We generalize Algorithm 4.2 in the following manner. We solve only one OPF problem OPF-$\mathcal{E}$, which assumes all the lines are closed. Then we sequentially choose one line with the smallest branch power flow in the remaining closed lines merely based on the solution to OPF-$\mathcal{E}$. Our simulations show that the simplification leads to negligible loss in optimality compared to Algorithm 4.3. The algorithm is stated in Algorithm 4.4.

Algorithm 4.4 General Network Reconfiguration (simplified)

$\mathcal{E}_T^* \leftarrow \mathcal{E}$
Solve OPF-$\mathcal{E}$; let x^* be an optimal solution.
while $D_{\mathcal{E}_T^*} > 0$ **do**
  Calculate $\hat{e} \in \arg\min_e\{|P_e^*| \mid D_{\mathcal{E}_T^* \setminus e} < D_{\mathcal{E}_T^*}\}$
  $\mathcal{E}_T^* \leftarrow \mathcal{E}_T^* \setminus \hat{e}$
end while
return $\mathcal{E}_T^*$

Remark: Algorithm 4.3 scales linearly with the number of redundant lines and Algorithm 4.4 is independent of the number of redundant lines. For large distribution system, solving one OPF problem requires a significant amount of time and Algorithm 4.4 can greatly reduce the computation time if there are many redundant lines.

4.4 Simulations

In this section we present examples to illustrate the effectiveness of the algorithms proposed in section 4.3 (the algorithms in section 4.2 are special cases). We used a Macbook Pro with 2.9 Ghz Intel Core i7 and 8GB memory. The algorithms are implemented in Matlab 2013a and the OPF problem is solved using Gurobi optimization solver.

We test the algorithms on four practical distribution networks. Test network 1 is from Taiwan Power Company and the network data is taken from [81]. Test network 2 is from Brazil and

Table 4.1: Network of Fig. 4.1: Line impedances, peak spot load KVA, Capacitors and PV generation's nameplate ratings.

Network Data																	
Line Data				Line Data				Line Data				Load Data		Load Data		PV Generators	
From Bus.	To Bus.	R (Ω)	X (Ω)	From Bus.	To Bus.	R (Ω)	X (Ω)	From Bus.	To Bus.	R (Ω)	X (Ω)	Bus No.	Peak MVA	Bus No.	Peak MVAR	Bus No.	Nameplate Capacity
1	2	0.259	0.808	8	41	0.107	0.031	21	22	0.198	0.046	1	30	34	0.2		
2	13	0	0	8	35	0.076	0.015	22	23	0	0	11	0.67	36	0.27	13	1.5MW
2	3	0.031	0.092	8	9	0.031	0.031	27	31	0.046	0.015	12	0.45	38	0.45	17	0.4MW
3	4	0.046	0.092	9	10	0.015	0.015	27	28	0.107	0.031	14	0.89	39	1.34	19	1.5 MW
3	14	0.092	0.031	9	42	0.153	0.046	28	29	0.107	0.031	16	0.07	40	0.13	23	1 MW
3	15	0.214	0.046	10	11	0.107	0.076	29	30	0.061	0.015	18	0.67	41	0.67	24	2 MW
4	20	0.336	0.061	10	46	0.229	0.122	32	33	0.046	0.015	21	0.45	42	0.13		
4	5	0.107	0.183	11	47	0.031	0.015	33	34	0.031	0.015	22	2.23	44	0.45	Shunt Capacitors	
5	26	0.061	0.015	11	12	0.076	0.046	35	36	0.076	0.015	25	0.45	45	0.2	Bus No.	Nameplate Mvar
5	6	0.015	0.031	15	18	0.046	0.015	35	37	0.076	0.046	26	0.2	46	0.45		
6	27	0.168	0.061	15	16	0.107	0.015	35	38	0.107	0.015	28	0.13				
6	7	0.031	0.046	16	17	0	0	42	43	0.061	0.015	29	0.13	V_{base}= 12.35kv		1	6
7	32	0.076	0.015	18	19	0	0	43	44	0.061	0.015	30	0.2	S_{base} = 1MW		3	1.2
7	8	0.015	0.015	20	21	0.122	0.092	43	45	0.061	0.015	31	0.07			37	1.8
8	40	0.046	0.015	20	25	0.214	0.046	1	12	0.076	0.146	32	0.13			47	1.8
8	39	0.244	0.046	21	24	0	0	1	30	0.116	0.146	33	0.27				

the network data is taken from [63]. There are no renewable generations in these two networks. Test networks 3 and 4 are from Southern California Edison with renewable generations and taken from [26,28]. Since the original data on these two networks consist of a single substation and contain no loop, we make several modifications to add loops in order to test our algorithms. The modified circuit diagram and network data of test network 3 are shown in Fig. 4.1 and Table 4.1. The modified circuit diagram and network data of test network 4 are shown in Fig. 4.2 and Table 4.4.

In the simulations, the voltage magnitude of the substations is fixed at 1 p.u. The voltage magnitudes at all other buses are allowed to vary within $[0.95, 1.05]$p.u. Our objective is to minimize the power loss, i.e. $\alpha_i = 0, \beta_i = 1$ for $i \in \mathcal{N}$ in (3.3), which means $F(p) := \sum_{i\in\mathcal{N}} p_i$. For all four networks, Algorithm 4.3 always computes an optimal configuration and Algorithm 4.4 computes a configuration with only up to 3% loss in optimality.

4.4.1 Case I: Tai-83 Bus System [81]

The Tai-83 bus system consists of 96 lines and 13 of them needs to be kept open to satisfy the configuration requirement. This network has been tested in [15, 18, 45, 80, 81, 87] using different approaches. In [45], Jabr, *et. al* show that opening lines (7, 13, 34, 39, 42, 55, 62, 72, 83, 86, 89, 90, 92) gives an optimal solution using mixed integer convex programming solver. The results are summarized in Table 4.3, where we also show the loss reduction[3], which represents the relative saving on power loss due to reconfiguration.

We run both Algorithm 4.3 and 4.4 for this network. Algorithm 4.3 returns the same optimal solution as [15,45,87]. However, Algorithm 4.3 is computationally very efficient since we only solved 39 OPF problems, which take 0.94 seconds on a laptop (MacBookpro). Algorithm 4.4 opens lines (7, 13, 33, 39, 42, 63, 72, 82, 84, 86, 89, 90, 92) with a power loss of 471.39KW. Compared with the optimal solution of 469.88KW, the difference in the power loss is less than 0.4% but we only need to solve 1 OPF problem, which takes 0.024 second on a laptop.

[3]loss reduction=$1 - \frac{\text{power loss (after reconfiguration)}}{\text{power loss (before reconfiguration)}}$

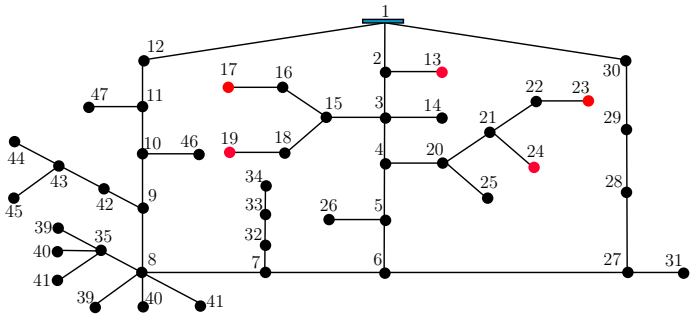


Figure 4.1: A modified SCE 47-bus feeder. The blue bar (1) represents the substation bus, the red dots (13, 17, 19, 23, 24) represent buses with PV panels, and the other dots represent load buses without PV panels.

Table 4.2: Summary on Brazil-135 Bus System

Method	Opened Lines	Losses (KW)	Loss reduction
[63]	51 ,106 ,136 ,137 ,138 ,139 ,141 ,142 ,143 ,144 ,145 ,146 ,147 ,148 ,149 ,150 ,151 ,152 ,154 ,155 ,156	285.77	10.80%
[15, 45]	7 ,35 ,51 ,90 ,96 ,106 ,118 ,126 ,135 ,137 ,138 ,141 ,142 ,144 ,145 ,146 ,147 ,148 ,150 ,151 ,155	280.19	12.54%
Algorithm 3	7 ,35 ,51 ,90 ,96 ,106 ,118 ,126 ,135 ,137 ,138 ,141 ,142 ,144 ,145 ,146 ,147 ,148 ,150 ,151 ,155	280.19	12.54%
Algorithm 4	35 ,51 ,55 ,84 ,90 ,106 ,126 ,135 ,136 ,137 ,138 ,141 ,143 ,144 ,145 ,147 ,148 ,152 ,150 ,151 ,155	288.01	10.10%

4.4.2 Case II: Brazil-135 Bus System [63]

The Brazil-135 bus system consists of 156 lines and 21 of them needs to be kept open to satisfy the configuration requirement. This network has been tested in [15, 45, 63] using different approaches. In [45], Jabr, *et. al* show that opening lines 7, 35, 51, 90, 96, 106, 118, 126, 135, 137, 138, 141, 142, 144, 145, 146, 147, 148, 150, 151, and 155 gives an optimal solution using mixed integer convex programming solver. The results are summarized in Table 4.2.

Algorithm 4.3 computes the same optimal solution as [15,45]. However, Algorithm 4.3 is computationally very efficient since we only solved 63 OPF problems, which take 2.2 seconds on a laptop. Algorithm 4.4 opens lines 35, 51, 55, 84, 90, 106, 126, 135, 136, 137, 138, 141, 143, 144, 145, 147, 148, 152, 150, 151, and 155 with a power loss of 288.01KW. Compared with the optimal solution of 280.19KW, the difference in the power loss is less than 2.8% but we only need to solve 1 OPF problem, which takes 0.055 seconds on a laptop.

Table 4.3: Summary on Tai-83 Bus System

Method	Opened Lines	Losses (KW)	Loss reduction
[18, 80, 81]	7 ,13 ,34 ,39 ,41 ,55 ,62 ,72 ,83 ,86 ,89 ,90 ,92	471.08	11.45%
[15, 45, 87]	7 ,13 ,34 ,39 ,42 ,55 ,62 ,72 ,83 ,86 ,89 ,90 ,92	469.88	11.68%
Algorithm 3	7 ,13 ,34 ,39 ,42 ,55 ,62 ,72 ,83 ,86 ,89 ,90 ,92	469.88	11.68%
Algorithm 4	7, 13 ,33 ,39 ,42 ,63 ,72 ,82 ,84 ,86 ,89 ,90 ,92	471.39	11.40%

Table 4.4: Network of Fig. 4.2: Line impedances, peak spot load KVA, Capacitors and PV generation's nameplate ratings.

Network Data																	
Line Data				Line Data				Line Data				Load Data		Load Data		Load Data	
From Bus.	To Bus.	R (Ω)	X (Ω)	From Bus.	To Bus.	R (Ω)	X (Ω)	From Bus.	To Bus.	R (Ω)	X (Ω)	Bus No.	Peak MVA	Bus No.	Peak MVA	Bus No.	Peak MVA
1	2	0.160	0.388	20	21	0.251	0.096	39	40	2.349	0.964	3	0.057	29	0.044	52	0.315
2	3	0.824	0.315	21	22	1.818	0.695	34	41	0.115	0.278	5	0.121	31	0.053	54	0.061
2	4	0.144	0.349	20	23	0.225	0.542	41	42	0.159	0.384	6	0.049	32	0.223	55	0.055
4	5	1.026	0.421	23	24	0.127	0.028	42	43	0.934	0.383	7	0.053	33	0.123	56	0.130
4	6	0.741	0.466	23	25	0.284	0.687	42	44	0.506	0.163	8	0.047	34	0.067	Shunt Cap	
4	7	0.528	0.468	25	26	0.171	0.414	42	45	0.095	0.195	9	0.068	35	0.094	Bus	Mvar
7	8	0.358	0.314	26	27	0.414	0.386	42	46	1.915	0.769	10	0.048	36	0.097	19	0.6
8	9	2.032	0.798	27	28	0.210	0.196	41	47	0.157	0.379	11	0.067	37	0.281	21	0.6
8	10	0.502	0.441	28	29	0.395	0.369	47	48	1.641	0.670	12	0.094	38	0.117	30	0.6
10	11	0.372	0.327	29	30	0.248	0.232	47	49	0.081	0.196	14	0.057	39	0.131	53	0.6
11	12	1.431	0.999	30	31	0.279	0.260	49	50	1.727	0.709	16	0.053	40	0.030	Photovoltaic	
11	13	0.429	0.377	26	32	0.205	0.495	49	51	0.112	0.270	17	0.057	41	0.046	Bus	Capacity
13	14	0.671	0.257	32	33	0.263	0.073	51	52	0.674	0.275	18	0.112	42	0.054		
13	15	0.457	0.401	32	34	0.071	0.171	51	53	0.070	0.170	19	0.087	43	0.083	45	5MW
15	16	1.008	0.385	34	35	0.625	0.273	53	54	2.041	0.780	22	0.063	44	0.057		
15	17	0.153	0.134	34	36	0.510	0.209	53	55	0.813	0.334	24	0.135	46	0.134	V_{base} = 12kV	
17	18	0.971	0.722	36	37	2.018	0.829	53	56	0.141	0.340	25	0.100	47	0.045	S_{base} = 1MVA	
18	19	1.885	0.721	34	38	1.062	0.406	1	32	0.113	0.434	27	0.048	48	0.196	Z_{base} = 144Ω	
4	20	0.138	0.334	38	39	0.610	0.238	53	57	0.1	0.3	28	0.038	50	0.045		
19	58	0.09	0.2														

4.4.3 Case III: SCE-47 Bus System

The original data for the SCE 47-bus system does not contain loops, so we added two lines to connect the substation bus 1 to two load buses, 12 and 30, respectively. Thus there are 49 lines and 2 of them needs to be open in the modified feeder. In addition to the loads, there are 5 PV panels and their power injections can be controlled. The nameplates for these 5 PV panels can be found in Table 4.1.

There are in total 95 feasible configurations. We first calculate the objective value of all the 95 configurations. The best configuration is opening lines $\{(1,2),(8,9)\}$, resulting in 32.6KW power loss. The average power loss is 63.7KW and the worst configuration's power loss is 136.9KW across the 95 configurations. Thus the average power loss is almost twice as bad as the minimum power loss and the worst configuration is 4 times as bad as the minimum!

Both Algorithms 4.3 and 4.4 find the optimal configuration for this network. Algorithm 4.3 solves 4 OPF problems (0.055 second) and Algorithm 4.4 solves 1 OPF problem (0.014 second). Compared with solving one OPF problem for each configuration to obtain the optimal solution, both algorithms are much more efficient without any loss in optimality.

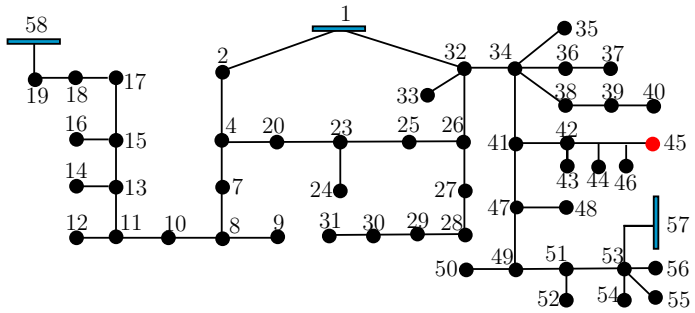


Figure 4.2: A modified SCE 56-bus feeder. The blue bars (1, 57, 58) represent the substation buses and the red dot (45) represents the bus with PV panels.

4.4.4 Case IV: SCE-56 Bus System

In contrast to the SCE-47 system, where there are 5 relatively small PV panels, the SCE 56-bus system consists of a single big PV system with a capacity of 5MW. We make the following modifications:

- We add a line between bus 1 and bus 32 to create a loop.
- We assume there are two additional substations (bus 57 and 58): attached to substation 19 and 53, respectively.

There are 59 lines and 3 lines need to be kept open. There are in total 724 feasible configurations. We first calculate the objective value of all the 724 configurations. The best configuration is opening lines $\{(11, 13), (23, 25), (41, 47)\}$, resulting in 9.89KW power loss. The average power loss is 23.4KW and the worst power loss is 211KW across the 724 configurations.

We run both Algorithm 4.3 and 4.4 for this network. Algorithm 4.3 computes the optimal solution by opening lines $\{(11, 13), (23, 25), (41, 47)\}$ but solves just 9 OPF problems, which take 0.14 seconds. Algorithm 4.4 opens lines $\{(11, 13), (23, 25), (47, 49)\}$ with a power loss of 9.92KW. Compared with the optimal solution of 9.89KW, the difference in the power loss is less than 0.3% but Algorithm 4.4 only needs to solve one OPF problem, which takes 0.015 seconds.

4.5 Conclusion

We propose two algorithms with different tradeoffs on efficiency and accuracy for feeder reconfiguration, based on the SOCP relaxation of OPF. We derive a sub-optimality gap of the proposed algorithm under certain conditions and argue that the gap is usually small for practical networks. We also demonstrate the effectiveness of our algorithms through simulations on four practical networks.

Appendix

4.A Proof of Lemma 4.2

We will first show that $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P}_r)$. Since $\mathcal{O}(\mathbb{P}_r) = \mathcal{O}(\mathbb{P})$, it is equivalent to prove $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P})$.

By property of *Pareto Front* [12], for each point $(p_0, p_{0'}) \in \mathcal{O}(\mathbb{P}_r) = \mathcal{O}(\mathbb{P})$, there exists a convex nondecreasing function $F^* : \mathbb{R}^2 \to \mathbb{R}$ such that $(p_0, p_{0'})$ is an optima for OPF-$\mathcal{E}$. Given any $(p_0, \hat{p}_{0'}) \in \mathcal{O}(\mathbb{P})$, let $F^*(p_0, p_{0'})$ be the objective function such that $(p_0, \hat{p}_{0'})$ solves OPF-$\mathcal{E}$. Since $F^*(p_0, p_{0'})$ is a nondecreasing function, OPF-$\mathcal{E}$ can be written equivalently as

$$\begin{aligned} \min_{p_0} \quad & F^*(p_0, p_{0'}) \\ \text{s.t.} \quad & p_{0'} = f(p_0). \end{aligned}$$

Therefore, at optimality, $f(p_0) = \hat{p}_{0'}$ and $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P})$.

Next, we prove $f(p_0)$ is a strictly convex decreasing function of p_0.

Lemma 4.5. *Let A be a compact and convex set in $\mathbb{R}^2$. Define $g(x) := y$ for any $(x, y) \in \mathcal{O}(A)$. Then $y = g(x)$ is a convex decreasing function of x for $(x, y) \in \mathcal{O}(A)$.*

Proof. We first show $g(x)$ is a decreasing function and then show $g(x)$ is also convex.

Let $(x_1, g(x_1))$ and $(x_2, g(x_2))$ be two points in $\mathcal{O}(A)$. Without loss of generality, assume $x_1 > x_2$. If $g(x_1) \geq g(x_2)$, it violates the fact that $(x_1, g(x_1)) \in \mathcal{O}(A)$ and hence $g(x_1) < g(x_2)$, which means that $g(x)$ is a decreasing function.

Next, we will show $g(\cdot)$ is convex. Recall that A is a compact set, we have $(x_1, g(x_1)), (x_2, g(x_2)) \in \mathcal{O}(A) \subseteq A$. A is also a convex set, and thus $(\frac{x_1+x_2}{2}, \frac{g(x_1)+g(x_2)}{2}) \in A$. By definition of Pareto front,

$$g(\frac{x_1 + x_2}{2}) = \inf_{(\frac{x_1+x_2}{2}, y) \in A} \{y\} \leq \frac{g(x_1) + g(x_2)}{2},$$

which shows $g(x)$ is a convex function. □

Since $\mathbb{X}_c(\mathcal{E})$ is convex and compact by A5, its projection on a two dimensional space $\mathbb{P}_r$ is also compact and convex. Note that $(p_0, f(p_0)) \in \mathcal{O}(\mathbb{P}_c)$ by part 1) of Lemma 4.2, we have part 2).

4.B Proof of Theorem 4.3

For the line network in Fig. 4.2, denote $\hat{e} = (\hat{k}, \hat{k}+1)$ (Line 4 in Algorithm 1). Without loss of generality, assume $P_{\hat{k},\hat{k}+1} > 0$ and we need to show that e^* is the optimal line to open for OFR, i.e. either $(\hat{k}, \hat{k}+1)$ or $(\hat{k}+1, \hat{k}+2)$ will be opened for the optimal solution.

Based on Theorem 4.1, there exists a unique solution x^* for any OPF problems with convex nondecreasing objective function. Therefore there is also a unique solution x^* to OPF-$\mathcal{E}s$ for any feasible real power injection p_0 at substation 0. In other words, x^* is a function of p_0 and let $x(p_0) := (s^*, S^*)$ represents the solution to OPF-$\mathcal{E}s$ with real power injection p_0 at substation 0. We skip v and ℓ in x since v_i is fixed by A2 and $\ell_{k,k+1}$ is uniquely determined by $S_{k,k+1}$ according to (3.2c). By Maximum theorem, $x(p_0)$ is a continuous function of p_0.

Lemma 4.6. *Suppose A2-A6 hold. Then $P_{k,k+1}(p_0)$ is an increasing function of p_0 for all $(k, k+1) \in \mathcal{E}$.*

Proof. See Appendix 4.C for the proof. □

Since $P_{k,k+1}(p_0)$ is an increasing and continuous function of p_0, there exists a unique p_0 to $P_{k,k+1}(p_0) = 0$ and denote $p_0(k) := P_{k,k+1}^{-1}(0)$, i.e. $P_{k,k+1}(p_0(k)) = 0$.

Lemma 4.7. *Suppose A1-A6 hold. Then $p_0(k) < p_0(k+1)$ for $0 \le k \le n$.*

Proof. By A1 ($\overline{p}_i < 0$ for $i \in \mathcal{N}_l$) and (3.2b), we have for $0 \le k < n$,

$$P_{k,k+1}(p_0) = P_{k+1,k+2}(p_0) - p_{k+1} > P_{k+1,k+2}(p_0),$$

which means

$$0 = P_{k,k+1}(p_0(k)) > P_{k+1,k+2}(p_0(k)).$$

By Lemma 4.6, $P_{k+1,k+2}(p_0)$ is a increasing function of p_0, hence $p_0(k) < p_0(k+1)$. □

Recall that $(p_0^k, p_{0'}^{k+1})$ is the minimal real power injection for subtrees $\mathcal{G}_0^k$ and $\mathcal{G}_{0'}^{k+1}$, respectively. Our next result shows that $(p_0^k, p_{0'}^{k+1}) = (p_0(k), f(p_0(k)))$.

Lemma 4.8. *Suppose A2–A6 hold, then*

$$(p_0^k, p_{0'}^{k+1}) = (p_0(k), f(p_0(k))) \quad (k, k+1) \in \mathcal{E}.$$

Proof. By definition, $P_{k,k+1}(p_0(k)) = 0$. Then, we have

$$2x_{k,k+1}Q_{k,k+1}(p_0(k)) = \ell_{k,k+1}(p_0(k))|z_{k,k+1}|^2$$
$$\ell_{k,k+1}(p_0(k)) = \frac{Q^2_{k,k+1}(p_0(k))}{v_k}$$

if $v_k = v_{k+1}$ based on (3.2b) and (3.2c). Solving the above two equations gives $Q_{k,k+1}(p_0(k)) = 0$. Therefore, $p_0(k)$ is a feasible power injection for subtree $\mathcal{G}_0^k$ and it means $p_0^k \leq p_0(k)$. Next, we will show that $p_0(k)$ is the smallest possible power injection for $\mathcal{G}_0^k$. Suppose we have $p_0^k < p_0(k)$, then $(p_0^k, f(p_0(k)))$ is a feasible power injection for network $\mathcal{G}$ with $p_0^k < p_0(k)$. It contradicts $(p_0(k), f(p_0(k))) \in \mathcal{O}(\mathbb{P})$ (Lemma 4.2). Therefore we have $p_0^k = p_0(k)$ and $p_{0'}^{k+1} = f(p_0(k))$. □

By Lemma 4.2, $(p_0(k), f(p_0(k))) \in \mathcal{O}(\mathbb{P})$. Then Lemma 4.8 means the minimal power injection for each partition of graph $\mathcal{G}$ locates exactly on the Pareto front of the feasible power injection region of OPF-$\mathcal{E}$. Therefore the OFR problem (4.8) is equivalent to the following problem:

$$\min_{0\leq k\leq n} F(p_0^k, p_{0'}^{k+1}) = \min_{0\leq k\leq n} F(p_0(k), f(p_0(k))), \tag{4.12}$$

whose minimizer is denoted by k^*.

On the other hand, by (4.10), OPF-$\mathcal{E}$ can be rewritten as

$$\min_{p_0} F(p_0, f(p_0)),$$

whose unique minimizer is denoted by p_0^*.

Lemma 4.9. *Suppose A1-A6 and $P_{\hat{k},\hat{k}+1}(p_0^*) > 0$ hold. Then,*

$$p_0(\hat{k}) \leq p_0^* \leq p_0(\hat{k}+1).$$

Proof. By our assumption at the beginning of the proof, $P_{\hat{k},\hat{k}+1}(p_0^*) > 0$, which implies that $p_0(\hat{k}) \leq p_0^*$. since $P_{\hat{k},\hat{k}+1}(p_0)$ is a increasing function of p_0 based on Lemma 4.6. On the other hand, by A1

$$P_{\hat{k},\hat{k}+1}(p_0^*) = P_{\hat{k}+1,\hat{k}+2}(p_0^*) - p_{\hat{k}+1}(p_0^*) > P_{\hat{k}+1,\hat{k}+2}(p_0^*),$$

implying that $P_{\hat{k}+1,\hat{k}+2}(p_0^*) < 0$. Otherwise $|P_{\hat{k},\hat{k}+1}(p_0^*)| > |P_{\hat{k}+1,\hat{k}+2}(p_0^*)|$, contradicting that $(\hat{k}, \hat{k}+1) = \arg\min_e\{|P_e^*(\mathcal{E})|\}$. Therefore, $p_0^* \leq p_0(\hat{k}+1)$ since $P_{\hat{k}+1,\hat{k}+2}(p_0)$ is an increasing function of p_0 by Lemma 4.6. □

Considering Lemma 4.7 and 4.9, we have

$$p_0(k) \le p_0^* \quad k \le \hat{k}$$
$$p_0(k) > p_0^* \quad k > \hat{k}.$$

Since $F(p_0, f(p_0))$ is a convex function with minimizer p_0^*, we have for $k_1 \le k_2 \le \hat{k}$

$$F(p_0(k_1), f(p_0(k_1))) \ge F(p_0(k_2), f(p_0(k_2))) \ge F(p_0^*, f(p_0^*)).$$

For $k_1 \ge k_2 \ge \hat{k}$,

$$F(p_0(k_1), f(p_0(k_1))) \le F(p_0(k_2), f(p_0(k_2))) \le F(p_0^*, f(p_0^*)),$$

which indicates the OFR problem (4.12) can be reduced to

$$\min_{0\le k\le n} F(p_0(k), f(p_0(k))) = \min_{k=\hat{k},\hat{k}+1} F(p_0(k), f(p_0(k))),$$

which is solved in line 6 of Algorithm 1. Hence Algorithm 1 solves the optimal solution to OFR.

4.C Proof of Lemma 4.6

For a line $(k, k+1)$ between two buses k and $k+1$ with fixed voltage magnitude, $(S_{k,k+1}, \ell_{k,k+1})$ are governed by (3.2b)-(3.2c) and $Q_{k,k+1}, \ell_{k,k+1}$ can be united solved given a $P_{k,k+1}$ if A3 holds. Denote

$$\phi(P_{k,k+1}) := -P_{k,k+1} = P_{k,k+1} - \ell_{k,k+1} r_{k,k+1}$$

and we have the following result.

Lemma 4.10. *Suppose A2 and A3 hold, $\phi(P_{k,k+1})$ is a concave increasing function of $P_{k,k+1}$ for $(k, k+1) \in \mathcal{E}$.*

Proof. By (3.2c), we have $\ell_{k,k+1} = (P_{k,k+1}^2 + Q_{k,k+1}^2)/v_i$ and substitute it in $\phi(P_{k,k+1})$, we have

$$\phi(P_{k,k+1}) = P_{k,k+1} - \frac{r_{k,k+1}}{v_i}\left(P_{k,k+1}^2 + Q_{k,k+1}^2\right).$$

The relation between $P_{k,k+1}$ and $Q_{k,k+1}$ is governed by (3.2b). Let $\theta_{k,k+1} := \theta_i - \theta_j$ and then $P_{k,k+1}$

and $Q_{k,k+1}$ can be written as

$$\begin{aligned} P_{k,k+1} &= \frac{v_i r_{k,k+1}}{|z_{k,k+1}|^2} + \sqrt{\frac{v_i v_j}{|z_{k,k+1}|^2}} \sin(\theta_{k,k+1} - \beta_{k,k+1}) \\ Q_{k,k+1} &= \frac{v_i x_{k,k+1}}{|z_{k,k+1}|^2} - \sqrt{\frac{v_i v_j}{|z_{k,k+1}|^2}} \cos(\theta_{k,k+1} - \beta_{k,k+1}), \end{aligned}$$

where $\beta_{k,k+1} := \arctan r_{k,k+1}/x_{k,k+1}$. Substitute them into $\phi(P_{k,k+1})$, we obtain

$$\phi(P_{k,k+1}) = -\frac{v_j r_{k,k+1}}{|z_{k,k+1}|^2} + \sqrt{\frac{v_i v_j}{|z_{k,k+1}|^2}} \sin(\theta_{k,k+1} + \beta_{k,k+1}).$$

Take derivative of $\phi(P_{k,k+1})$ with respect to $P_{k,k+1}$, we have

$$\frac{d\phi(P_{k,k+1})}{dP_{k,k+1}} = \frac{\cos(\theta_{k,k+1} + \beta_{k,k+1})}{\cos(\theta_{k,k+1} - \beta_{k,k+1})}.$$

which is always positive by assumption A3 that $|\theta_{k,k+1}| < \arctan x_{k,k+1}/r_{k,k+1}$. Furthermore,

$$\frac{d^2\phi(P_{k,k+1})}{dP_{k,k+1}^2} = -\sqrt{\frac{|z_{k,k+1}|^2}{v_i v_j}} \frac{\sin 2\beta_{k,k+1}}{\cos^3(\theta_{k,k+1} - \beta_{k,k+1})},$$

which is always negative by assumption A3 that $|\theta_{k,k+1}| < \arctan x_{k,k+1}/r_{k,k+1}$. Thus, $\phi(P_{k,k+1})$ is a concave increasing function of $P_{k,k+1}$.

□

Lemma 4.10 means that if the one end of the line increases its real power injection on the line, the other end should receive more real power under assumption A2 and A3. We now show that $P_{k,k+1}(p_0)$ is a nondecreasing function of p_0 for all $(k, k+1) \in \mathcal{E}$.

Suppose $P_{k,k+1}(p_0)$ is not a nondecreasing function of p_0 at p_0^* for a line $(k, k+1) \in \mathcal{E}$, then either C1 or C2 below will hold for arbitrary small $\epsilon > 0$,

C1: $\exists p_0 \in (p_0, p_0^* + \epsilon)$ such that $P_{k,k+1}(p_0) < P_{k,k+1}(p_0^*)$.

C2: $\exists p_0 \in (p_0 - \epsilon, p_0^*)$ such that $P_{k,k+1}(p_0) > P_{k,k+1}(p_0^*)$.

We will show by contradiction that $(p_0^*, f(p_0^*)) \notin \mathcal{O}(\mathbb{P})$ in this case, which violates Lemma 4.2. Assume without loss of generality that $P_{i,i+1}(p_0)$ is a nondecreasing function of p_0 for $0 \le i < k$.

Case I: $q_k(p_0^*) > \underline{q}_k$. Suppose C1 holds, then there exists a monotone decreasing sequence $p_0^{(m)} \downarrow p_0^*$ such that $\{P_{k,k+1}(p_0^{(m)}), m \in \mathbb{N}\}$ is a monotone increasing sequence that converges to $P_{k,k+1}(p_0^*)$

because $x(p_0)$ is continuous over p_0. By power balance equation (3.2b) at bus k, for any m, we have

$$\begin{aligned} p_k(p_0^{(m)};\mathcal{G}) &= P_{k,k+1}(p_0^{(m)}) - \phi(P_{k-1,k}(p_0^{(m)})) \\ &< P_{k,k+1}(p_0^{(m+1)}) - \phi(P_{k-1,k}(p_0^{(m+1)})) \\ &= p_k(p_0^{(m+1)}). \end{aligned}$$

Thus $\{p_k(p_0^{(m)}), n \in \mathbb{N}\}$ is a monotone increasing sequence that converges to $p_k(p_0^*)$. We now construct a point $\tilde{x} = (\tilde{P}, \tilde{Q}, \tilde{p}, \tilde{q})$ as follows. First, pick up $(\tilde{P}_{k,k+1}, \tilde{Q}_{k,k+1}, \tilde{p}_k, \tilde{q}_k)$ such that $\tilde{p}_k \in \{p_k(p_0^{(m)};\mathcal{G}), m \in \mathbb{N}\}$, $\tilde{q}_k \in (\underline{q}_k, q_k(p_0^*))$ and they satisfy the following equations:

$$\tilde{P}_{k,k+1} = P_{k,k+1}(p_0^*) - p_k(p_0^*) + \tilde{p}_k \tag{4.13a}$$

$$\tilde{Q}_{k,k+1} = Q_{k,k+1}(p_0^*) - q_k(p_0^*) + \tilde{q}_k \tag{4.13b}$$

$$v_{k+1} = v_k - 2(r_{k,k+1}\tilde{P}_{k,k+1} + x_{k,k+1}\tilde{Q}_{k,k+1}) + \frac{\tilde{P}_{k,k+1}^2 + \tilde{Q}_{k,k+1}^2}{v_k}|z_{k,k+1}|^2. \tag{4.13c}$$

The existence of $(\tilde{P}_{k,k+1}, \tilde{Q}_{k,k+1}, \tilde{p}_k, \tilde{q}_k)$ is guaranteed by the following two facts:

- $p_k(p_0^{(m)})$ is a monotone increasing sequence that converges to $p_k(p_0^*)$.
- $\tilde{q}_k$ is a continuous decreasing function of $\tilde{p}_k$ if they satisfy (4.13).

Since $\tilde{P}_{k,k+1} \in [P_{k,k+1}(p_0^{(1)}), P_{k,k+1}(p_0^*)]$ and $x(p_0)$ are continuous over p_0, then there exists a $p_0' \in [p_0^*, p_0^{(1)}]$ such that $S_{k,k+1}(p_0') = \tilde{S}_{k,k+1}$.

Next, we will construct the feasible physical variable for $i \neq k$. For $0 \leq i < k$, let $\tilde{s}_i = s_i(p_0^*)$ and $\tilde{S}_{i,i+1} = S_{i,i+1}(p_0^*)$. For $k < i \leq n$, let $\tilde{s}_i = s_i(p_0')$ and $\tilde{S}_{i,i+1} = S_{i,i+1}(p_0')$. Clearly that $\tilde{x} \in \mathbb{X}(\mathcal{E})$ with $(p_0^*, f(p_0'))$ as the real power injection at substation 0 and $0'$. However, $f(p_0') < f(p_0^*)$, which contradicts $(p_0^*, f(p_0^*)) \in \mathcal{O}(\mathbb{P})$.

Case II: $q_k(p_0^*) < \overline{q}_k$. Similar approach can be used to show C2 does not hold by contradiction.

So far, we have shown that $P_{k,k+1}(p_0)$ is non-decreasing either on its left or right neighborhood. Thus $P_{k,k+1}(p_0)$ is non-decreasing of p_0 if $\underline{q}_k < \overline{q}_k$ because $P(p_0)$ is a continuous function of p_0. The case where $\underline{q}_k = \overline{q}_k$ can be covered by taking limitation of the case of $\underline{q}_k < \overline{q}_k$.

4.D Proof of Theorem 4.4

Lemma 4.11. *Suppose A2+ and A3-A6 hold. Given a solution $x(p_0)$ to OPF-$\mathcal{E}$s, $P_{k,k+1}(p_0)$ is a concave increasing function of p_0 for all $(k, k+1) \in \mathcal{E}$.*

Proof. It is shown that $P_{k,k+1}(p_0)$ is a nondecreasing function of p_0 in Lemma 4.6. We now show it is also a concave function of p_0. Let $\mathcal{G}_1 = (\mathcal{N}_1, \mathcal{E}_1)$, where $\mathcal{N}_1 = \{i \mid 0 \leq i \leq k+1\}$ and

$\mathcal{E}_1 = \{(i, i+1) \mid 1 \le i \le k\}$. All the physical constraints are the same as $\mathcal{G}$ except the bus injection power s_{k+1} at node $k+1$, which is relaxed to be a free variable. Mathematically, it means

$$\begin{aligned}
\overline{\ell}_{i,i+1}(\mathcal{G}_1) &= \overline{\ell}_{i,i+1}(\mathcal{G}) \quad i \le k \\
\overline{s}_i(\mathcal{G}_1) &= \overline{s}_i(\mathcal{G}) \quad i \le k \qquad \text{and} \quad \overline{s}_{k+1}(\mathcal{G}_1) = \infty \\
\underline{s}_i(\mathcal{G}_1) &= \underline{s}_i(\mathcal{G}) \quad i \le k \qquad \text{and} \quad \underline{s}_{k+1}(\mathcal{G}_1) = -\infty.
\end{aligned}$$

Consider the following OPF problem:

$$\text{OPF-}\mathcal{G}s1: \qquad \min_{x \in \mathbb{X}(\mathcal{G}_1)} p_{k+1} \quad \text{s.t. } p_0 \text{ is a fixed.}$$

Let $p_{k+1}(p_0; \mathcal{G}_1)$ be the optimal value for OPF-$\mathcal{G}s1$ and $P_{k,k+1}(p_0; \mathcal{G}_1)$ be the real branch power flow across line $(k, k+1)$, respectively.

Next, we will show $P_{k,k+1}(p_0^*) = P_{k,k+1}(p_0^*; \mathcal{G}_1)$. Clearly that $P_{k,k+1}(p_0^*) \le P_{k,k+1}(p_0^*; \mathcal{G}_1)$. Otherwise, by Lemma 4.10, we have

$$-p_{k+1}(p_0^*; \mathcal{G}_1) = \phi(P_{k,k+1}(p_0^*; \mathcal{G}_1)) < \phi(P_{k,k+1}(p_0^*)),$$

which contradicts that $p_{k+1}(p_0^*; \mathcal{G}_1)$ is optimal for OPF-$\mathcal{G}s1$. Thus, it suffices to show $P_{k,k+1}(p_0^*) < P_{k,k+1}(p_0^*; \mathcal{G}_1)$ does not hold. Suppose $P_{k,k+1}(p_0^*) < P_{k,k+1}(p_0^*; \mathcal{G}_1)$ holds. By Lemma 4.6, $P_{k,k+1}(p_0)$ is a nondecreasing function of p_0, and thus there exists a $\hat{p}_0 > p_0^*$ such that

$$P_{k,k+1}(\hat{p}_0) \in [P_{k,k+1}(p_0^*), P_{k,k+1}(p_0^*; \mathcal{G}_1)].$$

Recall that $\mathbb{X}_r(\mathcal{E})$, which is the set of feasible solutions after the SOCP relaxation, is convex and is connected, there exists a $x \in \mathbb{X}_r(\mathcal{E})$ with $P_{k,k+1} = P_{k,k+1}(\hat{p}_0)$ but $p_0 = p_0^*$. It means $(p_0^*, f(\hat{p}_0))$ is also feasible for OPF-$\mathcal{E}$, which contradicts that $(\hat{p}_0, f(\hat{p}_0)) \in \mathcal{O}(\mathbb{P})$.

Now we have $P_{k,k+1}(p_0^*) = P_{k,k+1}(p_0^*; \mathcal{G}_1)$. Since the convex relaxation is exact, $p_{k+1}(p_0; \mathcal{G}_1)$ is a convex decreasing function of p_0 by Lemma 4.2. In addition,

$$\phi(P_{k,k+1}(p_0)) = \phi(P_{k,k+1}(p_0; \mathcal{G}_1)) = -p_{k+1}(p_0, \mathcal{G}_1),$$

where $\phi(\cdot)$ is a continuous increasing function, and thus is invertible. Thus, we have $P_{k,k+1}(p_0) = \phi^{-1}(-p_{k+1}(p_0, \mathcal{G}_1))$, which means $P_{k,k+1}(p_0)$ is a concave function of p_0. □

Let $p_0^f(k)$ be the solution to $P_{k,k+1}(p_0; \mathcal{G}) = 0$ and $p_0^b(k)$ be the solution to $P_{k+1,k}(p_0; \mathcal{G}) = 0$. The uniqueness of $p_0^f(k)$ and $p_0^b(k)$ can be shown in a similar manner as the uniqueness of $p_0(k)$ in the proof of Theorem 4.3. When the voltage magnitude of each bus is the same, $p_0^f(k) = p_0^b(k)$ and

they degenerate to $p_0(k)$.

Lemma 4.12. *Suppose A1,A2+,A3,A4 hold. For any* $(k, k+1) \in \mathcal{E}$, $p_0^k = p_0^f(k)$ *and* $p_{0'}^{k+1} = f(p_0^b(k))$.

Proof. It suffices to show $p_0^f(k)$ is optimal for $\mathcal{G}_0^k$ due to symmetry. First, we show $p_0^f(k)$ is feasible for $\mathcal{G}_0^k$. Given a solution $x(p_0^f(k))$ to OPF-$\mathcal{E}s$, let $\tilde{x} := (\tilde{P}, \tilde{Q}, \tilde{p}, \tilde{q})$, where

$$\begin{aligned}
\tilde{S}_{i,i+1} &= S_{i,i+1}(p_0^f(k)) \quad i < k \\
\tilde{s}_i &= s_i(p_0^f(k)) \quad i < k \\
\tilde{p}_k &= p_k(p_0^f(k)) \\
\tilde{q}_k &= q_k(p_0^f(k)) - Q_{k,k+1}(p_0^f(k)).
\end{aligned}$$

Thus, we have $\tilde{x} \in \mathbb{X}(\mathcal{G}_0^k)$, which means $p_0^f(k)$ is feasible for $\mathcal{G}_0^k$. Next, we will show $p_0^f(k)$ is the minimal power injection for $\mathcal{G}_0^k$. Suppose $\hat{p}_0 < p_0^f(k)$ is feasible for $\mathcal{G}_0^k$, then we can construct a feasible solution $\tilde{x} \in \mathbb{X}(\mathcal{E})$ and the real power injection at node 0 and $0'$ are $\hat{p}_0$ and $f(p_0^f(k))$, respectively. Therefore it contradicts that $(p_0^f(k), f(p_0^f(k))) \in \mathcal{O}(\mathbb{P})$. The construction process is as follows:

$$\tilde{S}_{i,i+1} = \begin{cases} S_{i,i+1}(\hat{p}_0) & i < k \\ S_{i,i+1}(p_0^f(k)) & i \geq k, \end{cases}$$

$$\tilde{s}_i = \begin{cases} s_i(\hat{p}_0) & i < k \\ p_k(\hat{p}_0) + \mathbf{i}(q_k(\hat{p}_0) + Q_{k,k+1}(p_0^f(k))) & i = k \\ s_i(p_0^f(k)) & i > k \end{cases}.$$

It can be verified that $\tilde{x} \in \mathbb{X}(\mathcal{E})$. Therefore, we show that $p_0^f(k)$ is the minimal power injection for $\mathcal{G}_0^k$. □

Lemma 4.13. *Suppose A1,A2+,A3,A4 hold. Then we have*

$$\frac{p_0^b(k) - p_0^f(k)}{p_0^f(k+1) - p_0^b(k)} \leq \frac{1}{R_k}.$$

Proof. By Lemma 4.11, $P_{k,k+1}(p_0)$ is a concave increasing function with respect to p_0. Therefore $-P_{k+1,k}(p_0) = \phi(P_{k,k+1}(p_0))$ is also a concave increasing function of p_0. Recall that $-P_{k+1,k}(p_0^f(k)) =$

$-L_k$, $-P_{k+1,k}(p_0^b(k)) = 0$ and $-P_{k+1,k}(p_0^f(k+1)) = -p_{k+1} \geq -\overline{p}_{k+1}$, and we have

$$\frac{0-(-L_k)}{p_0^b(k)-p_0^f(k)} \geq \frac{-\overline{p}_{k+1}-0}{p_0^f(k+1)-p_0^b(k)}$$

by definition of a concave function. Rearrange the above inequality, and we obtain

$$\frac{p_0^b(k)-p_0^f(k)}{p_0^f(k+1)-p_0^b(k)} \leq \frac{0-(-L_k)}{-\overline{p}_{k+1}-0} := \frac{1}{R_k}.$$

□

Lemma 4.14. *Let $g(x)$ be a strictly convex decreasing function supported on $[a,b]$ and $\kappa_g := \inf_{x\in(a,b)} g''_{++}(p_0)$. Define $G(x) := c_1 g(x) + c_2 x$ $(c_1, c_2 > 0)$, which is also strictly convex with a unique minimizer x^* on $[a,b]$. Let $a \leq y_1 \leq \cdots \leq y_{2n-1} \leq y_{2n} \leq b$ be a partition on $[a,b]$ such that*

$$\frac{y_{2i}-y_{2i-1}}{y_{2i+1}-y_{2i}} \leq \frac{1}{R} \qquad (1 \leq i \leq n-1) \tag{4.14}$$

for some $R > 0$. Then there exists a $0 \leq k \leq 2n$ such that $y_k \leq x^* \leq y_{k+1}$, where $y_0 = a$ and $y_{2n+1} = b$. Let $G_i := c_1 g(y_{2i}) + c_2 y_{2i-1}$ for $1 \leq i \leq n$ and $G^* := \min_{1\leq i\leq n}\{G_i\}$. Define

$$G_A := \begin{cases} G_1 & \text{if } k = 0 \\ G_n & \text{if } k = 2n \\ G_{(k-1)/2} & \text{if } k \text{ is odd} \\ \min\{G_{k/2}, G_{k/2+1}\} & \text{if } k \neq 0, 2n \text{ and is even} \end{cases}.$$

Then

$$G^* \leq G_A \leq G^* + \max\left\{\frac{c_1^2}{c_2}, \frac{c_2^2}{c_1}\right\}\frac{2}{R^2\kappa_g}.$$

Proof. Without loss of generality, assume $c_1 \leq c_2$ and let $\lambda := \frac{c_2}{c_1}$. The unique minimizer x^* of $G(x) := c_1 g(x) + c_2 x$ is

$$\begin{aligned} x^* &= \arg\sup_{x\in[a,b]}\{x \mid G'_+(x) \leq 0\} \\ &= \arg\sup_{x\in[a,b]}\{x \mid g'_+(x) \leq -\lambda\}. \end{aligned}$$

In addition, let

$$\begin{aligned} x_l &:= \arg \sup_{x\in[a,b]} \left\{ x \mid g'_+(x) \le -\lambda(1+\frac{1}{R}) \right\} \\ x_r &:= \arg \inf_{x\in[a,b]} \left\{ x \mid g'_+(x) \ge -\lambda(1-\frac{1}{R}) \right\}. \end{aligned}$$

Then we have $x_l \le x^* \le x_r$ because $g(x)$ is strictly convex. Let

$$\begin{aligned} t_1 &:= \max\{i \mid y_{2i-1} \le x^*\} \\ t_2 &:= \min\{i \mid y_{2i} \ge 2x_l - x^*\} \\ t_3 &:= \min\{i \mid y_{2i} \ge x^*\} \\ t_4 &:= \max\{i \mid y_{2i-1} \le 2x_r - x^*\}. \end{aligned}$$

Next, we will prove the result for different k as sequel.

Case I: $k = 2n$. In this case, we have $d_a = G_n$, $t_1 = n$. We need to further divide it into two categories.

(1.a) [$y_{2t_1-1} \le x_l$ or $y_{2t_1-1} \in [x_l, x^*]$ and $t_1 = t_2$].

For any $i < t_1$,

$$\begin{aligned} \frac{G_i - G_{i+1}}{c_0} &= \lambda g(y_{2i}) + y_{2i-1} - g(y_{2i+2}) - \lambda y_{2i+1} \\ &\ge \lambda g(y_{2i}) + y_{2i-1} - g(y_{2i+1}) - \lambda y_{2i+1} \\ &= \lambda(y_{2i-1} - y_{2i}) + G(y_{2i}) - G(y_{2i+1}) \\ &\ge \frac{\lambda}{R}(y_{2i} - y_{2i+1}) + G(y_{2i}) - G(y_{2i+1}) \\ &\ge 0. \end{aligned} \tag{4.15}$$

The first inequality follows from $g(x)$ is an increasing function and the second inequality follows from the assumption (4.14). For the last inequality, if $y_{2t_1-1} \le x_l$, we have $G'_+(y_{2i+1}) < -\lambda/R$ for all $i < n$ and the inequality holds according to mean value theorem. If $y_{2t_1-1} \in [x_l, x^*]$ and $t_1 = t_2 = n$, the inequality holds for $i < n-1$ due to similar reason above. When $i = n-1$,

$$G(y_{2i}) - G(y_{2i+1}) \ge \quad G'_+(x_l)(y_{2i} - y_{2i+1}) \ge -\tfrac{\lambda}{R}(y_{2i} - y_{2i+1})$$

because $y_{2n-2} \le 2x_l - x^*$ by definition of t_2 and $G(x)$ is convex. (4.15) means the sequence $\{G_i\}$ is of descending order and $G^* = G_n = G_A$, and thus $G^* - G_A = 0$.

(1.b) [$y_{2t_1-1} \in [x_l, x^*]$ and $t_2 < t_1$].

In this case, $y_{2i+1} - y_{2i} \leq 2(x^* - x_l)$ for $t_2 \leq i < t_1$. Denote

$$\delta y_i := y_{2i} - y_{2i-1} \leq \frac{y_{2i+1} - y_{2i}}{R} \leq \frac{2(x^* - x_l)}{R} \tag{4.16}$$

for $t_2 \leq i < t_1$. Note that the curvature of $g(x)$ is bounded below by κ_g, and that $x^* - x_l \leq \lambda/(R\kappa_g)$. Substitute it into (4.16), and we have for $t_2 \leq i < t_1$

$$\delta y_i \leq \frac{2\lambda}{R^2\kappa_g}. \tag{4.17}$$

Then for $t_2 \leq i < t_1$,

$$\begin{aligned} G_i - G_{t_1} &= c_1 g(y_{2i}) + c_2 y_{2i-1} - G_{t_1} \\ &\geq c_1 g(y_{2i}) + c_2 y_{2i-1} - G(y_{2t_1-1}) \\ &= -c_2 \delta y_i + G(y_{2i}) - G(y_{2t_1-1}) \\ &\geq -\frac{2c_2^2}{c_1 R^2 \kappa_g}. \end{aligned}$$

Clearly the first inequality holds. The second inequality follows from (4.17) and $G(x)$ is monotone decreasing for $x \leq x^*$.

For $i \leq t_2$, $G_i > G_{t_1}$ can be shown in a similar manner as (1.a). Thus, we have $G_i - G_n \geq -\frac{2c_2^2}{c_1 R^2 \kappa_g}$ for any $i \leq n$, which indicates $G_A - G^* \leq -\frac{2c_2^2}{c_1 R^2 \kappa_g}$.

Case II: $k = 0$. In this case, $G_A = G_1$ and the bound can be established in a similar manner as Case I.

Case III: k is odd. In this case, $G_A = G_{(k-1)/2}$, $t_1 = t_3 = (k-1)/2$. A similar approach can be applied as Case I and Case II to show

$$G_i \geq \begin{cases} G(y_{2t_1-1}) - \frac{2c_2^2}{c_1 R^2 \kappa_g} & \text{if } i \leq t_1 \\ G(y_{2t_3}) - \frac{2c_2^2}{c_1 R^2 \kappa_g} & \text{if } i \geq t_3. \end{cases}$$

And $G_A = G_{(k-1)/2} \leq \max\{G(y_{2t_1-1}), G(y_{2t_3})\} \leq G_i + \frac{2c_2^2}{c_1 R^2 \kappa_g}$ for $1 \leq i \leq n$.

Case IV: $k \neq 0$ and is even. In this case, $G_A = \min\{G_{k/2}, G_{k/2+1}\}$ and $t_1 = k/2$, $t_3 = k/2 + 1$. A similar approach can be applied as Case I and Case II to show

$$G_i \geq \begin{cases} G_{t_1} - \frac{2c_2^2}{c_1 R^2 \kappa_g} & \text{if } i \leq t_1 \\ G_{t_3} - \frac{2c_2^2}{c_1 R^2 \kappa_g} & \text{if } i \geq t_3 \end{cases}$$

and we arrive at our conclusion. □

Consider the sequence $p_0^f(0) \leq p_0^b(0) \leq \ldots \leq p_0^f(n) \leq p_0^b(n)$ as the partition on I_{p_0} and $f(p_0)$ as the function $g(x)$ in Lemma 4.14, and we can prove Theorem 4.4.

Chapter 5

Alternating Direction Method of Multipliers (ADMM)

Alternating direction method of multipliers (ADMM) blends the decomposability of dual decomposition with the superior convergence properties of the method of multipliers [11]. It is well suited to distributed convex optimization, and in particular to large-scale problems arising in statistics and machine learning. In this chapter, we first review some background knowledge of ADMM. Then we show how to develop efficient centralized and distributed algorithms for a broad class of optimization problem based on ADMM. The proposed approach is applied in the design of efficient algorithms that solve the OPF problem on balanced networks in Chapter 6 and unbalanced networks in Chapter 7.

5.1 Background on ADMM

Consider the following optimization problem[1]:

$$\begin{aligned} \min \quad & f(x)+g(y) \\ \text{over} \quad & x,y \in \mathbb{C}^{m\times n} \\ \text{s.t.} \quad & x \in \mathcal{K}_x, \quad y \in \mathcal{K}_y \\ & x = y, \end{aligned} \tag{5.1}$$

where $f(x), g(y)$ are convex functions and $\mathcal{K}_x, \mathcal{K}_y$ are convex sets. Let λ denote the Lagrange multiplier for the constraint $x = y$. Then the augmented Lagrangian is defined as

$$L_\rho(x,y,\lambda) := f(x)+g(y)+\langle \lambda, x-y\rangle + \frac{\rho}{2}\|x-y\|_2^2, \tag{5.2}$$

[1]This is a simplified version of the general ADMM introduced in [11]. The general problem allows a linear constraint $Ax+By=c$ between x and y instead of $x=y$. The z variable used in [11] is replaced by y since z represents impedance in power system.

where $\rho \geq 0$ is a constant. When $\rho = 0$, the augmented Lagrangian degenerates to the standard Lagrangian. At each iteration k, ADMM consists of the iterations:

$$x^{k+1} \in \arg\min_{x \in \mathcal{K}_x} L_\rho(x, y^k, \lambda^k) \tag{5.3a}$$

$$y^{k+1} \in \arg\min_{z \in \mathcal{K}_y} L_\rho(x^{k+1}, y, \lambda^k) \tag{5.3b}$$

$$\lambda^{k+1} = \lambda^k + \rho(x^{k+1} - y^{k+1}). \tag{5.3c}$$

Specifically, at each iteration, ADMM first updates x based on (5.3a), then updates y based on (5.3b), and after that updates the multiplier λ based on (5.3c). Compared to dual decomposition, ADMM is guaranteed to converge to an optimal solution under less restrictive conditions. Let

$$r^k := \|x^k - y^k\|_2 \tag{5.4a}$$

$$s^k := \rho\|y^k - y^{k-1}\|_2, \tag{5.4b}$$

which can be viewed as the residuals for primal and dual feasibility, respectively. They converge to 0 at optimality and are usually used as metrics of convergence. Assume:

- A1: f and g are closed proper and convex.
- A2: The unaugmented Lagrangian L_0 has a saddle point.

The correctness of ADMM is guaranteed by the following result; see [11, Chapter 3].

Proposition 5.1 ([11])**.** *Suppose A1 and A2 hold. Let p^* be the optimal objective value. Then*

$$\lim_{k\to\infty} r^k = 0, \quad \lim_{k\to\infty} s^k = 0,$$

and

$$\lim_{k\to\infty} f(x^k) + g(y^k) = p^*.$$

To better leverage ADMM, we generalize the above standard ADMM as below. Instead of using the quadratic penalty term $\frac{\rho}{2}\|x - y\|_2^2$ in (5.2), we will use a more general quadratic penalty term, $\frac{1}{2}(x-y)^H\Lambda(x-y)$, where Λ is a positive diagonal matrix. Then the augmented Lagrangian becomes

$$L_\Lambda(x, y, \lambda) := f(x) + g(y) + \langle\lambda, x - y\rangle + \frac{1}{2}(x - y)^H\Lambda(x - y). \tag{5.5}$$

The convergence result in Proposition 5.1 carries over directly to this general case.

ADMM has broad applicability in different areas, e.g. matrix factorization [83], image recovery [1]. It is particularly useful when the subproblems can be solved efficiently [34], for example when

the subproblems in the x-update (5.3a) and the y-update (5.3b) admit closed form expressions. In the following analysis, we show how to develop efficient algorithms for a broad class of optimization problem through ADMM.

5.2 Algorithm Design using ADMM

There are various generic optimization solvers for convex programs, e.g. Gurobi [69] and MOSEK [68], etc.. Most of them employ general purpose algorithms, such as interior point method [64], and do not leverage the special problem structures, which are conducive to improving the computation efficiency. In this section, we will apply ADMM to develop both centralized and distributed algorithms that can efficiently solve a broad class of problem defined in (5.6).

Consider the following optimization problem:

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i(x_i) && \text{(5.6a)}\\
\text{over} \quad & \{x_i \mid i \in \mathcal{N}\} && \text{(5.6b)}\\
\text{s.t.} \quad & \sum_{j\in N_i} A_{ij}x_j = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(5.6c)}\\
& x_i \in \mathcal{K}_i \quad \text{for} \quad i \in \mathcal{N}, && \text{(5.6d)}
\end{aligned}$$

where for each $i \in \mathcal{N}$, x_i is a complex vector, $f_i(x_i)$ is a convex function, $\mathcal{K}_i$ is a convex set, and A_{ij} $(j \in N_i, i \in \mathcal{N})$ are full row rank matrices with appropriate dimensions. A broad class of graphical optimization problems (including ROPF) can be formulated as (5.6). Each node $i \in \mathcal{N}$ is associated with some local variables stacked as x_i, which belongs to a local feasible set $\mathcal{K}_i$ and has a cost objective function $f_i(x_i)$. Variables in node i are coupled with variables from their neighbor nodes in N_i through linear constraints (5.6c). The goal is to minimize the total cost across all nodes.

5.2.1 Centralized Algorithm

In this subsection, we develop a computationally efficient centralized algorithm that solves the problem in (5.6). We introduce a set of variables y_i for $i \in \mathcal{N}$, which represents a "duplicate" of the

original variable x_i, and then (5.6) can be reformulated as

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i(x_i) && \text{(5.7a)}\\
\text{over} \quad & x = \{x_i \mid i \in \mathcal{N}\} \\
& y = \{y_i \mid i \in \mathcal{N}\} \\
\text{s.t.} \quad & \sum_{j\in N_i} A_{ij} y_j = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(5.7b)}\\
& x_i \in \mathcal{K}_i \quad \text{for} \quad i \in \mathcal{N} && \text{(5.7c)}\\
& x_i = y_i \quad \text{for} \quad i \in \mathcal{N}, && \text{(5.7d)}
\end{aligned}$$

where x and y represent the two groups of variables in standard ADMM. The consensus constraint (5.7d) ensures that each duplicate y_i equals the original variable x_i, and thus the solution x^* to (5.7) is also optimal to the original problem (5.6).

The problem (5.7) falls into the standard form of ADMM, where (5.7b) and (5.7c) correspond to $\mathcal{K}_y$ and $\mathcal{K}_x$, respectively. The key feature of (5.7) that makes it potentially easier to solve than (5.6) is that x_i are decoupled into local costs $f_i(x_i)$ and local constraints (5.7c), and that, taking the entire vector $y = \{y_i, i \in N\}$ as a single local variable, (5.7b) is a local constraint. Moreover, the cost function does not contain y and, as we will see below, this allows a closed form solution for the subproblem that updated the y variable. Then the only coupling constraint is the consensus constraint (5.7d), one for each i. This is not only simpler and more local than the constraint (5.6c), more importantly, it decouples variables x_i and x_j. This allows the x-update step to be completely local to each i (see (5.12) below).

Loosely speaking, the optimization subproblems ((5.3a) and (5.3b)) are potentially easy to solve because the constraints (5.7b) and (5.7c) are distributed into different optimization subproblems.

Following the standard ADMM procedure, we will relax the consensus constraint (5.7d), whose Lagrangian multiplier is denoted by λ_i for $i \in \mathcal{N}$. Then the augmented Lagrangian for (5.7) can be written as

$$L_\rho(x, y, \lambda) = \sum_{i\in\mathcal{N}} \left(f_i(x_i) + \langle \lambda_i, x_i - y_i \rangle + \frac{\rho}{2} \|x_i - y_i\|_2^2 \right). \tag{5.8}$$

In the x-update (5.3a) at each iteration k, we solve the following optimization problem to update x^{k+1}:

$$\min_{x\in\mathcal{K}_x} L_\rho(x, y^k, \lambda^k), \tag{5.9}$$

where $\mathcal{K}_x$ is the Cartesian product of each local constraint $\mathcal{K}_i$, i.e.

$$\mathcal{K}_x = \otimes_{i\in\mathcal{N}}\mathcal{K}_i.$$

For the objective, it can be written as a sum of local objective as shown below

$$\begin{aligned} L_\rho(x, y^k, \lambda^k) &= \sum_{i\in\mathcal{N}} \Big(f_i(x_i) + \langle \lambda_i^k, x_i - y_i^k \rangle + \frac{\rho}{2}\|x_i - y_i^k\|_2^2 \Big) \\ &= \sum_{i\in\mathcal{N}} H_i(x_i) - \sum_{i\in\mathcal{N}} \langle \lambda_i^k, y_i^k \rangle, \end{aligned}$$

where the last term $\sum_{i\in\mathcal{N}}\langle \lambda_i^k, y_i^k\rangle$ is independent of the decision variable x and

$$H_i(x_i) := f_i(x_i) + \langle \lambda_i^k, x_i\rangle + \frac{\rho}{2}\|x_i - y_i^k\|_2^2. \tag{5.10}$$

Then the problem (5.9) can be written explicitly as

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} H_i(x_i) \\ \text{over} \quad & x = \{x_i \mid i \in \mathcal{N}\} \\ \text{s.t.} \quad & x_i \in \mathcal{K}_i \quad \text{for} \quad i \in \mathcal{N}, \end{aligned} \tag{5.11}$$

whose objective and constraint are both separable, and hence can be decomposed into $|\mathcal{N}|$ independent problems that can be solved in parallel for each node $i \in \mathcal{N}$. The associated problem for node i is

$$\min_{x_i\in\mathcal{K}_i} \; H_i(x_i). \tag{5.12}$$

Whether (5.12) can be solved efficiently depends on the form of $f_i(x_i)$ in the objective $H_i(x_i)$ and the constraint set $\mathcal{K}_i$. For the ROPF problem on balanced radial distribution networks, we show in Chapter 6 the corresponding (5.12) can be solved with closed form solution.

Next, we investigate the subproblem solved in the y-update (5.3b). At iteration k, the variable y^{k+1} is updated according to (5.3b):

$$\min_{y\in\mathcal{K}_y} L_\rho(x^{k+1}, y, \lambda^k). \tag{5.13}$$

For the objective, it can be written as

$$\begin{aligned} L_\rho(x^{k+1}, y, \lambda^k) &= \sum_{i\in\mathcal{N}} \left(f_i(x_i^{k+1}) + \langle \lambda_i^k, x_i^{k+1} - y_i \rangle + \frac{\rho}{2}\|x_i^{k+1} - y_i\|_2^2 \right) \\ &= \sum_{i\in\mathcal{N}} G_i(y_i) + \sum_{i\in\mathcal{N}} \left(f_i(x_i^{k+1}) + \langle \lambda_i^k, x_i^{k+1} \rangle \right), \end{aligned}$$

where the last term $\sum_{i\in\mathcal{N}} \left(f_i(x_i^{k+1}) + \langle \lambda_i^k, x_i^{k+1} \rangle \right)$ is independent of y_i and

$$G_i(y_i) := -\langle \lambda_i^k, y_i \rangle + \frac{\rho}{2}\|x_i^{k+1} - y_i\|_2^2.$$

The constraint $\mathcal{K}_y$ can be written explicitly as

$$\mathcal{K}_y = \left\{ y \mid \sum_{j\in N_i} A_{ij} y_j = 0 \text{ for } i \in \mathcal{N} \right\}.$$

Then the subproblem solved in the y-update can be explicitly written as

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} G_i(y_i) \\ \text{over} \quad & y = \{y_i \mid i \in \mathcal{N}\} \\ \text{s.t.} \quad & \sum_{j\in N_i} A_{ij} y_j = 0 \quad \text{for} \quad i \in \mathcal{N}. \end{aligned} \tag{5.14}$$

To solve (5.14) in closed form, we can stack the real and imaginary part of the variables $y = \{y_i \mid i \in \mathcal{N}\}$ in a vector with appropriate dimensions and denote it as $\tilde{y}$. Then (5.14) takes the following form:

$$\begin{aligned} \min \quad & \frac{1}{2}\tilde{y}^T M \tilde{y} + c^T \tilde{y} \\ \text{over} \quad & \tilde{y} \\ \text{s.t.} \quad & \tilde{A}\tilde{y} = 0, \end{aligned} \tag{5.15}$$

where M is a positive diagonal matrix, $\tilde{A}$ is a full row rank real matrix since A_{ij} has full row rank, and c is a real vector. M, c, A are derived from (5.14). There exists a closed form solution for (5.15) given by

$$\tilde{y} = \left(M^{-1}\tilde{A}^T(\tilde{A}M^{-1}\tilde{A}^T)^{-1}\tilde{A}M^{-1} - M^{-1} \right) c. \tag{5.16}$$

which can be obtained by the KKT conditions of (5.15).

In summary, the original problem (5.7) is decomposed into two subproblems that can be solved

using ADMM. At each iteration, we solve (5.12) in the x-update and (5.14) in the y-update. Since there exists a closed form solution (5.16) to (5.14), whether (5.7) can be solved efficiently depends on the existence of an efficient solver for (5.12).

In Chapter 6, we show that the corresponding (5.12) admits a closed form solution for the ROPF problem on balanced radial distribution networks. However, for unbalanced radial distribution networks, the corresponding (5.12) does not admit any efficient solution. In the following, we will investigate how to further reformulate (5.7) such that all the subproblems can be solved efficiently for unbalanced networks as well.

For $i \in \mathcal{N}$, let $\mathcal{K}_{ir}$ $(0 \leq r \leq R_i)$ for some integer $R_i \geq 0$ denote a superset of $\mathcal{K}_i$ and

$$\cap_{r=0}^{R_i} \mathcal{K}_{ir} = \mathcal{K}_i.$$

Then (5.6) can be reformulated as

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{N}} f_i(x_{i0}) && \text{(5.17a)}\\
\text{over} \quad & x = \{x_{ir} \mid 0 \leq r \leq R_i, i \in \mathcal{N}\} \\
& y = \{y_i \mid i \in \mathcal{N}\} \\
\text{s.t.} \quad & \sum_{j \in N_i} A_{ij} y_j = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(5.17b)}\\
& x_{ir} \in \mathcal{K}_{ir} \quad \text{for} \quad 0 \leq r \leq R_i \;\; i \in \mathcal{N} && \text{(5.17c)}\\
& x_{ir} = y_i \quad \text{for} \quad 0 \leq r \leq R_i \;\; i \in \mathcal{N}. && \text{(5.17d)}
\end{aligned}$$

In contrast to (5.7), where there is one copy x_i for each node $i \in \mathcal{N}$, there are multiple copies x_{ir} in the x-update for each node $i \in \mathcal{N}$. Likewise, we apply ADMM and relax the consensus constraint (5.17d), whose Lagrangian multiplier is denoted by λ_{ir} for $0 \leq r \leq R_i$ and $i \in \mathcal{N}$. Then the augmented Lagrangian for (5.17) can be written as

$$L_\rho(x, y, \lambda) = \sum_{i \in \mathcal{N}} \left(f_i(x_{i0}) + \sum_{r=0}^{R_i} \left(\langle \lambda_{ir}, x_{ir} - y_i \rangle + \frac{\rho}{2} \|x_{ir} - y_i\|_2^2 \right) \right). \tag{5.18}$$

In the x-update, the optimization subproblem that updates x^{k+1} is

$$\min_{x \in \mathcal{K}_x} L_\rho(x, y^k, \lambda^k). \tag{5.19}$$

For the objective, it can be written as a sum of local objective as shown below:

$$\begin{aligned} L_\rho(x, y^k, \lambda^k) &= \sum_{i\in\mathcal{N}} \left(f_i(x_{i0}) + \sum_{r=0}^{R_i} \left(\langle \lambda_{ir}^k, x_{ir} - y_i^k \rangle + \frac{\rho}{2} \|x_{ir} - y_i^k\|_2^2 \right) \right) \\ &= \sum_{i\in\mathcal{N}} \sum_{r=0}^{R_i} H_{ir}(x_{ir}) - \sum_{i\in\mathcal{N}} \sum_{r=0}^{R_i} \langle \lambda_{ir}^k, y_i^k \rangle, \end{aligned}$$

where

$$H_{ir}(x_{ir}) = f_i(x_{i0}) 1_{\{r=0\}} + \langle \lambda_{ir}^k, x_{ir} \rangle + \frac{\rho}{2} \|x_{ir} - y_i^k\|_2^2.$$

For the constraint set $\mathcal{K}_x$, it can be represented as a Cartesian product of $\sum_{i\in\mathcal{N}}(R_i + 1)$ disjoint sets, i.e.

$$\mathcal{K}_x = \otimes_{i\in\mathcal{N}} \otimes_{r=0}^{R_i} \mathcal{K}_{ir}.$$

Thus, the x-update can be decomposed into $\sum_{i\in\mathcal{N}}(R_i + 1)$ subproblems and there are $R_i + 1$ subproblems associated with node i. The r_{th} $(0 \le r \le R_i)$ problem for node i is

$$\min_{x_{ir}\in\mathcal{K}_{ir}} H_{ir}(x_{ir}). \tag{5.20}$$

(5.20) is structurally simpler than (5.12) due to the simpler constraint. However, whether (5.20) admits an efficient solution relies on both the partition and the form of $\mathcal{K}_i$. In Chapter 6, we show that a single partition ($R_i = 0$) suffices for solving OPF efficiently in balanced networks, i.e. (5.7) can be used directly. In Chapter 7, we show that two partitions ($R_i = 1$) are required in order to have a computationally efficient algorithm for the ROPF problem on unbalanced radial distribution networks.

In the y-update, the optimization problem that updates y^{k+1} is

$$\min_{y\in\mathcal{K}_y} L_\rho(x^{k+1}, y, \lambda^k), \tag{5.21}$$

which has similar structure as (5.13) and can be written explicitly as below.

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} G_i(y_i) \\ \text{over} \quad & y = \{y_i \mid i \in \mathcal{N}\} \\ \text{s.t.} \quad & \sum_{j\in N_i} A_{ij} y_j = 0 \quad \text{for} \quad i \in \mathcal{N}, \end{aligned} \tag{5.22}$$

where

$$G_i(y_i) = \sum_{r=0}^{R_i} \left(-\langle \lambda_{ir}^k, y_i \rangle + \frac{\rho}{2} \| x_{ir}^{k+1} - y_i \|_2^2 \right).$$

(5.22) still takes the form of (5.15) and admits a closed form solution (5.16).

5.2.2 Distributed Algorithm

In section 5.2.1, we show how to develop an efficient solution using ADMM that solves (5.6) in a centralized manner. In this section, we further leverage ADMM to develop a computationally efficient distributed algorithm.

Recall that the original optimization problem (5.6) can be mapped onto a graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$. Then a distributed algorithm satisfies the following two properties:

- Computation is local to each node $i \in \mathcal{N}$ such that all the nodes in $\mathcal{N}$ can solve their own problems in parallel.
- Communication is only required among neighbors on the graph $\mathcal{G}$.

For the centralized algorithms that solve (5.17), the problem in the x-update (5.19) can be decomposed into $|\mathcal{N}|$ subproblems, i.e. the subproblems (5.20) are local to each node i. However, the problems solved in the y-update (5.22) involve variables from all the nodes in $\mathcal{N}$. In this subsection, we will further reformulate (5.17) such that the subproblems in the y-update can also be decomposed into local subproblems, which can be solved in parallel.

Note that the coupling among nodes is due to (5.17b) in (5.17). To decouple the constraint (5.17b), we introduce another set of slack variables y_{ji}, which represents the variables of node j observed at node i for $j \in N_i$, $i \in \mathcal{N}$. Then (5.17) can be further reformulated as

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{N}} f_i(x_{i0}) && \text{(5.23a)} \\
\text{over} \quad & x = \{x_{ir} \mid 0 \le r \le R_i, i \in \mathcal{N}\} \\
& y = \{y_{ji} \mid j \in N_i, i \in \mathcal{N}\} \\
\text{s.t.} \quad & \sum_{j \in N_i} A_{ij} y_{ji} = 0 \quad \text{for} \quad i \in \mathcal{N} && \text{(5.23b)} \\
& x_{ir} \in \mathcal{K}_{ir} \quad \text{for} \quad 0 \le r \le R_i \;\; i \in \mathcal{N} && \text{(5.23c)} \\
& x_{ir} = y_{ii} \quad \text{for} \quad 1 \le r \le R_i \;\; i \in \mathcal{N} && \text{(5.23d)} \\
& x_{i0} = y_{ij} \quad \text{for} \quad j \in N_i \;\; i \in \mathcal{N}. && \text{(5.23e)}
\end{aligned}$$

The constraints that involve both x and y in (5.23) are the two sets of consensus constraints (5.23d) and (5.23e). Therefore, we relax both constraints, whose Lagrangian multipliers are denoted by λ_{ir}

and μ_{ij}, respectively. The augmented Lagrangian then can be written as

$$\begin{aligned} & L_\rho(x, y, \lambda, \mu) \\ = & \sum_{i \in \mathcal{N}} \left(f_i(x_{i0}) + \sum_{r=1}^{R_i} \left(\langle \lambda_{ir}, x_{ir} - y_{ii} \rangle + \frac{\rho}{2} \|x_{ir} - y_{ii}\|_2^2 \right) + \sum_{j \in N_i} \left(\langle \mu_{ij}, x_{i0} - y_{ij} \rangle + \frac{\rho}{2} \|x_{i0} - y_{ij}\|_2^2 \right) \right). \end{aligned} \tag{5.24}$$

Next, we show that both the x-update (5.3a) and y-update (5.3b) can be decomposed into local subproblems that can be solved in parallel by each node i with only neighborhood communications, i.e. the problem (5.23) can be solved in a distributed manner using ADMM.

For each node $i \in \mathcal{N}$, it updates not only its own duplicates x_{ir} and the associated multiplier λ_{ir} for $0 \le r \le R_i$, but also the "observation" of variables y_{ji} from its neighbor N_i and the associated multiplier μ_{ji}. Let $\mathcal{A}_i$ denote the set of local variables for node i and

$$\mathcal{A}_i := \{x_{ir}, \lambda_{ir} \mid 0 \le r \le R_i\} \cup \{y_{ji}, \mu_{ji} \mid j \in N_i\}. \tag{5.25}$$

In the x-update, the optimization subproblem that updates x^{k+1} is

$$\min_{x \in \mathcal{K}_x} L_\rho(x, y^k, \lambda^k), \tag{5.26}$$

which has similar structure as (5.19) and there are $R_i + 1$ subproblems associated with each node i. The r_{th} subproblem is

$$\min_{x_{ir} \in \mathcal{K}_{ir}} H_{ir}(x_{ir}), \tag{5.27}$$

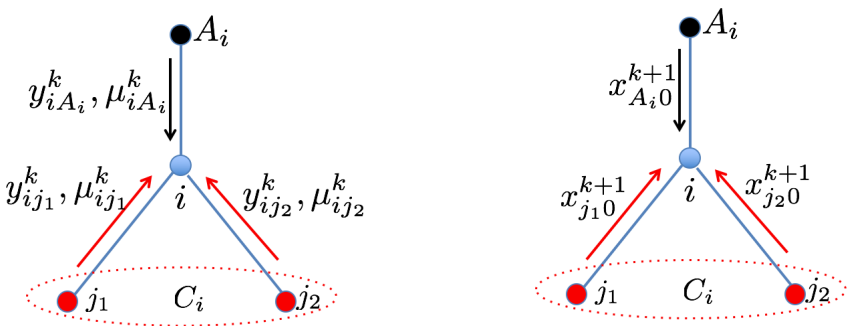
where

$$H_{ir}(x_{ir}) = \begin{cases} f_i(x_{i0}) + \sum_{j \in N_i} \left(\langle \mu_{ij}^k, x_{i0} \rangle + \frac{\rho}{2} \|x_{i0} - y_{ij}^k\|_2^2 \right) & \text{if } r = 0 \\ \langle \lambda_{ir}^k, x_{ir} \rangle + \frac{\rho}{2} \|x_{ir} - y_{ii}^k\|_2^2 & \text{otherwise} \end{cases}. \tag{5.28}$$

In (5.27), the variables $y_{ij}^k, \mu_{ij}^k \in \mathcal{A}_j$ are stored in i's neighbor N_i. Thus, each node i needs to collect (y_{ij}^k, μ_{ij}^k) from all of its neighbors prior to solving (5.27). The message exchanges in the x-update is illustrated in Fig. 5.1a.

In the y-update, we solve the following problem to update y^{k+1}:

$$\min_{y \in \mathcal{K}_y} L_\rho(x^{k+1}, y, \lambda^k), \tag{5.29}$$



(a) Before x update. (b) Before y update.

Figure 5.1: Message passing for a node i.

whose objective can be written as a sum of local objective as shown below.

$$
\begin{aligned}
& L_\rho(x^{k+1}, y, \lambda^{k+1}) \\
&= \sum_{i\in\mathcal{N}} \left(f_i(x^{k+1}_{i0}) + \sum_{r=1}^{R_i} \left(\langle \lambda^k_{ir}, x^{k+1}_{ir} - y_{ii} \rangle + \frac{\rho}{2} \|x^{k+1}_{ir} - y_{ii}\|_2^2 \right) + \sum_{j\in N_i} \left(\langle \mu^k_{ij}, x^{k+1}_{i0} - y_{ij} \rangle + \frac{\rho}{2} \|x^{k+1}_{i0} - y_{ij}\|_2^2 \right) \right) \\
&= \sum_{i\in\mathcal{N}} \left(f_i(x^{k+1}_{i0}) + \sum_{r=1}^{R_i} \left(\langle \lambda^k_{ir}, x^{k+1}_{ir} - y_{ii} \rangle + \frac{\rho}{2} \|x^{k+1}_{ir} - y_{ii}\|_2^2 \right) + \sum_{j\in N_i} \left(\langle \mu^k_{ji}, x^{k+1}_{j0} - y_{ji} \rangle + \frac{\rho}{2} \|x^{k+1}_{j0} - y_{ji}\|_2^2 \right) \right) \\
&= \sum_{i\in\mathcal{N}} G_i(\{y_{ji} \mid j \in N_i\}) + \sum_{i\in\mathcal{N}} \left(f_i(x^{k+1}_{i0}) + \sum_{r=0}^{R_i} \langle \lambda^k_{ir}, x^{k+1}_{ir} \rangle + \sum_{j\in N_i} \langle \mu^k_{ji}, x^{k+1}_{j0} \rangle \right),
\end{aligned}
$$

where

$$
G_i(\{y_{ji} \mid j \in N_i\}) = \sum_{r=1}^{R_i} \left(-\langle \lambda^k_{ir}, y_{ii} \rangle + \frac{\rho}{2} \|x^{k+1}_{ir} - y_{ii}\|_2^2 \right) + \sum_{j\in N_i} \left(-\langle \mu^k_{ji}, y_{ji} \rangle + \frac{\rho}{2} \|x^{k+1}_{j0} - y_{ji}\|_2^2 \right).
$$

For the constraint set $\mathcal{K}_y$, it can also be represented as a Cartesian product of $|\mathcal{N}|$ disjoint sets, i.e.

$$
\mathcal{K}_y = \otimes_{i\in\mathcal{N}} \{y_{ji}, j \in N_i \mid \sum_{j\in N_i} A_{ij} y_{ji} = 0\}.
$$

Therefore, the subproblem (5.29) in the y-update can be decomposed into $|\mathcal{N}|$ subproblems and the subproblem for node i is

$$
\begin{aligned}
\min \quad & G_i(\{y_{ji} \mid j \in N_i\}) \\
\text{over} \quad & \{y_{ji} \mid j \in N_i\} \\
\text{s.t.} \quad & \sum_{j\in N_i} A_{ij} y_{ji} = 0,
\end{aligned}
\tag{5.30}
$$

which still takes the form of (5.15) and admits a closed form solution (5.16). Note that in order to solve (5.30), not all the information is available to the node i, e.g. x_{j0}^{k+1} ($j \in N_i$) resides in node i's neighbor. Thus, before updating $\{y_{ji} \mid j \in N_i\}$, each node i needs to request those data from its neighbor and the communication is shown in Fig. 5.1b.

5.3 Applications

In [11], the authors show that many optimization problems, such as ℓ_1 regularization problem, consensus and sharing, distributed model fitting etc. can be cast into the general ADMM form (5.1). In this section, we show that another two important optimization problems: the optimal power flow problem and second order cone program, can be recast into the form (5.6). By using the proposed technique, closed form expression can be derived for subproblems for both the x-update and y-update. Hence those two problems can be solved more efficiently than using generic iterative optimization solvers.

5.3.1 Optimal Power Flow

The ROPF problem for both balanced (3.9) and unbalanced (3.17) networks takes the form of (5.6). Here, we will use ROPF for balanced networks as an example. Note that (3.9) can be written explicitly as

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i(s_i) && && \text{(5.31a)}\\
\text{over} \quad & v, s, S, \ell && && \text{(5.31b)}\\
\text{s.t.} \quad & v_j - v_i + (z_{ij}S_{ij}^* + S_{ij}z_{ij}^*) - \ell_{ij}|z_{ij}|^2 = 0 && (i,j)\in\mathcal{E} && \text{(5.31c)}\\
& \sum_{k\in C_i}(S_{ki} - \ell_{ki}z_{ki}) + s_i - \sum_{j\in A_i} S_{ij} = 0 && i\in\mathcal{N} && \text{(5.31d)}\\
& |S_{ij}|^2 \le v_i\ell_{ij} && (i,j)\in\mathcal{E} && \text{(5.31e)}\\
& s_i \in \mathcal{I}_i && i\in\mathcal{N} && \text{(5.31f)}\\
& \underline{v}_i \le v_i \le \overline{v}_i && i\in\mathcal{N} && \text{(5.31g)}
\end{aligned}$$

Denote $x_i := (v_i, s_i, \{S_{ij}, \ell_{ij} \mid j \in A_i\})$, then (5.31c) and (5.31d) correspond to the coupling constraints (5.6c). For constraint (5.31e), it can be written equivalently as

$$\begin{aligned}
& \otimes_{(i,j)\in\mathcal{E}}\{v_i, S_{ij}, \ell_{ij} \mid |S_{ij}|^2 \le v_i\ell_{ij}\}\\
= \quad & \otimes_{i\in\mathcal{N}}\{v_i, \{S_{ij}, \ell_{ij} \mid j\in A_i\} \mid |S_{ij}|^2 \le v_i\ell_{ij}, j\in A_i\}.
\end{aligned}$$

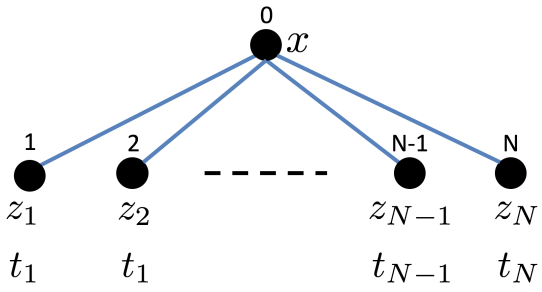


Figure 5.1: Graph representation of SOCP.

Thus, (5.31e) together with (5.31f), (5.31g) can be written in the form of (5.6d). The details are illustrated in Chapter 6.

5.3.2 Second Order Cone Program

Second order cone program (SOCP) is a nonlinear convex problem that minimizes a linear function over the intersection of second order cones. Various problems can be recast into SOCP, e.g. robust linear program [9], robust least square [29], etc. In [58], the general SOCP is written as:

$$\begin{aligned} \min \quad & f^T x \\ \text{over} \quad & x \in \mathbb{R}^n \\ \text{s.t.} \quad & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, N, \end{aligned} \tag{5.32}$$

where $f \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i - 1} \times n$, $b_i \in \mathbb{R}^{n_i - 1}$, $c_i \in \mathbb{R}^n$, and $d_i \in \mathbb{R}$. Next, we will show how to solve (5.32) efficiently by casting (5.32) into (5.6).

Denote $z_i := A_i x + b_i$ and $t_i := c_i^T x + d_i$ for $i = 1, \ldots, N$, then (5.32) can be written equivalently as

$$\min \quad f^T x \tag{5.33a}$$

$$\text{over} \quad x \in \mathbb{R}^n, \{z_i, t_i \mid i = 1, \ldots, N\} \tag{5.33b}$$

$$\text{s.t.} \quad A_i x - z_i + b_i = 0 \quad i = 1, \ldots, N \tag{5.33c}$$

$$c_i^T x - t_i + d_i = 0 \quad i = 1, \ldots, N \tag{5.33d}$$

$$\|z_i\|_2 \leq t_i \quad i = 1, \ldots, N. \tag{5.33e}$$

which takes the form of (5.6). Specifically, (5.33e) corresponds to the local constraint (5.6d), and (5.33c)–(5.33d) correspond to the coupling constraint (5.6c).

For brevity, we will only show how to solve (5.33) in a centralized manner using the proposed method and the distributed version can be derived in a similar manner. (5.33) can be represented

using the graph in Figure 5.1, where only node 0 is associated with an objective $f^T x$. Following the procedure in chapter 5.2.1, we introduce a duplicate for $x, \{z_i, t_i \mid i = 1, \ldots, N\}$, denoted by $y, \{w_i, v_i \mid i = 1, \ldots, N\}$, and (5.33) becomes

$$\begin{aligned}
\min \quad & f^T x && \text{(5.34a)}\\
\text{over} \quad & x, \{z_i, t_i \mid i = 1, \ldots, N\} && \text{(5.34b)}\\
& y, \{w_i, v_i \mid i = 1, \ldots, N\} && \text{(5.34c)}\\
\text{s.t.} \quad & A_i x - y_i + b_i = 0 \quad i = 1, \ldots, N && \text{(5.34d)}\\
& c_i^T x - t_i + d_i = 0 \quad i = 1, \ldots, N && \text{(5.34e)}\\
& \|w_i\|_2 \le v_i, \quad i = 1, \ldots, N, && \text{(5.34f)}\\
& y = x && \text{(5.34g)}\\
& w_i = z_i, \quad i = 1, \ldots, N && \text{(5.34h)}\\
& v_i = t_i, \quad i = 1, \ldots, N. && \text{(5.34i)}
\end{aligned}$$

Note that the problem (5.14) in the y-update has a closed form solution and can be solved efficiently. We show below the problem (5.11) in the x-update also has a closed form solution. Let λ, μ_i, γ_i denote the Lagrangian multiplier associated with (5.34g)–(5.34i), respectively. At each iteration k, the corresponding problem for node 0 in the x-update is

$$\min_x \frac{\rho}{2}\|y^k - x\|_2^2 - (\lambda^k)^T x,$$

whose closed form solution $x = y^k + \frac{\lambda^k}{\rho}$. For node $i = 1, \ldots, N$, the corresponding problem is

$$\begin{aligned}
\min \quad & \frac{\rho}{2}\left(\|w_i^k - z_i\|_2^2 + (t_i^k - v_i)^2\right) - (\mu_i^k)^T w_i - \gamma_i^k v_i\\
\text{over} \quad & w_i, v_i\\
\text{s.t.} \quad & \|w_i\|_2 \le v_i,
\end{aligned} \qquad \text{(5.35)}$$

which can be written equivalently as

$$\min_{v_i} \quad F(v_i) + \frac{\rho}{2}(t_i^k - v_i)^2 - \gamma_i^k v_i, \qquad \text{(5.36)}$$

where

$$\begin{aligned}
F(v_i) = \min_{w_i} \quad & \frac{\rho}{2}\|y_i^k - w_i\|_2^2 - (\mu_i^k)^T w_i\\
\text{s.t.} \quad & \|w_i\|_2 \le v_i,
\end{aligned} \qquad \text{(5.37)}$$

whose solution

$$F(v_i) = \begin{cases} -\frac{\|\mu_i^k\|_2^2}{2\rho} - (\mu_i^k)^T y_i^{k+1}, & \text{if } v_i \geq \|\frac{\mu_i^k}{\rho} + y_i^k\|_2 \\ \frac{\rho}{2}\left(v_i^2 + \|y_i^k\|_2^2\right) - \|\mu_i^k + \rho y_i^k\|_2 v_i, & \text{otherwise.} \end{cases}$$

Substitute it into (5.36), and we can obtain the closed form solution to (5.35) as:

$$(w_i, v_i) = \begin{cases} (y_i^k + \frac{\mu_i^k}{\rho}, t_i^k + \frac{\gamma_i^k}{\rho}), & \text{if } t_i^k + \frac{\gamma_i^k}{\rho} \geq \|\frac{\mu_i^k}{\rho} + y_i^k\|_2 \\ \frac{1}{2}\left(t_i^k + \frac{\gamma_i^k}{\rho} + \|\frac{\mu_i^k}{\rho} + y_i^k\|_2\right) \times \left(\frac{\mu_i^k + \rho y_i^k}{\|\mu_i^k + \rho y_i^k\|_2}, 1\right), & \text{otherwise.} \end{cases}$$

5.4 Conclusion

In this chapter, we first review the standard alternating direction method of multipliers (ADMM), which blends the decomposability of dual decomposition and superior convergence performance of augmented Lagrangian. Using the standard ADMM, we then develop efficient centralized and distributed algorithms that solve a broad class of graphical optimization problems.

Chapter 6

Distributed OPF Algorithm: Balanced Radial Distribution Networks

The optimal power flow (OPF) problem is fundamental in power system operations and planning. Large-scale renewable penetration in distribution networks calls for real-time feedback control, and hence the need for fast and distributed solutions for OPF. This is difficult because OPF is nonconvex and Kirchhoff's laws are global. In this chapter we propose a solution for *balanced* radial distribution networks. It exploits the results in Chapter 3.1 that suggests solving for a globally optimal solution of OPF over a radial network through the second-order cone program (SOCP) relaxation. Our distributed algorithm is based on the ADMM based algorithm proposed in Chapter 5. Unlike standard ADMM algorithms that often require iteratively solving optimization subproblems in each ADMM iteration, our decomposition allows us to derive closed form solutions for these subproblems, and greatly speed up each ADMM iteration.

Literature The optimal power flow (OPF) problem seeks to optimize certain objective such as power loss and generation cost subject to power flow equations and operational constraints. The continued growth of highly volatile renewable sources on distribution systems calls for real-time feedback control. Solving the OPF problems in such an environment has at least two challenges. First, the OPF problem is hard to solve because of its non-convex feasible set. We have addressed this issue through convex relaxation in Chapter 3.

Second, most algorithms proposed in the literature are centralized and meant for applications in today's energy management systems that centrally schedule a relatively small number of generators. In future networks that simultaneously optimize (possibly real-time) the operation of a large number of intelligent endpoints, a centralized approach will not scale because of its computation and communication overhead. In this chapter we address this challenge by developing a distributed algorithm that solves the SOCP relaxation of OPF for balanced radial distribution networks.

Various distributed algorithms have been developed to solve the OPF problem. Through optimization decomposition, the original OPF problem is decomposed into many local subproblems that can be solved in parallel. Based on the optimization decomposition used, existing algorithms roughly fall into four categories:

- **Dual decomposition**: Dual decompositions are applied to the convexified OPF problem in [54,55] under certain assumptions. In [55], only voltage constraints are considered and power injection constraints are ignored. On the other hand, in [54], voltage magnitude is assumed to be fixed for each bus. However, in typical distribution networks, those assumptions usually do not hold and the algorithms in [54, 55] do not apply directly.

- **Method of multipliers**: Different method of multipliers are leveraged to develop distributed OPF algorithms, including some of the early works [5, 52], which do not deal with the nonconvexity issue of the OPF problem. Therefore, the convergence of these algorithms is not guaranteed. In contrast, algorithms for the convexified OPF problem are proposed to guarantee convergence, e.g. the algorithm in [23] is based on SDP relaxation for the bus injection model and [57] is based on SOCP relaxation for the branch flow model.

- **ADMM**: Most of the recent algorithms are based on ADMM, e.g. [21,53,82], because ADMM blends the distributed structure of dual decomposition and the convergence property of augmented Lagrangian. In [82], ADMM is applied to the nonconvex OPF and convergence is not guaranteed. In [53], ADMM is applied to the DC-OPF model, which is extensively used in transmission networks but is a less accurate model for distribution networks. In [21], it is applied to SDP relaxation for the bus injection model.

- **Optimality condition decomposition**: Optimality condition decomposition [42, 67] only requires solving a single Newton step for each subproblem, instead of the optimal solutions as required by the above methods. However, convergence is not guaranteed because it depends on the optimal solutions to the problem, which are not known a prior.

One of the key performance metrics of a distributed algorithm is the time of convergence (ToC), which is the product of the number of iterations and the computation time to solve the subproblems in each iteration. To our knowledge, all the distributed OPF algorithms in the literature rely on generic iterative optimization solvers, which are computationally intensive, to solve the optimization subproblems. In this chapter, we will improve ToC by reducing the computation time for each subproblem.

Summary We develop a scalable distributed algorithm through decomposing the *convexified* OPF (ROPF) problem into many local subproblems based on the ADMM based algorithm proposed in

Chapter 5.2.2. In particular, the ROPF problem can be cast into the form (5.6) and the proposed algorithm has two advantages:

1. We provide a sufficient condition, which holds for practical applications, for the existence of closed form solutions to the optimization subproblems, thus eliminating the need for an iterative procedure to solve a SOCP problem for each ADMM iteration.

2. Communication is only required between adjacent buses.

We demonstrate the scalability of the proposed algorithm using a real-life network. In particular, we show that the algorithm converges within 0.6s on a laptop for a 2,065-bus system. To show the superiority of deriving close form solutions of each subproblems, we compare the computation time for solving a subproblem by our algorithm and an off-the-shelf optimization solver (SDPT3, [85]). Our solver requires on average 6.8×10^{-4}s while SDPT3 requires on average 0.5s. We also show that the convergence rate is mainly determined by the diameter [1] of the network through simulating the algorithm on networks of different topologies.

6.1 Problem formulation

The OPF problem and its SOCP relaxation for balanced networks are discussed in Chapter 3. Because the focus of this chapter is on network with radial topology, we will first simplify the notations used in Chapter 3, which allows arbitrary network topology. The following assumptions are made throughout this chapter.

A1 : The network graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ has a tree topology.

A2 : There is one substation indexed by 0 in the network, i.e. $\mathcal{N}_s = \{0\}$.

A3 : The SOCP relaxation is exact, i.e. the solution to the ROPF problem (3.9) is feasible for the original OPF problem (3.7).

A4 : There exists a closed form solution to the following optimization problem for all $i \in \mathcal{N}$

$$\begin{aligned} \min \quad & f_i(s) + \frac{\rho}{2} \|s - \hat{s}\|_2^2 \\ \text{over} \quad & s \\ \text{s.t.} \quad & s \in \mathcal{I}_i \end{aligned} \tag{6.1}$$

given any constants $\hat{s}$ and ρ.

[1] The diameter of a graph is defined as the number of hops between two furthest nodes.

Under assumption A1 and A2, for each node $i \in \mathcal{N} \setminus \{0\}$, there is only one element in its ancestor set A_i. Thus we will abuse notation and denote A_i as i's unique ancestor. Consequently the notation for the line set $\mathcal{E}$ can also be simplified, and for each directed line connecting node i and its ancestor A_i we will simply denote it as i instead of (i, A_i). Therefore the line set becomes $\mathcal{E} = \{1, \ldots, n\}$.

Using the simplified notations, the OPF problem (3.7) can be written as:

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{N}} f_i(s_i) \\
\text{over} \quad & v, s, S, \ell \\
\text{s.t.} \quad & v_{A_i} - v_i + z_i S_i^* + S_i z_i^* - \ell_i |z_i|^2 = 0 && i \in \mathcal{E} \\
& \sum_{i \in C_i} (S_j - z_j \ell_j) - S_i + s_i = 0 && i \in \mathcal{N} \\
& |S_i|^2 = v_i \ell_i && i \in \mathcal{E} \\
& s_i \in \mathcal{I}_i && i \in \mathcal{N} \\
& \underline{v}_i \le v_i \le \overline{v}_i && i \in \mathcal{N}
\end{aligned}$$

Under assumption A3, we can solve the ROPF problem (3.9) to obtain the optimal solution for OPF. The ROPF problem can be written explicitly as:

$$\begin{aligned}
\min \quad & \sum_{i \in \mathcal{N}} f_i(s_i) && && \text{(6.3a)} \\
\text{over} \quad & v, s, S, \ell && && \text{(6.3b)} \\
\text{s.t.} \quad & v_{A_i} - v_i + z_i S_i^* + S_i z_i^* - \ell_i |z_i|^2 = 0 && i \in \mathcal{E} && \text{(6.3c)} \\
& \sum_{i \in C_i} (S_j - z_j \ell_j) - S_i + s_i = 0 && i \in \mathcal{N} && \text{(6.3d)} \\
& |S_i|^2 \le v_i \ell_i && i \in \mathcal{E} && \text{(6.3e)} \\
& s_i \in \mathcal{I}_i && i \in \mathcal{N} && \text{(6.3f)} \\
& \underline{v}_i \le v_i \le \overline{v}_i && i \in \mathcal{N}. && \text{(6.3g)}
\end{aligned}$$

Denote

$$\begin{aligned}
x_i &:= \{v_i, s_i, S_i, \ell_i\} && \text{(6.4)} \\
\mathcal{K}_i &:= \{x_i \mid |S_i|^2 \le v_i \ell_i,\ s_i \in \mathcal{I}_i,\ \underline{v}_i \le v_i \le \overline{v}_i\}. && \text{(6.5)}
\end{aligned}$$

Then (6.3) takes the form of (5.6), where (6.3c)–(6.3d) correspond to (5.6c), and (6.3e)–(6.3g) correspond to (5.6d).

A4 is a technical assumption that is required for the existence of closed form solutions for all

the subproblems when the algorithm in Chapter 5.2.2 is applied to solve the ROPF problem. In practice, the objective function $f_i(s)$ usually takes the form of $f_i(s) := \frac{\alpha_i}{2}p^2 + \beta_i p$ $(\alpha_i, \beta_i \geq 0)$, which models both line loss and generation cost minimization as discussed in Chapter 3.1.2. For the injection region $\mathcal{I}_i$, it usually takes either (3.5a) or (3.5b). It is shown in Appendix 6.B that there exist closed form solutions for all of those cases. Thus, (5.27) can be solved efficiently for practical applications.

6.2 Distributed OPF Algorithm on Balanced Networks

We assume A1–A4 and now derive a distributed algorithm that solves the ROPF problem (6.3) based on the algorithm in Chapter 5.2.2. Specificially, using the ADMM based algorithm developed in Chapter 5.2.2, the global ROPF problem is decomposed into local subproblems that can be solved in a distributed manner with only neighborhood communication. In addition, we provide a sufficient condition for the existence of closed form solutions to all the optimization subproblems. Compared with existing methods, e.g. [5, 21, 23, 52, 54, 55, 57, 82], that use generic iterative optimization solver to solve each subproblem, the computation time is improved by more than 1000 times.

Note that in Chapter 5.2.2, the proposed algorithm requires specifying the partitions of set $\mathcal{K}_i$ for each $i \in \mathcal{N}$. For our application, we use $R_i = 0$, i.e. (5.27) is the same as (5.12), for $i \in \mathcal{N}$, which is sufficient to have a closed form solution.

Recall that there is always a closed form solution to the optimization subproblem (5.30) in the y-update. Now we will show if the objective function $f_i(s)$ and injection region $\mathcal{I}_i$ satisfy A4, the subproblems in the x-update can also be solved with closed form solutions for the ROPF problem.

Based on Chapter 5.2.2, we first need to transform (6.3) into the form of (5.23). Note that (6.3e)–(6.3g) are local constraints to bus i that correspond to $\mathcal{K}_i$ in (5.6d), and (6.3c), (6.3d) describe the coupling constraints among i and its neighbors that correspond to (5.6c). Since there is only one partition of $\mathcal{K}_i$, i.e. $R_i = 0$, we will skip the subscript r in x_{ir} and simply denote it as x_i. Then

ROPF can be written in the form of (5.23) as follows,

$$\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}} f_i(s_i^{(x)}) && && && (6.6a)\\
\text{over} \quad & x := \{x_i, i\in\mathcal{N}\} \\
& y := \{y_{ji},\ j\in N_i,\ i\in\mathcal{N}\} \\
\text{s.t.} \quad & v_{A_i i}^{(y)} - v_{ii}^{(y)} + z_i\left(S_{ii}^{(y)}\right)^* + S_{ii}^{(y)} z_i^* - \ell_{ii}^{(y)}|z_i|^2 = 0 && && i\in\mathcal{E} && (6.6b)\\
& \sum_{i\in C_i}\left(S_{ji}^{(y)} - z_j \ell_{ji}^{(y)}\right) - S_{ii}^{(y)} + s_{ii}^{(y)} = 0 && && i\in\mathcal{N} && (6.6c)\\
& |S_i^{(x)}|^2 \le v_i^{(x)}\ell_i^{(x)} && && i\in\mathcal{E} && (6.6d)\\
& s_i^{(x)} \in \mathcal{I}_i && && i\in\mathcal{N} && (6.6e)\\
& \underline{v}_i \le v_i^{(x)} \le \overline{v}_i && && i\in\mathcal{N} && (6.6f)\\
& x_i - y_{ij} = 0 && j\in N_i && i\in\mathcal{N}, && (6.6g)
\end{aligned}$$

where we put superscripts $(\cdot)^{(x)}$ and $(\cdot)^{(y)}$ on each variable to indicate whether the variables are updated in the x-update or y-update.

The problem (6.6) falls in the general form of (5.23) with $R_i = 0$ for all $i\in\mathcal{N}$, i.e. $\mathcal{K}_i$ in (6.5) is not partitioned. Specifically, (6.3c) models the voltage of its ancestor A_i as a function of the local variables of bus i, and (6.3d) describes the power flow balance among the set of children C_i and bus i itself. In other words, the coupling between a node and its ancestor is different from coupling with its children. Thus, each bus i does not need full duplicates of its neighbors' variables. It only needs voltage information $v_{A_i i}^{(y)}$ from its parent A_i and branch power $S_{ji}^{(y)}$ and current $\ell_{ji}^{(y)}$ information from its children $j\in C_i$ based on (6.6). Thus, y_{ij} contains only partial information about x_i, i.e.

$$y_{ij} \quad := \quad \begin{cases} (S_{ii}^{(y)}, \ell_{ii}^{(y)}, v_{ii}^{(y)}, s_{ii}^{(y)}) & j = i \\ (S_{iA_i}^{(y)}, \ell_{iA_i}^{(y)}) & j = A_i \\ (v_{ij}^{(y)}) & j\in C_i. \end{cases}$$

As a result, x_i and y_{ij} do not consist of the same component for $j\neq i$ in the consensus constraint (6.6g). Here, we abuse notation and $x_i - y_{ij}$ is composed of the components that appear in both x_i and y_{ij}, i.e.

$$x_i - y_{ij} := \begin{cases} (S_i^{(x)} - S_{ii}^{(y)}, \ell_i^{(x)} - \ell_{ii}^{(y)}, v_i^{(x)} - v_{ii}^{(y)}, s_i^{(x)} - s_{ii}^{(y)}) & j = i \\ (S_i^{(x)} - S_{iA_i}^{(y)}, \ell_i^{(x)} - \ell_{iA_i}^{(y)}) & j = A_i \\ (v_i^{(x)} - v_{ij}^{(y)}) & j\in C_i \end{cases}.$$

Table 6.1: Multipliers associated with constraints(6.6g)

$\mu_{ii}^{(1)}$:	$S_i^{(x)} = S_{ii}^{(y)}$	$\mu_{ii}^{(2)}$:	$\ell_i^{(x)} = \ell_{ii}^{(y)}$
$\mu_{ii}^{(3)}$:	$v_i^{(x)} = v_{ii}^{(y)}$	$\mu_{ii}^{(4)}$:	$s_i^{(x)} = s_{ii}^{(y)}$
$\mu_{iA_i}^{(1)}$:	$S_{iA_i}^{(x)} = S_{iA_i}^{(y)}$	$\mu_{iA_i}^{(2)}$:	$\ell_i^{(x)} = \ell_{iA_i}^{(y)}$
μ_{ij}:	$v_i^{(x)} = v_{ij}^{(y)}$		

Recall that the local variables to each node i include its original copy x_i, its "observation" of variables y_{ji} from its neighbor N_i, and the corresponding multiplier μ_{ji}. Then the set of local variables $\mathcal{A}_i$ (defined in (5.25)) for each bus i can be written explicitly as

$$\begin{aligned}\mathcal{A}_i &:= \{x_i\} \cup \{y_{ji}, \mu_{ji} \mid j \in N_i\} \\ &= \{S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)}, s_i^{(x)}\} \cup \{S_{ii}^{(y)}, \ell_{ii}^{(y)}, v_{ii}^{(y)}, s_{ii}^{(y)}, \mu_{ii}^{(1)}, \mu_{ii}^{(2)}, \mu_{ii}^{(3)}, \mu_{ii}^{(4)}\} \\ &\quad \cup \{S_{ji}^{(y)}, \ell_{ji}^{(y)}, \mu_{ji}^{(1)}, \mu_{ji}^{(2)} \mid j \in C_i\} \cup \{v_{A_i i}^{(y)}, \mu_{A_i i}\},\end{aligned}$$

where μ denotes the Lagrangian multipliers associated with (6.6g) and each component is defined in Table 6.1.

Next, we will show how to solve the subproblem (5.27) in the x-update with closed form solution under A4. For ease of presentation, we remove the iteration number k in (5.3) for all the variables, which will be updated accordingly after each subproblem is solved.

In the x-update, the subproblem (5.27) solved by each bus i can be written explicitly as

$$\begin{aligned} \min \quad & H_i(x_i) && \text{(6.7a)} \\ \text{over} \quad & x_i \\ \text{s.t.} \quad & |S_i^{(x)}|^2 \le v_i^{(z)} \ell_i^{(x)} && \text{(6.7b)} \\ & s_i^{(x)} \in \mathcal{I}_i && \text{(6.7c)} \\ & \underline{v}_i \le v_i^{(x)} \le \overline{v}_i. && \text{(6.7d)} \end{aligned}$$

$H_i(x_i)$ is defined in (5.28) and

$$\begin{aligned} H_i(x_i) &= f_i(s_i^{(x)}) - \sum_{j \in N_i} \langle \mu_{ij}, x_i \rangle + \frac{\rho}{2} \sum_{j \in N_i} \|x_i - y_{ij}\|_2^2 \qquad \text{(6.8)} \\ &= \rho H_i^{(1)}(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)}) + H_i^{(2)}(s_i^{(x)}) + \text{constant}, \end{aligned}$$

where

$$H_i^{(1)}(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)}) = |S_i^{(x)} - \hat{S}_i|^2 + |\ell_i^{(x)} - \hat{\ell}_i|^2 + \frac{|C_i|+1}{2}|v_i^{(x)} - \hat{v}_i|^2 \tag{6.9}$$

$$H_i^{(2)}(s_i^{(x)}) = f_i(s_i^{(x)}) + \frac{\rho}{2}\|s_i^{(x)} - \hat{s}_i\|_2^2. \tag{6.10}$$

The last step in (6.8) is obtained by completing the square and the variables labeled with hat are constants for the x-update, i.e.

$$\begin{aligned}
\hat{S}_i &= \frac{S_{ii}^{(y)} + S_{iA_i}^{(y)}}{2} + \frac{\mu_{ii}^{(1)} + \mu_{iA_i}^{(1)}}{2\rho} \\
\hat{\ell}_i &= \frac{\ell_{ii}^{(y)} + \ell_{iA_i}^{(y)}}{2} + \frac{\mu_{ii}^{(2)} + \mu_{iA_i}^{(2)}}{2\rho} \\
\hat{v}_i &= \frac{v_{ii}^{(y)} + \sum_{j\in C_i} v_{ij}^{(y)}}{|C_i|+1} + \frac{\mu_{ii}^{(3)} + \sum_{j\in C_i}\mu_{ij}}{\rho(|C_i|+1)} \\
\hat{s}_i &= s_{ii}^{(y)} + \frac{\mu_{ii}^{(4)}}{\rho}.
\end{aligned}$$

Thus, the objective (6.7a) in (6.7) can be decomposed into two parts, where the first part $H^{(1)}(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ involves variables $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ and the second part $H^{(2)}(s_i^{(x)})$ involves $s_i^{(x)}$. Note that the constraint (6.7b)–(6.7d) can also be separated into two independent constraints. Variables $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ only depend on (6.7b)–(6.7d) and $s_i^{(x)}$ depends on (6.7c). Hence, (6.7) can be decomposed into two subproblems, where the first one solves the optimal $(S_i^{(x)}, \ell_i^{(x)}, v_i^{(x)})$ and the second one solves the optimal $s_i^{(x)}$. The first subproblem can be written explicitly as

$$\begin{aligned}
\min \quad & \left|S_i^{(x)} - \hat{S}_i\right|^2 + \left|\ell_i^{(x)} - \hat{\ell}_i\right|^2 + \frac{|C_i|+1}{2}\left|v_i^{(x)} - \hat{v}_i\right|^2 \\
\text{over} \quad & v_i^{(x)}, \ell_i^{(x)}, S_i^{(x)} \\
\text{s.t.} \quad & \left|S_i^{(x)}\right|^2 \le v_i^{(x)}\ell_i^{(x)} \\
& \underline{v}_i \le v_i^{(x)} \le \overline{v}_i,
\end{aligned} \tag{6.11}$$

which has a quadratic objective, a second order cone constraint and a box constraint. (6.11) has a closed form solution and the procedure is illustrated in Appendix 6.A. Compared with using generic iterative solver, the procedure is computationally efficient since it only requires one to solve the zero of three polynomials with a degree less than or equal to 4, which have closed form solutions.

The second subproblem is

$$\begin{aligned}
\min \quad & f_i\left(s_i^{(x)}\right) + \frac{\rho}{2}\left\|s_i^{(x)} - \hat{s}_i\right\|_2^2 \\
\text{over} \quad & s_i^{(x)} \\
\text{s.t.} \quad & s_i^{(x)} \in \mathcal{I}_i,
\end{aligned} \tag{6.12}$$

which has exactly the same structure as A4 and thus can be solved with closed form solution. Thus, all the subproblems involved in the x-update can be solved with closed form solutions and we have the following theorem.

Theorem 6.1. *Under assumption A4, there exists a closed form solution to the subproblem in the x-update* (5.27) *for ROPF with $R_i = 0$.*

In the y-update, the subproblem solved by each agent i takes the form of (5.30) and can be written as

$$\begin{aligned} \min \quad & G_i(\{y_{ji} \mid j \in N_i\}) \\ \text{over} \quad & \{y_{ji} \mid j \in N_i\} \\ \text{s.t.} \quad & v_{A_i i}^{(y)} - v_{ii}^{(y)} + z_i \left(S_{ii}^{(y)}\right)^* + S_{ii}^{(y)} z_i^* - \ell_{ii}^{(y)} |z_i|^2 = 0 \\ & \sum_{i \in C_i} \left(S_{ji}^{(y)} - z_j \ell_{ji}^{(y)}\right) - S_{ii}^{(y)} + s_{ii}^{(y)} = 0, \end{aligned} \tag{6.13}$$

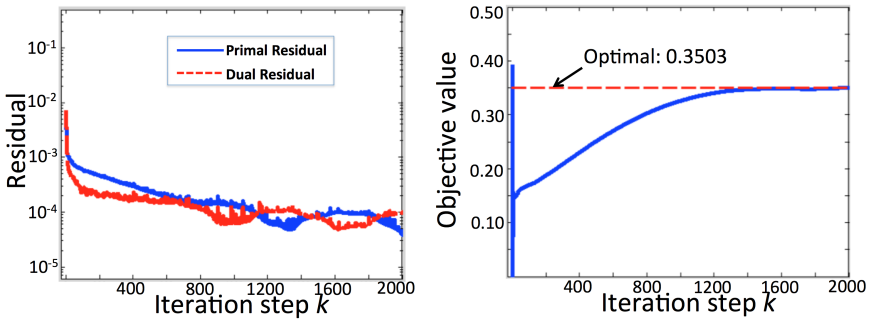
which has a closed form solution given in (5.16) and we do not reiterate here.

Algorithm 6.1 Initialization of the Algorithm

1: $v_i^{(x)} = 1$ for $i \in \mathcal{N}$
2: Randomly pick a point in $\mathcal{I}_i$ to initialize $s_i^{(x)}$ for $i \in \mathcal{N}$
3: Initialize $S^{(x)}$ by calling DFS(0)
4: $\ell_i^{(x)} = \frac{|S_i^{(x)}|^2}{v_i^{(x)}}$ for $i \in \mathcal{N}$
5: $y_{ji} = x_j$ for $j \in N_i$ and $i \in \mathcal{N}$

6: **function** DFS(i)
7: $\quad S_i^{(x)} = s_i^{(x)}$
8: $\quad$ **for** $j \in C_i$ **do**
9: $\quad\quad S_i^{(x)} += \text{DFS}(j)$
10: $\quad$ **end for**
11: $\quad$ **return** $S_i^{(x)}$
12: **end function**

Finally, we specify the initialization and stopping criteria for the algorithm. A good initialization can greatly reduce the number of iterations for convergence. We use the following initialization suggested by our empirical results. We first initialize the x variable. The voltage magnitude square $v_i^{(x)} = 1$. The power injection $s_i^{(x)}$ is randomly picked from a feasible point in the feasible region $\mathcal{I}_i$. The branch power $S_i^{(x)}$ is assumed to be the aggregate power injection $s_i^{(x)}$ from the nodes connected by line i (Note that the network has a tree topology.). The branch current $\ell_i^{(x)} = \frac{|S_i^{(x)}|^2}{v_i^{(x)}}$ according to (3.2c). The y variables are initialized using the corresponding x variable according to (6.6g). Intuitively, the above initialization procedure can be interpreted as finding a solution to the branch flow equation (3.2) assuming zero impedance on all the lines. The procedure is formally stated in



(a) Primal and dual residual (b) Objective value

Figure 6.1: Simulation results for 2065 bus distribution network.

Algorithm 6.1.

For the stopping criteria, there is no general rule for ADMM based algorithm and it usually hinges on the particular problem. In [11], it is argued that a reasonable stopping criteria is that both the primal residual r^k defined in (5.4a) and the dual residual s^k defined in (5.4b) are below $10^{-4}\sqrt{|\mathcal{N}|}$. We adopt this criteria and the empirical results show that the solution is accurate enough. The pseudo code for the complete algorithm is summarized in Algorithm 6.2.

Algorithm 6.2 Distributed OPF algorithm on Balanced Radial Networks

Input:network $\mathcal{G}(\mathcal{N},\mathcal{E})$, power injection region $\mathcal{I}_i$, voltage region $(\underline{v}_i,\overline{v}_i)$, line impedance z_i for $i\in\mathcal{N}$.
Output:voltage v, power injection s.

Initialize the x and y variables using Algorithm 6.1.
while $r^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **or** $s^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **do**
    In the x-update, each agent i solves (6.7) to update x.
    In the y-update, each agent i solves (6.13) to update y.
    In the multiplier update, update μ by (5.3c).
end while

6.3 Case Study

In this section, we first demonstrate the scalability of the proposed distributed algorithm by testing it on the model of a 2,065-bus distribution circuit in the service territory of the Southern California Edison (SCE). We also show the advantage of deriving closed form expression by comparing the computation time of solving the subproblems using off-the-shelf solver (SDPT3 [85] through CVX [37]) and using our algorithm. Second, we simulate the proposed algorithm on networks of different sizes to understand the impacts of network size and diameter on the convergence rate. The algorithm is implemented in Matlab 2014 and run on a Macbook pro 2014 with an i5 dual core processor.

Table 6.1: Statistics of different networks

Network	Diameter	Iteration	Total Time(s)	Avg time(s)
2065Bus	64	1114	1153	0.56
1313Bus	53	671	471	0.36
792Bus	45	524	226	0.29
363Bus	36	289	112	0.24
108Bus	16	267	16	0.14

6.3.1 Simulation on a 2,065-bus circuit

In the 2,065-bus distribution network, there are 1,409 household loads, whose power consumptions are within 0.07kw–7.6kw, and 142 commercial loads, whose power consumptions are within 5kw–36.5kw. There are 135 rooftop PV panels, whose nameplates are within 0.7–4.5kw, distributed across the 1,409 houses.

The network is unbalanced three phase. We assume that the three phases are decoupled such that the network becomes identical single phase network. The voltage magnitude at each load bus is allowed to lie within $[0.95, 1.05]$ per unit (pu), i.e. $\overline{v}_i = 1.05^2$ and $\underline{v}_i = 0.95^2$ for $i \in \mathcal{N} \setminus \{0\}$. The control devices are the rooftop PV panels whose real and reactive power injections are controlled. The objective is to minimize power loss across the network, namely $f_i(s_i) = p_i$ for $i \in \mathcal{N}$. Each bus is a node and there are 2,065 nodes in the network that solve the OPF problem in a distributed manner.

We mainly focus on the time of convergence (ToC) for the proposed distributed algorithm. The algorithm is run on a single machine. To roughly estimate the ToC (excluding communication overhead) if the algorithm is run on distributed machines, we divide the total time by the number of nodes. Recall that the stopping criteria is that both the primal and dual residual are below $10^{-4}\sqrt{|\mathcal{N}|}$ and Figure 6.1a illustrates the evolution of $r^k/\sqrt{|\mathcal{N}|}$ and $s^k/\sqrt{|\mathcal{N}|}$ versus iterations k. The stopping criteria are satisfied after $1,114$ iterations. The evolution of the objective value is illustrated in Figure 6.1b. It takes 1,153s to run 1,114 iterations on a single computer. Then the ToC is roughly 0.56s if we implement the algorithm in a distributed manner not counting communication overhead.

Finally, we show the advantage of closed form solution by comparing the computation time of solving the subproblems by an off-the-shelf solver (SDPT3) and by our algorithm. In particular, we compare the average computation time of solving the subproblem in both the x-update and the y-update step. In the x-update step, the average time required to solve the subproblem is 1.7×10^{-4}s for the proposed algorithm but 0.2s for SDPT3. In the y-update step, the average time required to solve the subproblem is 5.1×10^{-4}s for the proposed algorithm but 0.3s for SDPT3. Thus, each ADMM iteration only takes about 6.8×10^{-4}s for the proposed algorithm but 0.5s for using the

(a) Line networks (b) Fat tree networks

Figure 6.2: Topologies for tree and fat tree networks.

Table 6.2: Statistics of line and fat tree networks

Size	# of iterations (Line)	# of iterations (Fat tree)
5	43	31
10	123	51
15	198	148
20	286	87
25	408	173
30	838	119
35	1471	187
40	2201	109
45	2586	182
50	3070	234

iterative algorithm, which is a 1,000x speedup.

6.3.2 Rate of Convergence

In section 6.3.1, we demonstrate that the proposed distributed algorithm can dramatically reduce the computation time within each iteration and therefore is scalable to a large practical 2,065-bus distribution network. The time of convergence(ToC) is determined both by the computation time required within each iteration and the number of iterations. In this subsection, we study the number of iterations.

To the best of our knowledge, most of the works on convergence rate for ADMM based algorithms study how the primal/dual residual changes as the number of iterations increases. Specifically, it is proved in [41, 88] that the general ADMM based algorithms converge linearly under certain assumptions. Here, we study empirically how the rate of convergence depends on the network size N and diameter D, i.e. given the termination criteria in Algorithm 6.2, the impact of network size and diameter on the number of iterations.

First, we simulate the algorithm on different networks (that are subnetworks of the 2,065-bus system) and some statistics are given in Table 6.1. For simplicity, we assume the number of iterations T to converge takes the linear form $T = aN + bD$. Using the data in Table 6.1, the parameters

$a = 0.34$, $b = 5.53$ give the least square error. It means that the network diameter has a stronger impact than the network size on the rate of convergence.

To further illustrate the phenomenon, we simulate the algorithm on two extreme cases: 1) Line network in Fig. 6.2a, whose diameter is the largest given the network size, and 2) Fat tree network in Fig. 6.2b, whose diameter is the smallest (2) given the network size. In Table 6.2, we record the number of iterations for both line and fat tree network of different sizes. For line network, the number of iterations increases notably as the size increases. For fat tree network, the trend is much less prominant compared to the line network.

6.4 Conclusion

In this chapter, we have developed an ADMM based distributed algorithm for the optimal power flow problem on balanced radial distribution networks. We have derived a closed form solution for the subproblems solved by each bus, and thus significantly reduced the computation time. Preliminary simulation shows that the algorithm is scalable to a 2,065-bus system and the optimization subproblems in each ADMM iteration are solved 1,000x faster than using generic iterative optimization solver SDPT3.

Appendix

6.A Solution Procedure for Problem (6.11)

Denote $x_1 := \mathbf{Re}(S_i^{(x)})$, $x_2 := \mathbf{Im}(S_i^{(x)})$, $x_3 := \sqrt{\frac{|C_i|+1}{2}} v_i^{(x)}$ and $x_4 := \ell_i^{(x)}$. Then the optimization problem (6.11) can be written equivalently as

$$\begin{aligned}
\min \quad & \sum_{i=1}^{4} (x_i^2 + c_i x_i) && \text{(6.14a)} \\
\text{over} \quad & x_1,\ x_2,\ x_3,\ x_4 \\
\text{s.t.} \quad & \frac{x_1^2 + x_2^2}{x_3} \leq k^2 x_4 && \text{(6.14b)} \\
& \underline{x}_3 \leq x_3 \leq \overline{x}_3, && \text{(6.14c)}
\end{aligned}$$

where $\overline{x}_3 > \underline{x}_3 > 0$ and c_i, k are constants that hinges on the constants in (6.11).

Next, we will derive a procedure that solves (6.14). Let $\gamma \geq 0$ denote the Lagrangian multiplier for constraint (6.14b) and $\underline{\lambda}, \overline{\lambda} \geq 0$ denote the Lagrangian multipliers for constraint (6.14c), then the Lagrangian of P1 is

$$L(x, \gamma, \lambda) = \sum_{i=1}^{4} (x_i^2 + c_i x_i) + \gamma \left(\frac{x_1^2 + x_2^2}{x_3} - k^2 x_4 \right) + \overline{\lambda}(x_3 - \overline{x}_3) - \underline{\lambda}(x_3 - \underline{x}_3).$$

The KKT optimality conditions imply that the optimal solution x^* together with the multipliers $\gamma^*, \underline{\lambda}^*, \overline{\lambda}^*$ satisfies the following equations. For ease of notations, we sometimes skip the superscript

$\star$ of the variables in the following analysis.

$$2x_1 + c_1 + 2\gamma\frac{x_1}{x_3} = 0 \tag{6.15a}$$

$$2x_2 + c_2 + 2\gamma\frac{x_2}{x_3} = 0 \tag{6.15b}$$

$$2x_3 + c_3 - \gamma\frac{x_1^2 + x_2^2}{x_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{6.15c}$$

$$2x_4 + c_4 - k^2\gamma = 0 \tag{6.15d}$$

$$\overline{\lambda}(x_3 - \overline{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \overline{x}_3 \tag{6.15e}$$

$$\underline{\lambda}(\underline{x}_3 - x_3) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{6.15f}$$

$$\gamma\left(\frac{x_1^2 + x_2^2}{x_3} - k^2 x_4\right) = 0 \quad \gamma \geq 0 \quad \frac{x_1^2 + x_2^2}{x_3} \leq k^2 x_4. \tag{6.15g}$$

Lemma 6.2. *There exists a unique solution* $(x^*, \gamma^*, \underline{\lambda}^*, \overline{\lambda}^*)$ *to* (6.15) *if* $\overline{x}_3 > \underline{x}_3 \geq 0$.

Proof. P1 is feasible since $z = (0, 0, \underline{x}_3, 1)$ satisfies (6.14b)-(6.14c). In addition, P1 is a strictly convex optimization problem since the objective (6.14a) is a strictly convex function of z and the constraints (6.14b) and (6.14c) are also convex. Therefore, there exists a unique solution z^* to P1, which indicates there exists a unique solution $(x^*, \gamma^*, \underline{\lambda}^*, \overline{\lambda}^*)$ to the KKT optimality conditions (6.15). □

Lemma 6.2 says that there exists a unique solution to (6.15), which is also the optimum to P1. In the following, we will solve (6.15) through enumerating value of the multipliers $\gamma, \overline{\lambda}, \underline{\lambda}$. Specifically, we first assume $\gamma^* = 0$ (Case 1 below), which is equivalent to assuming that the constraint (6.14b) is inactive. If there is a feasible solution to (6.15), it is the unique solution to (6.15). Otherwise, we assume $\gamma^* = 0$ (Case 2 below), which is equivalent to assume that the equality is obtained at optimality in (6.14b).

Case 1: If $\gamma = 0$, (6.15) becomes

$$2x_1 + c_1 = 0 \tag{6.16a}$$

$$2x_2 + c_2 = 0 \tag{6.16b}$$

$$2x_3 + c_3 + \overline{\lambda} - \underline{\lambda} = 0 \tag{6.16c}$$

$$2x_4 + c_4 = 0 \tag{6.16d}$$

$$\overline{\lambda}(x_3 - \overline{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \overline{x}_3 \tag{6.16e}$$

$$\underline{\lambda}(x_3 - \underline{x}_3) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{6.16f}$$

$$\frac{x_1^2 + x_2^2}{x_3} \leq k^2 x_4. \tag{6.16g}$$

The solution to (6.16a)–(6.16f) ignoring (6.16g) is

$$x_1 = -\frac{c_1}{2},\ x_2 = -\frac{c_2}{2},\ x_3 = \left[-\frac{c_3}{2}\right]_{\underline{x}_3}^{\overline{x}_3},\ x_4 = -\frac{c_4}{2}$$
$$\overline{\lambda} = -(2x_3 + c_3)1_{\{x_3=\overline{x}_3\}}, \underline{\lambda} = -(2x_3 + c_3)1_{\{x_3=\underline{x}_3\}},$$

and if the solution satisfies (6.16g), it is the solution to (6.15). Otherwise, we go to Case 2.

Case 2: If $\gamma > 0$, (6.15) becomes

$$2x_1 + c_1 + 2\gamma\frac{x_1}{x_3} = 0 \tag{6.17a}$$
$$2x_2 + c_2 + 2\gamma\frac{x_2}{x_3} = 0 \tag{6.17b}$$
$$2x_3 + c_3 - \gamma\frac{x_1^2 + x_2^2}{x_3^2} + \overline{\lambda} - \underline{\lambda} = 0 \tag{6.17c}$$
$$\overline{\lambda}(x_3 - \overline{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \overline{x}_3 \tag{6.17d}$$
$$\underline{\lambda}(x_3 - \underline{x}_3) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3 \tag{6.17e}$$
$$\gamma = \frac{1}{k^2}(2x_4 + c_4) \tag{6.17f}$$
$$x_4 = \frac{x_1^2 + x_2^2}{k^2 x_3}. \tag{6.17g}$$

Substitute (6.17g) into (6.17f), and we obtain

$$\gamma = \frac{1}{k^2}(2x_4 + c_4) = \frac{2(x_1^2 + x_2^2)}{k^4 x_3} + \frac{c_4}{k^2}. \tag{6.18}$$

Then substituting (6.17f) into (6.17a)-(6.17e), we can write (6.17) equivalently as

$$2 + \frac{c_1}{x_1} + 4\frac{(x_1^2 + x_2^2)}{k^4 x_3^2} + \frac{2c_4}{k^2 x_3} = 0 \tag{6.19a}$$
$$2 + \frac{c_2}{x_2} + 4\frac{(x_1^2 + x_2^2)}{k^4 x_3^2} + \frac{2c_4}{k^2 x_3} = 0 \tag{6.19b}$$
$$2 + \frac{c_3}{x_3} - 2\frac{(x_1^2 + x_2^2)^2}{k^4 x_3^4} - c_4\frac{x_1^2 + x_2^2}{x_3^3} + \frac{\overline{\lambda} - \underline{\lambda}}{x_3} = 0 \tag{6.19c}$$
$$\overline{\lambda}(x_3 - \overline{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \overline{x}_3 \tag{6.19d}$$
$$\underline{\lambda}(x_3 - \underline{x}_3) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3, \tag{6.19e}$$

where (6.19a)–(6.19c) are obtained through dividing both sides of (6.17a)–(6.17c) by x_1, x_2 and x_3, respectively. The variables γ, x_4 can be recovered via (6.17f) and (6.17g) after we solve (6.19).

By (6.19a) and (6.19b),

$$\frac{c_1}{x_1} = \frac{c_2}{x_2}$$

Denote $p := \frac{x_1}{c_1 x_3} = \frac{x_2}{c_2 x_3}$. Then (6.19) is equivalent to the following equations:

$$p = \frac{x_1}{c_1 x_3} = \frac{x_2}{c_2 x_3} \tag{6.20a}$$

$$2 + \frac{1}{p x_3} = -\left(\frac{4(c_1^2 + c_2^2)}{k^4} p^2 + 2\frac{c_4}{k^2 x_3}\right) \tag{6.20b}$$

$$2 + \frac{c_3}{x_3} = \frac{2(c_1^2 + c_2^2)^2}{k^4} p^4 + \frac{c_4(c_1^2 + c_2^2)}{k^2}\frac{p^2}{x_3} + \frac{\underline{\lambda} - \overline{\lambda}}{x_3} \tag{6.20c}$$

$$\overline{\lambda}(x_3 - \overline{x}_3) = 0 \quad \overline{\lambda} \geq 0 \quad x_3 \leq \overline{x}_3 \tag{6.20d}$$

$$\underline{\lambda}(x_3 - \underline{x}_3) = 0 \quad \underline{\lambda} \geq 0 \quad x_3 \geq \underline{x}_3, \tag{6.20e}$$

where (6.20b) is obtained by substituting $x_2 = c_2 p x_3$ into (6.19b), (6.20c) is obtained by substituting $x_1 = c_1 p x_3$, and $x_2 = c_2 p x_3$ into (6.19c). To solve (6.20), we further divide our analysis into two sub-cases depending on whether x_3^* hits the lower or upper bound.

- **Case 2.1**: $x_3^* = \overline{x}_3$. $(\underline{\lambda} = 0, \overline{\lambda} > 0)$ (The case of $x_3^* = \underline{x}_3$ can be solved using similar procedure.)

  We first substitute $x_3 = \overline{x}_3$ into (6.20b) and have

  $$\frac{4(c_1^2 + c_2^2)}{k^4} p^3 + \left(2\frac{c_4}{k^2 \overline{x}_3} + 2\right) p + \frac{1}{\overline{x}_3} = 0, \tag{6.21}$$

  whose solution[2] is denoted by p^*. Then substitute p^* and $\overline{x}_3$ into (6.20a), we can recover x_1^* and x_2^*. Then we can obtain $\gamma^*, \overline{\lambda}^*$ using (6.18) and (6.19c) by substituting $x_1^*, \ldots, x_4^*$. If $\gamma^*, \overline{\lambda}^* \geq 0$, they collectively solve (6.15). Otherwise, we go to Case 2.2.

- **Case 2.2**: $\underline{x}_3 < x_3^* < \overline{x}_3$ $(\underline{\lambda}, \overline{\lambda} = 0)$.

  Since $\overline{\lambda}$ and $\underline{\lambda} = 0$, (6.20) reduces to

  $$p = \frac{x_1}{c_1 x_3} = \frac{x_2}{c_2 x_3} \tag{6.22a}$$

  $$2 + \frac{1}{p x_3} = -\left(\frac{4(c_1^2 + c_2^2)}{k^4} p^2 + 2\frac{c_4}{k^2 x_3}\right) \tag{6.22b}$$

  $$2 + \frac{c_3}{x_3} = \frac{2(c_1^2 + c_2^2)^2}{k^4} p^4 + \frac{c_4(c_1^2 + c_2^2)}{k^2}\frac{p^2}{x_3}. \tag{6.22c}$$

[2]There are potentially multiple solutions and we need to check all the real solution p^* using the following procedure.

Dividing each side of (6.22b) by (6.22c) gives

$$\frac{2x_3 + \frac{1}{p}}{2x_3 + c_3} = -\frac{2}{(c_1^2 + c_2^2)p^2},$$

which implies

$$x_3 = -\frac{(c_1^2 + c_2^2)p + 2c_3}{2((c_1^2 + c_2^2)p^2 + 2)}. \tag{6.23}$$

Then substitute (6.23) into (6.22b), we have

$$\frac{(c_1^2 + c_2^2)p^2 + 2}{(c_1^2 + c_2^2)p^2 + 2c_3 p} - \frac{2(c_1^2 + c_2^2)}{k^4}p^2 + \frac{2c_4((c_1^2 + c_2^2)p^2 + 2)}{k^2((c_1^2 + c_2^2)p + 2c_3)} = 1,$$

which is equivalent to

$$\frac{(c_1^2 + c_2^2)^2}{k^4}p^4 + \frac{c_1^2 + c_2^2}{k^2}\left(\frac{2c_3}{k^2} - c_4\right)p^3 + \left(c_3 - \frac{2c_4}{k^2}\right)p - 1 = 0,$$

whose solution[2] is denoted by p^*. Substitute p^* into (6.23), and we can recover x_3^*, and then x_1^*, x_2^* can be recovered via (6.22a). γ^* is recovered using (6.18). If $\gamma^* \geq 0$, the corresponding solution solves (6.15).

6.B Solution Procedure for Problem (6.1).

We assume $f_i(s) := \frac{\alpha_i}{2}p^2 + \beta_i p$ $(\alpha_i, \beta_i \geq 0)$ and derive a closed form solution to (6.1).

6.B.1 $\mathcal{I}_i$ takes the form of (3.5a)

In this case, (6.1) takes the following form:

$$\begin{aligned} \min_{p,q} \quad & \frac{a_1}{2}p^2 + b_1 p + \frac{a_2}{2}q^2 + b_2 q \\ \text{s.t.} \quad & \underline{p}_i \leq p \leq \overline{p}_i \\ & \underline{q}_i \leq q \leq \overline{q}_i, \end{aligned}$$

where $a_1, a_2 > 0$ and b_1, b_2 are constants. Then the closed form solution is

$$p = \left[-\frac{b_1}{a_1}\right]_{\underline{p}_i}^{\overline{p}_i} \quad q = \left[\hat{-}\frac{2_1}{a_2}\right]_{\underline{q}_i}^{\overline{q}_i},$$

where $[x]_a^b := \min\{a, \max\{x, b\}\}$.

6.B.2 $\mathcal{I}_i$ takes the form of (3.5b)

The optimization problem (6.1) takes the following form:

$$\min_{p,q} \quad \frac{a_1}{2}p^2 + b_1 p + \frac{a_2}{2}q^2 + b_2 q \tag{6.24a}$$

$$\text{s.t.} \quad p^2 + q^2 \le c^2 \tag{6.24b}$$

$$p \ge 0, \tag{6.24c}$$

where $a_1, a_2, c > 0$, b_1, b_2 are constants. The solutions to (6.24) are given as below.

Case 1: $b_1 \ge 0$:

$$p^* = 0 \qquad q^* = \left[-\frac{b_2}{a_2}\right]_{-c}^{c}.$$

Case 2: $b_1 < 0$ and $\frac{b_1^2}{a_1^2} + \frac{b_2^2}{a_2^2} \le c^2$:

$$p^* = -\frac{b_1}{a_1} \qquad q^* = -\frac{b_2}{a_2}.$$

Case 3: $b_1 < 0$ and $\frac{b_1^2}{a_1^2} + \frac{b_2^2}{a_2^2} > c^2$:
First solve the following equation in terms of variable λ:

$$b_1^2(a_2 + 2\lambda)^2 + b_2^2(a_1 + 2\lambda)^2 = (a_1 + 2\lambda)^2(a_2 + 2\lambda)^2, \tag{6.25}$$

which is a polynomial with degree of 4 and has closed form expression. There are four solutions to (6.25), but there is only one strictly positive λ^*, which can be proved via the KKT conditions of (6.24). Then we can recover p^*, q^* from λ^* using the following equations:

$$p^* = -\frac{b_1}{a_1 + 2\lambda^*} \quad \text{and} \quad q^* = -\frac{b_2}{a_2 + 2\lambda^*}.$$

The above procedure to solve (6.24) is derived from standard applications of the KKT conditions of (6.24). For brevity, we skip the proof here.

Chapter 7

Distributed OPF Algorithm: Unbalanced Radial Distribution Networks

In Chapter 6, we have developed an ADMM based distributed algorithm that solves the OPF problem on *balanced* radial networks. We show that by using the technique in Chapter 5.2.2, a closed form expression for each optimization subproblem can be derived. Thus, the computation time is reduced significantly and the proposed algorithm is scalable to a 2,065-bus network.

In this chapter, we will develop an ADMM based distributed algorithm that solves the OPF problem on *unbalanced* radial networks. The main idea is still to exploit the tree topology of distribution networks and decompose the OPF problem in a way that the subproblems in each ADMM macro-iteration can be solved efficiently. In contrast to the algorithm in Chapter 6 for balanced networks, where there exist closed form solutions for all the subproblems, the optimization subproblems reduce to either closed form solutions or eigenvalue problems whose size remains constant as the network size scales up. However, we show that the proposed algorithm can still speed up the algorithm by 100x compared with using the generic iterative optimization solvers. We present simulations on IEEE 13, 34, 37, and 123 bus unbalanced distribution networks to illustrate the scalability and optimality of the proposed algorithm.

Summary We develop a scalable distributed algorithm that solves the *convexified* OPF (ROPF) problem for unbalanced radial networks. In particular, we decompose the ROPF problem into local subproblems that can be solved in parallel based on the ADMM based algorithm in Chapter 5.2.2. The proposed algorithm has two advantages:

1. We provide a sufficient condition, which holds for practical applications, for the existence of a computationally efficient solution (either solved with closed form solution or through eigen-decomposition of a 6 by 6 matrix) to the optimization subproblems. It eliminates the need for

an iterative procedure to solve SDP problems for each ADMM iteration.

2. Communication is only required between adjacent buses.

We demonstrate the scalability of the proposed algorithms using standard IEEE test networks [48]. The proposed algorithm converges within 0.7s on all the IEEE-13, 34, 37, 123 bus systems. To show the superiority of using the proposed procedure to solve each subproblem, we compare the computation time for solving a subproblem by our algorithm and an off-the-shelf SDP optimization solver (SDPT3 [85]). Our solver requires on average 3.8×10^{-3}s while SDPT3 requires on average 0.58s, which is a 100x speedup on a laptop.

The rest of this chapter is structured as follows. We formulate the OPF problem on unbalanced radial networks in section 7.1. The distributed algorithm is developed in section 7.2 and simulation results on IEEE standard test networks are demonstrated in section 7.3.

7.1 Problem formulation

The OPF problem and its SDP relaxation on *unbalanced* networks are discussed in Chapter 3.2. Because the focus of this chapter is on network with radial topology, we will first simplify the notations used in Chapter 3, which allows arbitrary network topology. Similar to Chapter 6, the following assumptions are made throughout this chapter.

A1 : The network graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ has a tree topology.

A2 : There is one substation indexed by 0 in the network, i.e. $\mathcal{N}_s = \{0\}$.

A3 : The SDP relaxation is exact, i.e. the solution to the ROPF problem (3.17) is feasible to the original OPF problem (3.16).

A4 : There exists a closed form solution to the following optimization problem for all $i \in \mathcal{N}$ and $\phi \in \Phi_i$

$$\begin{aligned} \min \quad & f_i^\phi\left(s^\phi\right) + \frac{\rho}{2}\left\|s^\phi - \hat{s}^\phi\right\|_2^2 \\ \text{over} \quad & s^\phi \\ \text{s.t.} \quad & s^\phi \in \mathcal{I}_i^\phi \end{aligned} \tag{7.1}$$

given any constant $\hat{s}^\phi$ and ρ.

Under assumption A1 and A2, for each node $i \in \mathcal{N} \setminus \{0\}$, there is only one element in its ancestor set A_i. Thus we will abuse notation and denote A_i as i's unique ancestor. Consequently the notation for the line set $\mathcal{E}$ can also be simplified and for each directed line connecting node i and its ancestor A_i, we will denote by i instead of (i, A_i). Therefore the line set becomes $\mathcal{E} = \{1, \ldots, n\}$.

Using the simplified notations, the OPF problem (3.16) for unbalanced radial networks can be written as:

$$
\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}}\sum_{\phi\in\Phi_i} f_i^\phi(s_i^\phi) \\
\text{over} \quad & v, s, S, \ell \\
\text{s.t.} \quad & \mathcal{P}_i(v_{A_i}) - v_i + z_i S_i^H + S_i z_i^H - z_i \ell_i z_i^H = 0 && i \in \mathcal{E} \\
& \text{diag}\left(\sum_{i\in C_i} \mathcal{P}_i(S_j - z_j\ell_j) - S_i\right) + s_i = 0 && i \in \mathcal{N} \\
& \begin{pmatrix} v_i & S_i \\ S_i^H & \ell_i \end{pmatrix} \in \mathbb{S}_+ && i \in \mathcal{E} \\
& \text{rank}\begin{pmatrix} v_i & S_i \\ S_i^H & \ell_i \end{pmatrix} = 1 && i \in \mathcal{E} \\
& s_i^\phi \in \mathcal{I}_i^\phi && \phi \in \Phi_i \text{ and } i \in \mathcal{N} \\
& \underline{v}_i^\phi \le v_i^{\phi\phi} \le \overline{v}_i^\phi && \phi \in \Phi_i \text{ and } i \in \mathcal{N},
\end{aligned}
$$

and the ROPF problem (3.17), which removes the rank-1 constraint, can be written explicitly as:

$$
\begin{aligned}
\min \quad & \sum_{i\in\mathcal{N}}\sum_{\phi\in\Phi_i} f_i^\phi(s_i^\phi) && && \text{(7.3a)} \\
\text{over} \quad & v, s, S, \ell && && \text{(7.3b)} \\
\text{s.t.} \quad & \mathcal{P}_i(v_{A_i}) - v_i + z_i S_i^H + S_i z_i^H - z_i \ell_i z_i^H = 0 && i \in \mathcal{E} && \text{(7.3c)} \\
& \text{diag}\left(\sum_{i\in C_i} \mathcal{P}_i(S_j - z_j\ell_j) - S_i\right) + s_i = 0 && i \in \mathcal{N} && \text{(7.3d)} \\
& \begin{pmatrix} v_i & S_i \\ S_i^H & \ell_i \end{pmatrix} \in \mathbb{S}_+ && i \in \mathcal{E} && \text{(7.3e)} \\
& s_i^\phi \in \mathcal{I}_i^\phi && \phi \in \Phi_i \text{ and } i \in \mathcal{N} && \text{(7.3f)} \\
& \underline{v}_i^\phi \le v_i^{\phi\phi} \le \overline{v}_i^\phi && \phi \in \Phi_i \text{ and } i \in \mathcal{N} && \text{(7.3g)}
\end{aligned}
$$

where $\mathbb{S}_+$ is the set of (Hermitian) positive semi-defintie matrices. Under assumption A3, the relaxation (7.3) is exact and has the same optimal solution with the the original OPF problem.

Denote

$$x_i := \{v_i, s_i, S_i, \ell_i\} \tag{7.4}$$

$$\mathcal{K}_i := \left\{ \begin{pmatrix} v_i & S_i \\ S_i^H & \ell_i \end{pmatrix} \in \mathbb{S}_+,\ \{s_i^\phi \in \mathcal{I}_i^\phi \mid \phi \in \Phi_i\},\ \{\underline{v}_i^\phi \le v_i^{\phi\phi} \le \overline{v}_i^\phi \mid \phi \in \Phi_i\} \right\}. \tag{7.5}$$

Then (7.3) takes the form of (5.6), where (7.3c)–(7.3d) correspond to (5.6c) and (7.3e)–(7.3g) correspond to (5.6d).

A4 is a technical assumption that is required for the existence of efficient solutions for all the subproblems when the algorithm in Chapter 5.2.2 is applied to solve the ROPF problem. In practice, the objective function $f_i^\phi(s)$, usually takes the form of $f_i^\phi(s) := \frac{\alpha_i^\phi}{2} p^2 + \beta_i^\phi p$, which models both line loss and generation cost minimization as discussed in Chapter 3.2. For the injection region $\mathcal{I}_i^\phi$, it usually takes either (3.14a) or (3.14b). Then we can use the same procedure as (6.1) to solve (7.1) with closed form solution, which is shown in Appendix 6.B.

7.2 Distributed OPF Algorithm on Unbalanced Networks

We develop in this section a distributed algorithm that solves the ROPF problem (7.3) based on the approach developed in Chapter 5.2.2 under assumption A1–A4. Specifically, the global problem is decomposed into local subproblems that can be solved in a distributed manner with only neighborhood communication. In addition, we provide a sufficient condition, which holds in practice, for the existence of efficient solution to the optimization subproblems, thus eliminating the need for an iterative procedure to solve a SDP problem for each ADMM iteration.

Following a similar method in Chapter 6 for *balanced* radial networks, we first rewrite (7.3) in the form of (5.23), the differences between balanced and unbalanced networks are threefold:

1. For *balanced* networks, we show that $R_i = 0$ for $i \in \mathcal{N}$ suffices to provide an efficient solution. In contrast, for *unbalanced* networks, $R_i = 1$ for $i \in \mathcal{N}$ are required to have computationally efficient solutions for all the subproblems.

2. There are closed form solutions for all the subproblems for the algorithm of *balanced* networks. For *unbalanced* networks, there exists one subproblem that can only be reduced to an eigenvalue problem for a 6×6 matrix, which does not admits a closed form solution. However, we demonstrate in the simulation that a 100x speedup is still achieved on the standard IEEE test networks compared with using iterative optimization solvers.

3. For balanced networks, we use standard augmented Lagrangian (5.2) for our ADMM based algorithm. For *unbalanced* network, we rely on using the generalized augmented Lagrangian (5.5) to have an efficient ADMM based algorithm.

As stated above, we will use $R_i = 1$, i.e. we have two partitions $\mathcal{K}_{i0}$ and $\mathcal{K}_{i1}$ of set $\mathcal{K}_i$ for each node $i \in \mathcal{N}$, given as

$$\mathcal{K}_{i0} := \left\{ \begin{pmatrix} v_i & S_i \\ S_i^H & \ell_i \end{pmatrix} \in \mathbb{S}_+,\ \{s_i^\phi \in \mathcal{I}_i^\phi \mid \phi \in \Phi_i\} \right\} \tag{7.6}$$

$$\mathcal{K}_{i1} := \{\underline{v}_i^\phi \le v_i^{\phi\phi} \le \overline{v}_i^\phi \mid \phi \in \Phi_i\}. \tag{7.7}$$

Recall that there is always a closed form solution to the optimization subproblem (5.30) in the y-update, we next show the subproblems in the x-update can also be solved efficiently under assumption A4.

Note that the ROPF problem (7.3) falls in the form of (5.6), where (7.3e)–(7.3f) are local constraints to node (bus) i that correspond to $\mathcal{K}_i$ in (5.6d), (7.3c) and (7.3d) describe the coupling constraints among node i and its neighbors that corresponds to (5.6c). Similar to the design of the algorithm for balanced networks, we can transform (7.3) in the form of (5.23) as below:

$$\min \quad \sum_{i\in\mathcal{N}} \sum_{\phi\in\Phi_i} f_i^\phi((s_{i0}^\phi)^{(x)}) \tag{7.8a}$$

$$\text{over} \quad x := \{x_{ir} \mid 0 \le r \le 1,\ i \in \mathcal{N}\}$$

$$y := \{y_{ji} \mid j \in N_i,\ i \in \mathcal{N}\}$$

$$\text{s.t.} \quad \mathcal{P}_i(v_{A_i i}^{(y)}) - v_{ii}^{(y)} + z_i(S_{ii}^{(y)})^H + S_{ii}^{(y)} z_i^H - z_i \ell_{ii}^{(y)} z_i^H = 0 \qquad i \in \mathcal{E} \tag{7.8b}$$

$$\text{diag}\left(\sum_{i\in C_i} \mathcal{P}_i(S_{ji}^{(y)} - z_j \ell_{ji}^{(y)}) - S_{ii}^{(y)} \right) + s_{ii}^{(y)} = 0 \qquad i \in \mathcal{N} \tag{7.8c}$$

$$\begin{pmatrix} v_{i0}^{(x)} & S_{i0}^{(x)} \\ (S_{i0}^{(x)})^H & \ell_{i0}^{(x)} \end{pmatrix} \in \mathbb{S}_+ \qquad i \in \mathcal{E} \tag{7.8d}$$

$$(s_{i0}^\phi)^{(x)} \in \mathcal{I}_i^\phi \qquad \phi \in \Phi_i \text{ and } i \in \mathcal{N} \tag{7.8e}$$

$$\underline{v}_i^\phi \le (v_{i1}^{\phi\phi})^{(x)} \le \overline{v}_i^\phi \qquad \phi \in \Phi_i \text{ and } i \in \mathcal{N} \tag{7.8f}$$

$$x_{ir} - y_{ii} = 0 \qquad r = 1 \text{ and } i \in \mathcal{N} \tag{7.8g}$$

$$x_{i0} - y_{ij} = 0 \qquad j \in N_i \text{ and } i \in \mathcal{N}, \tag{7.8h}$$

where we put superscript $(\cdot)^{(x)}$ and $(\cdot)^{(y)}$ on each variable to denote whether the variable is updated in the x-update or y-update step. The problem (7.8) falls in the general form of (5.23) with $R_i = 1$, where $\mathcal{K}_{i0}$ and $\mathcal{K}_{i1}$ are defined in (7.7). Similar to balanced radial networks, each bus i does not need full information of its neighbor. For each node i, only voltage information $v_{A_i i}^{(y)}$ is needed from its parent A_i and branch power $S_{ji}^{(y)}$ and current $\ell_{ji}^{(y)}$ information from its children $j \in C_i$ based on

(7.8). Thus, y_{ij} contains only partial information about x_{i0}, i.e.

$$y_{ij} := \begin{cases} (S_{ii}^{(y)}, \ell_{ii}^{(y)}, v_{ii}^{(y)}, s_{ii}^{(y)}) & j = i \\ (S_{iA_i}^{(y)}, \ell_{iA_i}^{(y)}) & j = A_i \\ (v_{ij}^{(y)}) & j \in C_i \end{cases}.$$

On the other hand, only x_{i0} needs to hold all the variables and it suffices for x_{i1} to only have a duplicate of v_i, i.e.

$$x_{ir} := \begin{cases} (S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)}, s_{i0}^{(x)}) & r = 0 \\ (v_{i1}^{(x)}) & r = 1 \end{cases}.$$

As a result, x_{ir}, y_{ii} in (7.8g) and x_{i0}, y_{ij} in (7.8h) do not consist of the same components. Here, we abuse notations in both (7.8g) and (7.8h), which are composed of components that appear in both items, i.e.

$$\begin{aligned} x_{i0} - y_{ij} &:= \begin{cases} (S_{i0}^{(x)} - S_{ii}^{(y)}, \ell_{i0}^{(x)} - \ell_{ii}^{(y)}, v_{i0}^{(x)} - v_{ii}^{(y)}, s_{i0}^{(x)} - s_{ii}^{(y)}) & j = i \\ (S_{i0}^{(x)} - S_{iA_i}^{(y)}, \ell_{i0}^{(x)} - \ell_{iA_i}^{(y)}) & j = A_i \\ (v_{i0}^{(x)} - v_{ij}^{(y)}) & j \in C_i \end{cases} \\ x_{ir} - y_{ii} &:= \begin{cases} (v_{i1}^{(x)} - v_{ii}^{(y)}) & r = 1 \end{cases}. \end{aligned}$$

Recall that the local variables to each node i include x_{ir} $(0 \le r \le 1)$, y_{ji} $j \in N_i$, and the corresponding multiplier λ_{ir}, μ_{ji}. Then the set of local variables $\mathcal{A}_i$ (defined in (5.25)) for node i can be written explicitly as

$$\mathcal{A}_i := \{x_{ir},\ \lambda_{ir} \mid 0 \le r \le 1\} \cup \{y_{ji}, \mu_{ji} \mid j \in N_i\},$$

where λ denote the Lagrangian multiplier for (7.8g) and μ the Lagrangian multiplier for (7.8g). The detailed mapping between variables and those multipliers are illustrated in Table 7.1.

Next, we apply the approach in Chapter 5.2.2 to develop a distributed algorithm that solves the ROPF problem. To obtain an efficient solution, we will use the generalized augmented Lagrangian

Table 7.1: Multipliers associated with constraints (7.8g)-(7.8h)

λ_{i1}:	$v_{i1}^{(x)} = v_{ii}^{(y)}$		
$\mu_{ii}^{(1)}$:	$S_{i0}^{(x)} = S_{ii}^{(y)}$	$\mu_{ii}^{(2)}$:	$\ell_{i0}^{(x)} = \ell_{ii}^{(y)}$
$\mu_{ii}^{(3)}$:	$v_{i0}^{(x)} = v_{ii}^{(y)}$	$\mu_{ii}^{(4)}$:	$s_{i0}^{(x)} = s_{ii}^{(y)}$
$\mu_{iA_i}^{(1)}$:	$S_{iA_i}^{(x)} = S_{iA_i}^{(y)}$	$\mu_{iA_i}^{(2)}$:	$\ell_{i}^{(x)} = \ell_{iA_i}^{(y)}$
μ_{ij}:	$v_{i}^{(x)} = v_{ij}^{(y)}$		

(5.5), which can be explicitly written as below for our problem.

$$L_\rho(x, y, \lambda, \mu) \tag{7.9a}$$

$$= \sum_{i \in \mathcal{N}} \left(f_i(s_{i0}^{(x)}) + \langle \lambda_{i1}, x_{i1} - y_{ii} \rangle + \langle \mu_{ii}, x_{i0} - y_{ii} \rangle + \langle \mu_{iA_i}, x_{i0} - y_{iA_i} \rangle + \sum_{j \in C_i} \langle \mu_{ij}, x_{i0} - y_{ij} \rangle + \right.$$

$$\left. \frac{\rho}{2} \left(\|x_{i1} - y_{ii}\|_2^2 + \|x_{i0} - y_{ii}\|_s^2 + \|x_{i0} - y_{iA_i}\|_n^2 + \sum_{j \in C_i} \|x_{i0} - y_{ij}\|_2^2 \right) \right), \tag{7.9b}$$

where

$$\|x_{i0} - y_{ii}\|_s^2 := (2|C_i| + 3)\|S_{i0}^{(x)} - S_{ii}^{(y)}\|_2^2 + (|C_i| + 1)\|\ell_{i0}^{(x)} - \ell_{ii}^{(y)}\|_2^2 + 2\|v_{i0}^{(x)} - v_{ii}^{(y)}\|_2^2 + \|s_{i0}^{(x)} - s_{ii}^{(y)}\|_2^2 \tag{7.10a}$$

$$\|x_{i0} - y_{iA_i}\|_n^2 := \|S_{i0}^{(x)} - S_{iA_i}^{(y)}\|_2^2 + \|\ell_{i,A_i}^{(x)} - \ell_i^{(z)}\|_2^2. \tag{7.10b}$$

Here, the weights in front of each term of (7.10a) are different and the corresponding Lagrangian (7.9) takes the form of (5.2). We still denote the augmented Lagrangian as $L_\rho(\cdot)$ instead of $L_\Lambda(\cdot)$ since the only parameter that appears in the generalized quadratic penalty term is ρ.

Using the generalized augmented Lagrangian (7.9), we now show how to solve the x-update subproblem (5.27) using eigen-decomposition under assumption A4. For ease of presentation, we remove the iteration number k in (5.3) for all the variables, which will be updated accordingly after each subproblem is solved.

In the x-update, each bus i needs to solve two subproblems (7.11) and (7.12), which update x_{i0}

and x_{i1}, respectively. The first one that updates x_{i0} is

$$\begin{aligned} \min \quad & H_{i0}(x_{i0}) && (7.11a)\\ \text{over} \quad & x_{i0} = \{v_{i0}^{(x)}, \ell_{i0}^{(x)}, S_{i0}^{(x)}, s_{i0}^{(x)}\} && (7.11b)\\ \text{s.t.} \quad & \begin{pmatrix} v_{i0}^{(x)} & S_{i0}^{(x)} \\ (S_{i0}^{(x)})^H & \ell_{i0}^{(x)} \end{pmatrix} \in \mathbb{S}_+ && (7.11c)\\ & (s_{i0}^{\phi})^{(x)} \in \mathcal{I}_i^{\phi} \qquad \phi \in \Phi_i, && (7.11d) \end{aligned}$$

where $H_{i0}(x_{i0})$ is defined in (5.28) and can be written explicitly as

$$\begin{aligned} H_{i0}(x_{i0}) \quad = \quad & f_i(s_{i0}^{(x)}) + \langle \mu_{ii}, x_{i0} \rangle + \langle \mu_{iA_i}, x_{i0} \rangle + \sum_{j \in C_i} \langle \mu_{ij}, x_{i0} \rangle \\ & + \frac{\rho}{2} \left(\|x_{i0} - y_{ii}\|_s^2 + \|x_{i0} - y_{iA_i}\|_n^2 + \sum_{j \in C_i} \|x_{i0} - y_{ij}\|_2^2 \right). \end{aligned}$$

The second problem that updates x_{i1} is

$$\begin{aligned} \min \quad & H_{i1}(x_{i1}) \\ \text{over} \quad & x_{i1} = \{v_{i1}^{(x)}\} && (7.12)\\ \text{s.t.} \quad & \underline{v}_i^{\phi} \leq (v_{i1}^{\phi\phi})^{(x)} \leq \overline{v}_i^{\phi} \qquad \phi \in \Phi_i, \end{aligned}$$

where

$$H_{i1}(x_{i1}) = \langle \lambda_{i1}, x_{i1} \rangle + \frac{\rho}{2} \left(\|x_{i1} - y_{ii}\|_2^2 \right).$$

We first derive an efficient solution for problem (7.11). Note that $H_{i0}(x_{i0})$ can be further decomposed as

$$\begin{aligned} H_{i0}(x_{i0}) \quad = \quad & f_i(s_{i0}^{(x)}) + \langle \mu_{ii}, x_{i0} \rangle + \langle \mu_{iA_i}, x_{i0} \rangle + \sum_{j \in C_i} \langle \mu_{ij}, x_{i0} \rangle && (7.13)\\ & + \frac{\rho}{2} \left(\|x_{i0} - y_{ii}\|_s^2 + \|x_{i0} - y_{iA_i}\|_n^2 + \sum_{j \in C_i} \|x_{i0} - y_{ij}\|_2^2 \right) \\ = \quad & \frac{\rho(|C_i| + 2)}{2} H_i^{(1)}(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)}) + H_i^{(2)}(s_{i0}^{(x)}) + \text{constant}, \end{aligned}$$

where

$$\begin{aligned} H_i^{(1)}(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)}) &= \left\| \begin{pmatrix} v_{i0}^{(x)} & S_{i0}^{(x)} \\ (S_{i0}^{(x)})^H & \ell_{i0}^{(x)} \end{pmatrix} - \begin{pmatrix} \hat{v}_i & \hat{S}_i \\ \hat{S}_i^H & \hat{\ell}_i \end{pmatrix} \right\|_2^2 \\ H_i^{(2)}(s_{i0}^{(x)}) &= f_i(s_{i0}^{(x)}) + \frac{\rho}{2} \|s_{i0}^{(x)} - \hat{s}_i\|_2^2. \end{aligned}$$

The last step in (7.13) is obtained by completing the square and the variables labeled with hat are some constants for the x-update.

Hence, the objective (7.11a) in (7.11) can be decomposed into two parts, where the first part $H^{(1)}(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)})$ involves variables $(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)})$ and the second part $H^{(2)}(s_{i0}^{(x)})$ involves $s_{i0}^{(x)}$. Note that the constraint (7.11c)–(7.11d) can also be separated into two independent constraints. Variables $(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)})$ only depend on (7.11c) and $s_{i0}^{(x)}$ depends on (7.11d). Then (7.11) can be decomposed into two subproblems, where the first one (7.14) solves the optimal $(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)})$ and the second one (7.15) solves the optimal $s_{i0}^{(x)}$. The first subproblem can be written explicitly as

$$\begin{aligned} \min \quad & H_i^{(1)}(S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)}) \\ \text{over} \quad & S_{i0}^{(x)}, \ell_{i0}^{(x)}, v_{i0}^{(x)} \\ \text{s.t.} \quad & \begin{pmatrix} v_{i0}^{(x)} & S_{i0}^{(x)} \\ (S_{i0}^{(x)})^H & \ell_{i0}^{(x)} \end{pmatrix} \in \mathbb{S}_+, \end{aligned} \tag{7.14}$$

which can be solved using eigen-decomposition of a 6×6 matrix via the following theorem.

Theorem 7.1. *Suppose $W \in \mathbb{S}^n$ and denote $X(W) := \arg\min_{X \in \mathbb{S}_+} \|X - W\|_2^2$. Then $X(W) = \sum_{i:\lambda_i > 0} \lambda_i u_i u_i^H$, where λ_i, u_i are the i^{th} eigenvalue and orthonormal eigenvector of matrix W, respectively.*

Proof. The proof is in Appendix 7.A. □

Denote

$$W := \begin{pmatrix} \hat{v}_i & \hat{S}_i \\ \hat{S}_i^H & \hat{\ell}_i \end{pmatrix} \text{ and } X := \begin{pmatrix} v_{i0}^{(x)} & S_{i0}^{(x)} \\ (S_{i0}^{(x)})^H & \ell_{i0}^{(x)} \end{pmatrix}.$$

Then (7.14) can be written compactly as

$$\min_X \|X - W\|_2^2 \quad \text{s.t. } X \in \mathbb{S}_+,$$

which can be solved efficiently using eigen-decomposition based on Theorem 7.1. The second problem

is

$$\begin{aligned} \min \quad & f_i(s_{i0}^{(x)}) + \frac{\rho}{2}\|s_{i0}^{(x)} - \hat{s}_i\|_2^2 \\ \text{over} \quad & s_{i0}^{(x)} \\ \text{s.t.} \quad & (s_{i0}^{\phi})^{(x)} \in \mathcal{I}_i^{\phi} \qquad \phi \in \Phi_i. \end{aligned} \tag{7.15}$$

Recall that if $f_i(s_{i0}^{(x)}) = \sum_{\phi\in\Phi_i} f_i^{\phi}((s_{i0}^{\phi})^{(x)})$, then both the objective and constraint are separable for each phase $\phi \in \Phi_i$. Therefore, (7.15) can be further decomposed into $|\Phi_i|$ number of subproblems as below.

$$\begin{aligned} \min \quad & f_i^{\phi}((s_{i0}^{\phi})^{(x)}) \\ \text{over} \quad & (s_{i0}^{\phi})^{(x)} \\ \text{s.t.} \quad & (s_{i0}^{\phi})^{(x)} \in \mathcal{I}_i^{\phi}, \end{aligned} \tag{7.16}$$

which takes the same form as of assumption A4 and thus can be solved with closed form solution based on the assumptions.

For the problem (7.12) that updates x_{i1}, which only has one component $v_{i1}^{(x)}$, the closed form solution is given as

$$(v_{i1}^{\phi_1\phi_2})^{(x)} = \begin{cases} \left[\frac{\lambda_{i1}^{\phi_1\phi_2}}{\rho} + (v_{ii}^{\phi_1\phi_2})^{(y)}\right]_{\underline{v}_i^{\phi_1}}^{\overline{v}_i^{\phi_1}} & \phi_1 = \phi_2 \\ \frac{\lambda_{i1}^{\phi_1\phi_2}}{\rho} + (v_{ii}^{\phi_1\phi_2})^{(y)} & \phi_1 \neq \phi_2 \end{cases}.$$

To summarize, we show that under assumption A4, the problems in the x-update for each bus i either has a closed form solution or can be solved via eigen-decomposition. Then we have the following theorem.

Theorem 7.2. *Under assumption A4, the subproblems for ROPF in the x-update* (5.27) *with* $R_i = 1$ *can be solved via either closed form solutions or eigen-decomposition of a* 6×6 *hermitian matrix.*

In the y-update, the subproblem solved by each node i takes the form of (5.30) and can be written explicitly as

$$\begin{aligned} \min \quad & G_i(\{y_{ji} \mid j \in N_i\}) \\ \text{over} \quad & \{y_{ji} \mid j \in N_i\} \\ \text{s.t.} \quad & \mathcal{P}_i(v_{A_i i}^{(y)}) - v_{ii}^{(y)} + z_i(S_{ii}^{(y)})^H + S_{ii}^{(y)} z_i^H - z_i \ell_{ii}^{(y)} z_i^H = 0 \\ & \operatorname{diag}\left(\sum_{i\in C_i} \mathcal{P}_i(S_{ji}^{(y)} - z_j \ell_{ji}^{(y)}) - S_{ii}^{(y)}\right) + s_{ii}^{(y)} = 0, \end{aligned} \tag{7.17}$$

which has a closed form solution given in (5.16) and we do not reiterate here.

Finally, we specify the initialization and stopping criteria for the algorithm. Similar to the algorithm for balanced networks, a good initialization usually reduces the number of iterations for convergence. We use the following initialization suggested by our empirical results. We first initialize the auxiliary variables $\{V_i \mid i \in \mathcal{N}\}$ and $\{I_i \mid i \in \mathcal{E}\}$, which represent the complex nodal voltage and branch current, respectively. Then we use these auxiliary variables to initialize the variables in (7.8). Intuitively, the above initialization procedure can be interpreted as finding a solution to the bus injection model (3.10), assuming zero impedance on all the lines. The procedure is formally stated in Algorithm 7.1.

Algorithm 7.1 Initialization of the Algorithm

$V_i^a = 1$, $V_i^b = e^{-\frac{2}{3}\pi}$, $V_i^c = e^{\frac{2}{3}\pi}$ for $i \in \mathcal{N}$
Initialize s_i^ϕ using any point in the injection region $\mathcal{I}_i^\phi$ for $i \in \mathcal{N}$
Initialize $\{I_i^\phi \mid \phi \in \Phi_i \; i \in \mathcal{N}\}$ by calling DFS(0,ϕ) for $\phi \in \Phi_i$
$v_{i0}^{(x)} = V_i V_i^H$, $\ell_{i0}^{(x)} = I_i I_i^H$, $S_{i0}^{(x)} = V_i I_i^H$ and $s_{i0}^{(x)} = s_i$ for $i \in \mathcal{N}$
$y_{ij} = x_{i0}$ for $j \in N_i$ and $i \in \mathcal{N}$
$x_{i1} = x_{i0}$ for $i \in \mathcal{N}$

function DFS(i,ϕ)
    $I_i^\phi = (\frac{s_i^\phi}{V_i^\phi})^*$
    for $j \in C_i$ **do**
        $I_i^\phi +=$ DFS(j, ϕ)
    end for
    return I_i^ϕ
end function

We use the same stopping criteria as the one used for balanced networks in Chapter 6, i.e. both the primal residual r^k and dual residual s^k are below $10^{-4}\sqrt{|\mathcal{N}|}$. The pseudo code for the complete algorithm is summarized in Algorithm 7.2.

Algorithm 7.2 Distributed OPF algorithm on Unbalanced Radial Networks

Input:network $\mathcal{G}(\mathcal{N}, \mathcal{E})$, power injection region $\mathcal{I}_i$, voltage region $(\underline{v}_i, \overline{v}_i)$, line impedance z_i for $i \in \mathcal{N}$.
Output:voltage v, power injection s

Initialize the x and y variables using Algorithm 7.1.
while $r^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **or** $s^k > 10^{-4}\sqrt{|\mathcal{N}|}$ **do**
    In the x-update, each agent i solves both (7.11) and (7.12) to update x_{i0} and x_{i1}.
    In the y-update, each agent i solves (7.17) to update y_{ji} for $j \in N_i$.
    In the multiplier update, update λ, μ by (5.3c).
end while

Table 7.1: Statistics of different networks

Network	Diameter	Iteration	Total Time(s)	Avg time(s)
IEEE 13Bus	6	289	17.11	1.32
IEEE 34Bus	20	547	78.34	2.30
IEEE 37Bus	16	440	75.67	2.05
IEEE 123Bus	30	608	306.3	2.49

7.3 Case Study

In this section, we first demonstrate the scalability of the distributed algorithm proposed in section 7.2 by testing it on the standard IEEE test feeders [48]. To show the efficiency of the proposed algorithm, we also compare the computation time of solving the subproblems in both the x and y-update with off-the-shelf solver (SDPT3 [85]). Second, we run the proposed algorithm on networks of different topology to understand the factors that affect the convergence rate. The algorithm is implemented in Python and run on a Macbook pro 2014 with i5 dual core processor.

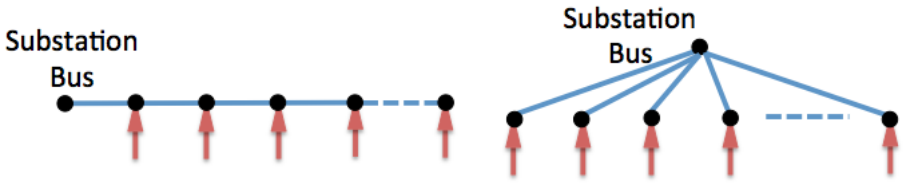
7.3.1 Simulations on IEEE test feeders

We test the proposed algorithm on the IEEE 13, 34, 37, 123 bus distribution systems. All the networks have unbalanced three phase. The substation is modeled as a fixed voltage bus (1 p.u.) with infinite power injection capability. The other buses are modeled as load buses whose voltage magnitude at each phase can vary within $[0.95, 1.05]$ p.u. and power injections are specified in the test feeder. There is no controllable device in the original IEEE test feeders, and hence the OPF problem degenerates to a power flow problem, which is easy to solve. To demonstrate the effectiveness of the algorithm, we replace all the capacitors with inverters, whose reactive power injection ranges from 0 to the maximum ratings specified by the original capacitors. The objective is to minimize power loss across the network, namely $f_i^\phi(s_i^\phi) = p_i^\phi$ for $\phi \in \Phi_i$ and $i \in \mathcal{N}$.

We mainly focus on the time of convergence (ToC) for the proposed distributed algorithm. The algorithm is run on a single machine. To roughly estimate the ToC (excluding communication overhead) if the algorithm is run on multiple machines in a distributed manner, we divide the total time by the number of buses.

In Table 7.1, we record the number of iterations to converge, total computation time required to run on a single machine and the average time required for each node if the algorithm is run on multiple machines excluding communication overhead. From the simulation results, the proposed algorithm converges within 2.5 second for all the standard IEEE test networks if the algorithm is run in a distributed manner.

Moreover, we show the advantage of using the proposed algorithm by comparing the computa-



(a) Line network (b) Fat tree network

Figure 7.1: Topologies for line and fat tree networks.

Table 7.2: Statistics of line and fat tree networks

Size	# of iterations (Line)	# of iterations (Fat tree)
5	57	61
10	253	111
15	414	156
20	579	197
25	646	238
30	821	272
35	1353	304
40	2032	337
45	2026	358
50	6061	389

tion time of solving the subproblems between off-the-shelf solver (SDPT3) and our algorithm. In particular, we compare the average computation time of solving the subproblem in both the x and y update. In the x-update, the average time required to solve the subproblem (7.17) is 9.8×10^{-5}s for our algorithm but 0.13s for SDPT3. In the y-update, the average time required to solve the subproblems (7.11)–(7.12) are 3.7×10^{-3}s for our algorithm but 0.45s for SDPT3. Thus, each ADMM iteration takes about 3.8×10^{-3}s for our algorithm but 5.8×10^{-1}s for using iterative algorithm, a more than 100x speedup.

7.3.2 Rate of convergence

In section 7.3.1, we demonstrate that the proposed distributed algorithm can dramatically reduce the computation time within each iteration. The time of convergence (ToC) is determined by both the computation time required within each iteration and the number of iterations. In this subsection, we study the number of iterations, namely rate of convergence.

Rate of convergence is determined by many different factors. Here, we only consider two factors, network size N, and diameter D, i.e. given the termination criteria in Algorithm 7.2, the impact of network size and diameter on the number of iterations. The impact from other factors, e.g. form of objective function and constraints, is beyond the scope of this thesis.

Similar to the method used for balanced networks in Chapter 6, we simulate the algorithm on two extreme cases: 1) Line network in Fig. 7.1a, whose diameter is the largest given the network size, and 2) Fat tree network in Fig. 7.1b, whose diameter is the smallest given the network size. In Table 7.2, we record the number of iterations for both line and fat tree network of different sizes. For the line network, the number of iterations increases notably as the size increases. For the fat tree network, the trend is much less prominent compared to the line network. It means that network diameter has a stronger impact than network size on the rate of convergence.

7.4 Conclusion

In this chapter, we have developed a distributed algorithm for OPF problem on unbalanced radial distribution networks based on ADMM. We show that the optimization subproblems for each node reduce to either a closed form solution or an eigenvalue problem whose size remains constant. We have tested the algorithm on standard IEEE test networks, and the optimization subproblems in each ADMM iteration are solved 100x faster than using generic iterative optimization solver.

Appendix

7.A Proof of Theorem 7.1

Let Λ_W :=diag$(\lambda_i, 1 \le i \le n)$ denote the diagonal matrix consisting of the eigenvalues of matrix W. Let $U := (u_i, 1 \le i \le n)$ denote the unitary matrix. Since $W \in \mathbb{H}^n$, $U^{-1} = U^H$ and $W = U\Lambda_W U^H$. Then

$$\begin{aligned} \|X - W\|_2^2 &= tr((X-W)^H(X-W)) \\ &= tr((X-W)(X-W)) \\ &= tr(U^H(X-W)UU^H(X-W)U) \\ &= tr((U^HXU - \Lambda_W)(U^HXU - \Lambda_W)). \end{aligned}$$

Denote $\hat{X} := U^HXU = (\hat{x}_{i,j}, i, j \in [1, n])$, note that $\hat{X} \in \mathbb{S}_+$ since $X \in \mathbb{S}_+$. Then

$$\begin{aligned} \|X - W\|_2^2 &= \sum_{i=1}^{n}(\hat{x}_{ii} - \lambda_i)^2 + \sum_{i \ne j}|\hat{x}_{ij}|^2 && (7.18) \\ &\ge \sum_{i=1}^{n}(\hat{x}_{ii} - \lambda_i)^2 && (7.19) \\ &\ge \sum_{i:\lambda_i \le 0}\lambda_i^2, && (7.20) \end{aligned}$$

where the last inequality follows from $\hat{x}_{ii} > 0$ because $\hat{X} \in \mathbb{S}_+$. The equality in (7.20) can be obtained by letting

$$\hat{x}_{ij} := \begin{cases} \lambda_i & i = j, \quad \lambda_i > 0, \\ 0 & \text{otherwise} \end{cases},$$

which means $X(W) = U\hat{X}U^H = \sum_{i:\lambda_i > 0}\lambda_i u_i u_i^H$.

Bibliography

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356, 2010.

[2] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

[3] O. Alsac, J. Bright, M. Prais, and B. Stott. Further developments in lp-based optimal power flow. *Power Systems, IEEE Transactions on*, 5(3):697–711, 1990.

[4] X. Bai, H. Wei, K. Fujisawa, and Y. Wang. Semidefinite programming for optimal power flow problems. *Int'l J. of Electrical Power & Energy Systems*, 30(6-7):383–392, 2008.

[5] R. Baldick, B. H. Kim, C. Chase, and Y. Luo. A fast distributed implementation of optimal power flow. *Power Systems, IEEE Transactions on*, 14(3):858–864, 1999.

[6] M. Baran and F. Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Trans. on Power Delivery*, 4(2):1401–1407, Apr 1989.

[7] M. E. Baran and F. F. Wu. Optimal Capacitor Placement on radial distribution systems. *IEEE Trans. Power Delivery*, 4(1):725–734, 1989.

[8] M. E. Baran and F. F. Wu. Optimal Sizing of Capacitors Placed on A Radial Distribution System. *IEEE Trans. Power Delivery*, 4(1):735–743, 1989.

[9] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.

[10] A. Borghetti. A mixed-integer linear programming approach for the computation of the minimum-losses radial configuration of electrical distribution networks. *Power Systems, IEEE Transactions on*, 27(3):1264–1273, 2012.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[13] M. Carbone and L. Rizzo. Dummynet revisited. *ACM SIGCOMM Computer Communication Review*, 40(2):12–20, 2010.

[14] J. Carpentier. Contribution to the economic dispatch problem. *Bulletin de la Societe Francoise des Electriciens*, 3(8):431–447, 1962.

[15] E. M. Carreno, R. Romero, and A. Padilha-Feltrin. An efficient codification to solve distribution network reconfiguration for loss reduction problem. *Power Systems, IEEE Transactions on*, 23(4):1542–1551, 2008.

[16] H.-D. Chiang and M. E. Baran. On the existence and uniqueness of load flow solution for radial distribution power networks. *Circuits and Systems, IEEE Transactions on*, 37(3):410–416, 1990.

[17] H.-D. Chiang and R. Jean-Jumeau. Optimal network reconfigurations in distribution systems: Part 1: A new formulation and a solution methodology. *IEEE Trans. Power Delivery*, 5(4):1902–1909, November 1990.

[18] J.-P. Chiou, C.-F. Chang, and C.-T. Su. Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems. *Power Systems, IEEE Transactions on*, 20(2):668–674, 2005.

[19] S. Civanlar, J. Grainger, H. Yin, and S. Lee. Distribution feeder reconfiguration for loss reduction. *Power Delivery, IEEE Transactions on*, 3(3):1217–1223, 1988.

[20] E. Dall'Anese and G. B. Giannakis. Sparsity-leveraging reconfiguration of smart distribution systems. arXiv:1303.5802v2, August 2013.

[21] E. Dall'Anese, H. Zhu, and G. B. Giannakis. Distributed optimal power flow for smart microgrids. *arXiv preprint arXiv:1211.5856*, 2012.

[22] E. Dall'Anese, H. Zhu, and G. B. Giannakis. Distributed optimal power flow for smart microgrids. *Smart Grid, IEEE Transactions on*, 4(3):1464–1475, 2013.

[23] E. Devane and I. Lestas. Stability and convergence of distributed algorithms for the opf problem. In *52nd IEEE Conference on Decision and Control*, 2013.

[24] R. El Ramli, M. Awad, and R. Jabr. Ordinal optimization for dynamic network reconfiguration. *Electric Power Components and Systems*, 39(16):1845–1857, 2011.

[25] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy. Inverter var control for distribution systems with renewables. In *Proceedings of IEEE SmartGridComm Conference*, October 2011.

[26] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy. Inverter var control for distribution systems with renewables. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 457–462. IEEE, 2011.

[27] M. Farivar and S. H. Low. Branch flow model: relaxations and convexification (parts I, II). *IEEE Trans. on Power Systems*, 28(3):2554–2572, August 2013.

[28] M. Farivar, R. Neal, C. Clarke, and S. H. Low. Optimal inverter VAR control in distribution systems with high pv penetration. In *IEEE Power & Energy Society (PES) General Meeting*, July 2012.

[29] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):476–480, 1999.

[30] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. Tcp extensions for multipath operation with multiple addresses. *IETF MPTCP proposal*, 2009.

[31] L. Gan, N. Li, U. Topcu, and S. H. Low. Optimal power flow in distribution networks. In *Proc. 52nd IEEE Conference on Decision and Control*, December 2013. in arXiv:12084076.

[32] L. Gan, N. Li, U. Topcu, and S. H. Low. Exact convex relaxation of optimal power flow in radial networks. *IEEE Trans. on Automatic Control*, 2014.

[33] L. Gan and S. H. Low. Convex relaxations and linear approximation for optimal power flow in multiphase radial network. In *18th Power Systems Computation Conference (PSCC)*, 2014.

[34] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Trans. on Automatic Control*, 60(3):644–658, 2013.

[35] F. V. Gomes, S. Carneiro Jr, J. L. R. Pereira, M. P. Vinagre, P. A. N. Garcia, and L. R. De Araujo. A new distribution system reconfiguration approach using optimum power flow and sensitivity analysis for loss reduction. *Power Systems, IEEE Transactions on*, 21(4):1616–1623, 2006.

[36] S. Goswami and S. Basu. A new algorithm for the reconfiguration of distribution feeders for loss minimization. *Power Delivery, IEEE Transactions on*, 7(3):1484–1491, 1992.

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

[38] H. Han, S. Shakkottai, C. Hollot, R. Srikant, and D. Towsley. Overlay tcp for multi-path routing and congestion control. In *IMA Workshop on Measurements and Modeling of the Internet*, 2004.

[39] H. L. Hijazi and S. Thiebaux. Optimal ac distribution systems reconfiguration. In *18th Power Systems Computation Conference (PSCC), Wroclaw, POLAND*, 2014.

[40] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda. Multipath congestion control for shared bottleneck. In *Proc. PFLDNeT workshop*, 2009.

[41] M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.

[42] G. Hug-Glanzmann and G. Andersson. Decentralized optimal power flow control for overlapping areas in power systems. *Power Systems, IEEE Transactions on*, 24(1):327–336, 2009.

[43] J. R. Iyengar, P. D. Amer, and R. Stewart. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *Networking, IEEE/ACM Transactions on*, 14(5):951–964, 2006.

[44] R. Jabr. Radial Distribution Load Flow Using Conic Programming. *IEEE Trans. on Power Systems*, 21(3):1458–1459, Aug 2006.

[45] R. A. Jabr, R. Singh, and B. C. Pal. Minimum loss network reconfiguration using mixed-integer convex programming. *Power Systems, IEEE Transactions on*, 27(2):1106–1115, 2012.

[46] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *ACM SIGCOMM Computer Communication Review*, 35(2):5–12, 2005.

[47] F. P. Kelly, A. K. Maulloo, and D. K. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[48] W. Kersting. Radial distribution test feeders. *Power Systems, IEEE Transactions on*, 6(3):975–985, 1991.

[49] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Inc., 2 edition, 1996.

[50] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec. Mptcp is not pareto-optimal: Performance issues and a possible solution. *IEEE/ACM Transactions on Networking (ToN)*, 21:1651–1665, 2013.

[51] H. Khodr, J. Martinez-Crespo, M. Matos, and J. Pereira. Distribution systems reconfiguration based on opf using benders decomposition. *Power Delivery, IEEE Transactions on*, 24(4):2166–2176, 2009.

[52] B. H. Kim and R. Baldick. Coarse-grained distributed optimal power flow. *Power Systems, IEEE Transactions on*, 12(2):932–939, 1997.

[53] M. Kraning, E. Chu, J. Lavaei, and S. Boyd. Dynamic network energy management via proximal message passing. *Optimization*, 1(2):1–54, 2013.

[54] A. Lam, B. Zhang, A. Dominguez-Garcia, and D. Tse. Optimal distributed voltage regulation in power distribution networks. *arXiv preprint arXiv:1204.5226*, 2012.

[55] A. Lam, B. Zhang, and D. N. Tse. Distributed algorithms for optimal power flow problem. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 430–437. IEEE, 2012.

[56] J. Lavaei and S. H. Low. Zero duality gap in optimal power flow problem. *Power Systems, IEEE Transactions on*, 27(1):92–107, 2012.

[57] N. Li, L. Chen, and S. H. Low. Demand response in radial distribution networks: Distributed algorithm. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pages 1549–1553. IEEE, 2012.

[58] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1):193–228, 1998.

[59] S. H. Low. A duality model of tcp and queue management algorithms. *Networking, IEEE/ACM Transactions on*, 11(4):525–536, 2003.

[60] S. H. Low. Convex relaxation of optimal power flow, I: formulations and relaxations. *IEEE Trans. on Control of Network Systems*, 1(1):15–27, March 2014.

[61] S. H. Low. Convex relaxation of optimal power flow, II: exactness. *IEEE Trans. on Control of Network Systems*, 1(2):177–189, June 2014.

[62] S. H. Low and D. E. Lapsley. Optimization flow control-i: basic algorithm and convergence. *IEEE/ACM Transactions on Networking (TON)*, 7(6):861–874, 1999.

[63] J. R. Mantovani, F. Casari, and R. A. Romero. Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão. *Revista Controle e Automação, Sociedade Brasileira de Automática, SBA*, 11(03):150–159, 2000.

[64] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.

[65] A. Merlin and H. Back. Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. In *Proc. of the Fifth Power System Conference (PSCC), Cambridge*, pages 1–18, 1975.

[66] Multipath TCP Linux implementation.

[67] F. J. Nogales, F. J. Prieto, and A. J. Conejo. A decomposition methodology applied to the multi-area optimal power flow problem. *Annals of operations research*, 120(1-4):99–116, 2003.

[68] G. Optimization et al. *URL: https://www.mosek.com/*.

[69] G. Optimization et al. Gurobi optimizer reference manual. *URL: http://www. gurobi. com*, 2012.

[70] Q. Peng and S. Low. Distributed algorithm for optimal power flow on a radial network. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014.

[71] Q. Peng and S. Low. Distributed algorithm for optimal power flow on an unbalanced radial network. In *Decision and Control (CDC), 2015 IEEE 54rd Annual Conference on*. IEEE, 2015.

[72] Q. Peng and S. H. Low. Optimal branch exchange for feeder reconfiguration in distribution networks. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2960–2965. IEEE, 2013.

[73] Q. Peng, Y. Tang, and S. H. Low. Feeder reconfiguration in distribution networks based on convex relaxation of opf. *Power Systems, IEEE Transactions on*, 30(1):1793–1804, 2014.

[74] Q. Peng, A. Walid, J.-S. Hwang, and S. H. Low. Multipath tcp: Analysis, design, and implementation. *IEEE/ACM Transactions on Networking (ToN)*, 2015.

[75] Q. Peng, A. Walid, and S. H. Low. Multipath tcp algorithms: theory and design. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, pages 305–316. ACM, 2013.

[76] R. J. Sarfi, M. Salama, and A. Chikhani. A survey of the state of the art in distribution system reconfiguration for system loss reduction. *Electric Power Systems Research*, 31(1):61–70, 1994.

[77] S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends® in Networking*, 2(3):271–379, 2007.

[78] B. Stott and O. Alsaç. Fast decoupled load flow. *power apparatus and systems, ieee transactions on*, (3):859–869, 1974.

[79] B. Stott, J. Jardim, and O. Alsac. Dc power flow revisited. *Power Systems, IEEE Transactions on*, 24(3):1290–1300, 2009.

[80] C.-T. Su, C.-F. Chang, and J.-P. Chiou. Distribution network reconfiguration for loss reduction by ant colony search algorithm. *Electric Power Systems Research*, 75(2):190–199, 2005.

[81] C.-T. Su and C.-S. Lee. Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution. *Power Delivery, IEEE Transactions on*, 18(3):1022–1027, 2003.

[82] A. X. Sun, D. T. Phan, and S. Ghosh. Fully decentralized ac optimal power flow algorithms. In *Power and Energy Society General Meeting (PES), 2013 IEEE*, pages 1–5. IEEE, 2013.

[83] D. L. Sun and C. Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6201–6205. IEEE, 2014.

[84] J. Taylor, F. S. Hover, et al. Convex models of distribution system reconfiguration. *Power Systems, IEEE Transactions on*, 27(3):1407–1413, 2012.

[85] K.-C. Toh, M. J. Todd, and R. H. Tütüncü. Sdpt3–a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[86] K. Turitsyn, S. Backhaus, M. Chertkov, et al. Options for control of reactive power by distributed photovoltaic generators. *Proceedings of the IEEE*, 99(6):1063–1073, 2011.

[87] G. Viswanadha Raju and P. Bijwe. An efficient algorithm for minimum loss reconfiguration of distribution system based on sensitivity and heuristics. *Power Systems, IEEE Transactions on*, 23(3):1280–1287, 2008.

[88] E. Wei and A. Ozdaglar. On the o (1/k) convergence of asynchronous distributed alternating direction method of multipliers. *arXiv preprint arXiv:1307.8254*, 2013.

[89] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, pages 8–8. USENIX Association, 2011.